

**SESSION**  
**SECURITY AND ALLIED TECHNOLOGIES**

**Chair(s)**

**TBA**



# A Survey of Security Services and Techniques in Distributed Storage Systems

Zhiqian Xu<sup>1,2</sup>, Keith Martin<sup>1</sup>, and Clifford L. Kotnik<sup>2</sup>

<sup>1</sup>Information Security Group, Royal Holloway, University of London, London, UK

<sup>2</sup>FedEx Corporation, Memphis, TN, USA

**Abstract**—*The rapid growth of data and data sharing have been driven an evolution in distributed storage infrastructure. The need for sensitive data protection and the capacity to handle massive data sets have encouraged the research and development of secure and scalable storage systems. This paper identifies major security issues and requirements of data protection related to distributed data storage systems. We classify the security services and techniques in existing or proposed storage systems. We then discuss potential research topics and future trends.*

**Keywords:** Distributed Storage Systems, Data Security

## 1. Introduction

The need for distributed data storage has been driven by distributed computing and data sharing which is the result of advances in Internet, network infrastructure and technologies in the past decades [40]. The amount of electronic stored sensitive data, such as health care records, customer records or financial data, increases rapidly every day. Such data has to be shared, replicated, and retained online in order to satisfy various information system requirements such as performance, availability, and recovery. As a result, storage systems are becoming more vulnerable to security breaches, which can result in damaging losses. The goal of storage security is to prevent sensitive data at rest or in transit from being accessed and modified by non-legitimate users or applications.

In this paper we present a survey of security services provided by existing distributed data storage systems and answer the following questions: 1) What are the data security features that should be provided by a distributed storage system? 2) What kind of data security features and techniques are provided by current distributed data storage systems? 3) What are the emerging issues concerning security of distributed data storage systems?

The paper is organized as the following: Section 2 covers distributed data storage infrastructures and systems, security risks and mechanisms are specified in Section 3. Section 4, and 5 addresses the existing data security services and techniques in the current distributed data storage systems based on two types of protection models. Section 6 analyzes the existing protection mechanisms and points out the possible trends. We conclude in Section 7.

## 2. Distributed Data storage System

From a user and application standpoint, distributed data storage systems can be classified as either centralized, decentralized or hybrid storage systems.

In a centralized storage system, data and files are managed by a central component. A uniform interface provides a single point of view to the underlying storage system. In a decentralized storage system, storage servers or devices manage files or data individually. Users have to access individual storage servers or devices to locate or access data. A hybrid storage system separates data access and metadata management from the data I/O path. Data access is through a central component, while data retrieval is through contacting each individual storage server or device. This hybrid approach allows data to be split into data blocks. Those data blocks are stored on different network servers. Simultaneously retrieving data blocks from different network servers achieves maximum throughput.

Protecting distributed data storage presents many challenges, including:

- 1) Data is highly distributed across a network, increasing the management complexity and introducing more points of vulnerability.
- 2) A decentralized system often creates isolated islands of management which are more vulnerable to security breaches.
- 3) Sensitive data stored in distributed storage systems are shared by users and applications that often reside in different security domains, which may have different security policies.
- 4) Legislation and regulations place strict demands on long time data preservation. Extended retention time provides wider time windows for attackers. This in turn raises issues concerning the potential need for backwards compatibility of data migration processes, including encryption algorithm migration.
- 5) Not a single method can protect a distributed data storage system and address all the vulnerabilities.

## 3. Threats and Protection Mechanisms to Distributed Data Storage System

Different distributed data storage systems and infrastructures have different weak points. Understanding threats at

each layer and entry point is essential before determining the right protection strategy.

Distributed Data Storage protection also has trade-offs. For example, encryption can impact performance, usability and data recovery. Data replication providing high availability can open up more entry points for attacks. Storage standards which define interoperability of various storage systems also need to be followed in order to come up with the right countermeasures.

### 3.1 Threats to Distributed Data Storage System

Data security is often considered to have four dimensions: Confidentiality, Integrity, Availability and Authentication (CIAA). Hasan et al [17] classified storage threats and attacks based on this notion of CIAA.

General threats to confidentiality include sniffing storage traffic, snooping on buffer caches and de-allocated memory. File system profiling is an attack that uses access type, timestamps of last modification, file names, and other file system metadata to gain insight about the storage system operation. Storage and backup media may also be stolen in order to access data.

General threats to integrity include storage jamming (a malicious but surreptitious modification of stored data) to modify or replace the original data, metadata modification to disrupt a storage system, and subversion attacks to gain unauthorized OS level access in order to modify critical system data, and man-in-the-middle attacks in order to change data contents in transit.

General threats to availability include (distributed) denial-of-service, disk fragmentation, network disruption, hardware failure and file deletion. Centralized data location management or indexing servers can be points of failure for denial-of-service attacks, which can be launched using malicious code. Long-term data archiving systems present additional challenges such as long-term key management and backwards compatibility, which threaten availability if they are not conducted carefully[42].

General threats to authentication include wrapping attacks to SOAP messages to access unauthorized data, federated authentication using browsers that can possibly open a door to steal authentication tokens, and replay attacks to deceive the system into processing unauthorized operations.

Cloud-based storage and virtualization pose further threats. For example, outsourcing leads to data owners losing physical control of their data, bringing issues of auditing, trust, obtaining support for investigations, accountability, and compatibility of security systems. Multi-tenant virtualization environments can result in applications losing their security context, enabling an adversary to attack other virtual machine instances hosted on the same physical server.

### 3.2 Data Protection Mechanisms

We now briefly review some common data storage protection mechanisms.

Access Control typically includes both authentication and authorization. Centralized and de-centralized access management are two models for distributed storage systems. Both models require entities to be validated against pre-defined policies before accessing sensitive data. These access privileges need to be periodically reviewed or re-granted.

Encryption is the standard method for providing confidentiality protection. This relies on the necessary encryption keys being carefully managed, including processes for flexible key sharing, refreshing and revocation. In distributed environments secret sharing mechanisms can also be used to split sensitive data into multiple component shares.

Storage integrity violation can either be accidental (from hardware/software malfunctions) or malicious attacks [39]. Accidental modification of data is typically protected by mirroring, or the use of basic parity or erasure codes. The detection of unauthorized data modification requires the use of Message Authentication Codes (MACs) or digital signature schemes. The latter provides the stronger notion of non-repudiation, which prevents an entity from successfully denying unauthorized modification of data.

Data availability mechanisms include replication and redundancy. Recovery mechanisms may also be required in order to repair damaged data, for example re-encrypting data when a key is lost. Intrusion Detection or Prevention mechanisms detect or prevent malicious activities that could result in theft or sabotage of data. Audit logs can also be used to assist recovery, provide evidence for security breaches, as well as being important for compliance.

Most of these security mechanisms incur extra overheads. A storage system thus needs to be designed in such a way that appropriate trade-offs are made between security, usability, flexibility, manageability, scalability and performance. While the provision of security is important, in many systems the full costs of strong security may result in impractical design approaches to distributed storage protection.

Kher et al. [27] presented a comprehensive survey of the security services of existing storage systems in 2005. Since then, many distributed data storage systems have emerged and advanced with the help of new technologies and mechanisms.

Data protection schemes in distributed storage systems can be generally classified into two major categories: storage centric and user centric protection. In storage centric protection, the storage system takes responsibility for data protection. Users and network connections are untrusted parties and data access is centrally managed. In user centric protection, data owners take responsibility for protecting their own data. Storage servers, archive systems and other users are assumed to be untrusted parties. This results in a decentralized protection model which requires end-to-end

protection mechanisms.

In the following two sections, we will examine a number of representative storage systems, their security features and their vulnerabilities.

## 4. Storage Centric Data Protection

In this section we examine two representatives of storage centric data protection.

### 4.1 Network Attached Storage Devices

Network Attached Storage Devices (NASD) [13] is a distributed file system that attaches the storage devices directly to the network. The NASD architecture changes the server's role from being actively involved in every request to a management role of providing high-level application-specific semantics to clients. The server (file manager) is responsible for defining policy with regard to who can access storage as well as adding high-level functions such as cache consistency and namespace management. Directly attached disks are no longer be hidden behind the server and thus must rely on their own security rather than the server's protection. EMC's NAS and SAN platforms product lines [21], HP StorageWorks [19], FreeNAS [18] are examples of NASD.

NASD uses a cryptographic capability-based access control model [15][11][14] with three parties: file manager, storage device and clients. The file manager is the central component for authenticating users, granting access rights to requested data operations, and issuing capabilities. The file manager maintains access control lists and a sets of unique symmetric keys that are shared with every storage device. Users authenticate themselves to the file manager to obtain capabilities containing authentication information and access privileges for the requested operations. The capabilities are presented to storage devices by users. Storage devices validate the capabilities before any requests can be fulfilled.

NASD assumes that the file manager and storage devices are trusted, while users and network connections are not trusted. The capabilities are subjected to replay, hijacking and man-in-the-middle attacks if they are not protected through secure channels. The file manager is a potential central point of failure. Data or metadata privacy and integrity protection at rest and in transit are not addressed. Availability and recovery of the data are not provided by NASD either, with data owners expected to take appropriate responsibility.

### 4.2 Object-based Storage Devices

Object-based Storage Devices (OSDs) [23] divides files into data blocks and wrap each data block into an object. The system consists of two logical components: a metadata server and OSDs. The metadata server maintains the metadata of files, which includes the objects' metadata and locations. Objects are stored on storage devices. Examples of commercial

and open source products are Ceph [44], Panas[45], Lustre [8] and IBM ObjectStore [10].

The separation of data and metadata management provides two benefits. It enables the ability of simultaneous object access from different storage devices, which dramatically increases I/O throughput and improves performance to large data file retrievals. The management separation also enables the OSD security model to separate policy from enforcement. Policy is managed and executed by the metadata server. Policy enforcement is conducted by each individual OSD.

The data protection mechanism provided by OSDs is the capability-based authentication protocol OSD T10 [35]. The protocol defines four aspects of access protection: 1) authentication and capability acquirement; 2) capability protection and validation; 3) management of the secret keys shared between security manager and OSDs; 4) commands used for object requests. A client first authenticates to the metadata server. Next, the server issues a capability according to the access policy. The client uses the capability to contact each OSD for the requested data. Each OSD validates the capability with the request, ensuring that: 1) the capability has not been tampered with and is rightfully obtained by the client, and 2) the requested operation is permitted by the capability.

The secret keys shared between metadata server and OSDs are managed and specified by a hierarchical key structure in OSD T10. OSD T10 also defines four security methods to be used, based on the level of protection. Several OSDs have implemented OSD T10 protocols, which include OpenSolaris T10 OSD Project [22] and the DISC OSD T10 implementation [20].

Sharing secret keys between the metadata server and OSDs has limitations. A compromised or lost key not only needs a replacement key to be generated, but the impact can ripple through the hierarchical key structure and force refresh of lower-level keys. Any key refreshment can further invalidate the capabilities protected by those keys. Since the OSD T10 protocol issues a capability for every object access, it cannot scale well due to the overhead of network communications and capability management.

To address the overhead of transferring large amounts of capabilities in terabyte or petabyte file storage systems, Leung at al [30] introduced three modifications: 1) making a capability to authorize a user instead of authorizing each user and file pair, reducing the number of required capabilities; 2) automatic revocation to shorten the capability lifetime and enable automatic capability renewal process; 3) secure delegation to allow users acting on behalf of a group to open files. However when terabyte or petabyte file systems have large numbers of OSDs, key management becomes complex, especially when a key is stolen or comprised.

Role-based access control was proposed by Kher and Kim [26] to reduce the amount of capabilities. Capabilities

are generated based on roles instead of individual users. However the authorization decisions now have to be made at OSDs instead of at the central metadata server. OSDs also have to store the entire role-based ACL for each object, which in addition introduces the possible access policy synchronization among OSDs issue.

Other than capability-based access control, OSDs do not provide data confidentiality protection at rest. Insider and intrusion attacks incur high risks for sensitive data stored in the clear on OSDs. Most of the systems use network layer protocols to protect data in transit.

## 5. User Centric Data Protection Systems

In this section we examine two types of user centric storage systems.

### 5.1 Cryptographic Storage System

Symmetric encryption is used in many systems, such as SiRiUS[24], and CRUST [12]. SiRiUS provides file-level encryption. File owners are responsible for file encryption, key distribution and access policy specification. The key distribution of SiRiUS does not scale well as a file encryption key has to be wrapped to users' public keys. Data integrity protection is provided by using hash trees.

CRUST was designed to eliminate the key distribution and scalability issues of SiRiUS. CRUST used the Leighton-Micali key pre-distribution scheme [29]. However it requires each user to share a long-term key encryption key with every other user, which results in multiple key management issues.

Miller et al [34] developed a scheme for data secrecy and integrity protection on NASD. They used a similar approach to SiRiUS, but the encryption is at the data block level. An encryption key is encrypted by each legitimate user's public key and stored in a key object associated with the file on the metadata server.

Wrapping the encryption key into users' public keys creates a user access right revocation issue. There are three possible solutions: 1) simply remove the user from the key object (which wraps the encryption key in users' public key for encryption key distribution); however the user may still cache the encryption key and be able to read the data; 2) immediately re-encrypt the file with a new encryption key and encrypt the new key with the public keys of those users who should still have access to the file, which is slower, but will ensure that the revoked user cannot access the file; 3) apply the second solution lazily (lazy revocation); although the revoked user continues to have access to the old data, this prevents his/her access to any new data that is encrypted with a different key. Integrity protection is achieved through a non-linear check-sum of the unencrypted data which is attached to the encrypted data.

Kher [27] and Storer et al. [42] provide detailed surveys of other encryption file systems such as NCryptfs, Microsoft EFS, Plutus, Cepheus, etc.

Recently, Attribute Based Encryption (ABE), a type of asymmetric key encryption, has been applied to address fine-grained data access control and privacy protection. Introduced by Sahai and Waters in [38], ABE extended Identity Based Encryption (IBE) to design flexible and scalable access control systems. There are two kinds of ABE: key-policy ABE (KP-ABE) [16] and ciphertext-policy ABE (CP-ABE) [5] [7]. KP-ABE is a per-key based access control. In KP-ABE, the ciphertext is associated with a set of attributes and the secret key is associated with the access policy. The encryptor defines the set of descriptive attributes necessary to decrypt the ciphertext. The trusted authority who generates user's secret key defines the combination of attributes for which the secret key can be used. In CP-ABE, the idea is reversed: the ciphertext is associated with the access policy and the encrypting party determines the policy under which the data can be decrypted. The secret key is now associated with a set of attributes. Therefore CP-ABE is a per-message based access control. In order to address privacy of the access control policy, anonymous ABE was introduced and further improved by [36]. User accountability and illegal key sharing are addressed in [31].

Secret sharing schemes provides data secrecy protection without encrypting the data. Lakshmanan, et al. [28] proposed a distributed store that uses secret sharing to provide confidentiality at rest. Secret share replication is used to improve performance and provide availability. A dissemination protocol is used by servers to propagate new data shares among replication servers. A share renewal protocol is used to periodically generate new shares for long-term confidentiality. The recoverability of secret sharing schemes is leveraged by POTSHARDS [41] to protect data over indefinitely long period of time. Approximate pointers in conjunction with secure distributed RAID techniques are used for availability and reliability. Other storage systems such as PASIS, CleverSafe and GridSharing also use secret sharing for data protection [42].

### 5.2 Cloud-based Storage

Cloud-based storage is gaining rapid interest. The uniqueness of cloud storage over traditional or object-based storage is its ability to leverage virtualization techniques to provide a storage service composed of thousands of networked storage devices, distributed file systems, and storage middleware. This enables on-demand service, capacity, and management to users anywhere via the Internet. There are many cloud storage service providers, such as IBM, Amazon (S3), Google (GFS), Microsoft (Azure), EMC (Atmos), open source CloudStore, HDFS, etc. However the common standards for cloud service are still in development.

In most existing cloud storage systems, data in transit is protected through network layer security, such as SSL/TLS. User access control is provided through authentication and access control lists. Data privacy protection is not typically

provided by the service provider.

By outsourcing data into the cloud, data owners physically release their information to external servers that are not under their control. Confidentiality and integrity are thus put at risk. Most of time users will not know how data is maintained, transferred, backed-up and replicated. In addition to safeguarding confidential data from attackers and unauthorized users, there is a need to protect the privacy of the data from so-called honest-but-curious servers, which may be trusted to properly manage the data, but not to read data content. As well as where it is stored, the locations where data is used and transferred needs to be considered.

Good encryption and key distribution can automatically enforce an access control policy. Some research has been conducted into using key access control to govern data access control. [6] proposed a framework to allow different users to view different parts of some data based on their privileges. [9] proposed a selective encryption scheme to encrypt data with different keys and assign each user a set of keys necessary to decrypt all or partial authorized resources. In order to reduce the number of shared secret keys and achieve more efficient and scalable key management, a key derivation graph was used to derive new keys by combining existing ones and public tokens. This model requires authorization be able to form a hierarchical access structure which is more suitable to database tables and files.

Data integrity and reliability are the other two biggest concerns for storing data on untrusted servers or archive systems. One of the main issues is to frequently, efficiently and securely verify that a storage server is faithfully storing clients' data. Research has been done on how to enable data owners to periodically perform integrity verification on untrusted servers without their local copy of data files. Generally speaking, there are two models: Provable Data Possession (PDP) [3][4] and Proof of Retrievability (POR) [25]. Based on the role of the verifier, all schemes presented so far fall into two categories: private and public verifiability. Although schemes with private verifiability can achieve higher efficiency, public verifiability allows anyone, not just the client (data owner), to challenge the cloud server for correctness of data storage without any private information.

PDP demonstrates to a client that a server possesses a file but it is weaker than POR since it does not guarantee that the client can retrieve the file. POR is a compact proof to demonstrate to a client that a target file is intact or that the client can fully recover it under certain conditions. Juels and Kaliski in [25] proposed a POR scheme and protocols by combining cryptographic sentinels for spot checking and error-correction encoding to ensure both data possession and retrievability in storage systems. However Juels and Laliski's POR does not address file updates with block modification, insertion and deletion. In addition, the number of queries is limited and public verifiability is not supported. Wang et al [43] extended the public verifiability model and enhanced

the scheme to allow fully dynamic data updates.

## 6. Discussions

Based on the security features and techniques in current storage systems, some possible trends have become clear.

### 6.1 Distributed Storage System Protection Models

In a distributed storage systems, user centric protection is often adopted if the storage system is assumed to be untrusted. Data owners take the responsibility of protecting their data, hence encryption keys need to be managed by data owners. When data is shared with other users, key distribution and data access control become challenges. Auditing and policy enforcement create further challenges to data owners.

Centralized storage systems tend to use storage centric protection. However this model has a central point of failure and most existing storage systems do not provide data at rest protection.

### 6.2 Long-term Data Protection

Encryption for data at rest results in potential long-term data retention issues. While most current encryption algorithms are designed to provide security over an extended time, long-term data retention does provide greater time windows for attackers to operate. The main challenges arise from long-term key management, due to potential unavailability of key owners, migrations issues, key losses, etc.

### 6.3 Secret Sharing

Secret sharing mechanisms provide an alternative means of providing data at rest protection in distributed systems. However shared components need careful management (as for keys, they may need renewed or refreshed) and secret sharing involves extra storage and network overheads. The practicality of secret sharing mechanisms for large data file protection thus needs careful consideration.

### 6.4 Cloud-based Storage Systems

Cloud-based storage systems provide massive capacity and high performance, but sensitive data protection in the cloud is still in its infancy. With the uncertainty of how data is stored and transferred within a cloud, data owners have to take the responsibility of protecting their own data. While security concerns for stored data in cloud-based storage have much in common with those associated with untrusted file servers, cloud-based storage differs in their persistence and availability.

As users no longer physically possess the storage for their data, some applications require cloud storage providers to become the middle man for data access and transfer. The potential consequences of outsourcing data protection responsibility and trust management to third party vendors

requires further investigation. The ability to verify the correctness of the data in a cloud environment can be formidable and expensive to cloud users [1]. The notion of public auditability has been proposed in the context of ensuring remotely stored data integrity with different systems and security models. However, most of these schemes do not support the privacy protection of users' data against external auditors. How to efficiently audit and provide data secrecy at the same time also requires further study.

Resource pooling with location independence, rapid elasticity, and on-demand self-service are three out of the five essential cloud characteristics [33]. The highly distributed and dynamic storage environment requires a security service to be highly flexible and configurable. On-demand security control and configurable security features are thus highly desirable.

## 6.5 Virtualization

With the increasing complexity of massive storage management and data sharing in heterogeneous environments, more efficient and intelligent storage systems are in demand. Zeng et al. [46] proposed a virtual storage architecture to integrate heterogeneous storage systems and abstract their management, collaboration and interaction. With more demands on massive and high capacity storage, we believe that heterogeneous storage systems will converge in term of usability. Also, as Zeng pointed out in [46], "network is storage and storage is the network".

Data protection becomes even more important as collaboration is required amongst mutually untrusted distributed storage systems. The appearance of Fabric, a new system and language to build secure distributed information systems [32], has indicated the need of federate storage systems to share computational resources across various security domains. We believe federate storage systems with different storage types across mutual distrust domains will eventually emerge. Their security management, data transformation, auditing, usability and scalability will be yet another research area.

## 6.6 Optimization

Reliability and availability of storage systems are implemented through redundancy. Sometimes this can introduce significant system overheads. Data de-duplication [2] removes high redundancy among files or data blocks, cuts storage capacity requirements, reduces network traffic, and improves performance.

The main challenges of data protection in de-duplication are integrity, data segmentation and privacy. De-duplication's breaking of files into chunks/segments/blocks causes data integrity concerns. It can also erase the boundaries of data or file groupings. Encrypting the same data with different keys will generate different ciphertexts. Even if there is only one key, management issues, such as key refreshing, can cause

severe problems because each data block can come from different file version.

De-duplication systems often use meta data, index trees and hash databases to detect and locate chunks/segments/blocks in storage. Such information needs to be properly protected against different threats. Data de-duplication on encrypted data still remains challenging.

Secure keyword searching or pattern matching received attention [37], but still remains challenging. Consistency issues of access right, especially the copies cached on the client side, has not received enough attention.

## 7. Conclusions

This paper presents a survey of security features and techniques in existing distributed storage systems. We classified distributed storage systems into three categories: centralized, distributed and hybrid. We then listed the threats and protection mechanisms and identified two protection models (user centric and storage centric) for distributed storage systems. We then examined several distributed storage systems and identified emerging issues. In reality, it will be difficult (impossible) to build a distributed storage system that can satisfy all the potential requirements of the environments in which these systems are needed. However it is hoped that this discussion has helped to raise awareness of the challenges and potential solutions that can be applied in order to incorporate security into systems of this type.

## References

- [1] Cloud Security Alliance. <http://www.cloudsecurityalliance.org>. Security guidance for critical areas of focus in cloud computing.
- [2] Lior Aronovich, Ron Asher, Eitan Bachmat, Haim Bitner, Michael Hirsch, and Shmuel T. Klein. The design of a similarity based deduplication system. In *SYSTOR '09*, 2009.
- [3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In *14th ACM conference on Computer and communications security*, page 598-609. ACM, 2007.
- [4] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik. Scalable and efficient provable data possession. In *SecureComm'08*, 2008.
- [5] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy - S&P*, pages 321-334, 2007.
- [6] C. Blundo, S. Cimato, S. De Capitani di Vimercati, A. De Santis, S. Foresti, S. Paraboschi, and P. Samarati. Efficient key management for enforcing access control in outsourced scenarios. In *IFIP Advances in Information and Communication Technology*, volume 297, pages 364-375. Springer Boston, 2009.
- [7] Ling Cheung and Calvin Newport. Provably secure ciphertext policy attribute. In *the 14th ACM conference on Computer and communications security*, pages 456 - 465. ACM, 2007.
- [8] Inc Cluster File Systems. Lustre: A scalable high performance file system. White paper, Jan 2002.
- [9] E. Damiani, S. De Capitani di Vimercati, S. Jajodia, S. Foresti, S. Paraboschi, and P. Samarati. Selective data encryption in outsourced dynamic environments. In *VODCA*, 2006.
- [10] M. Factor, K. Meth, D. Naor, O. Rodeh, and J. Satran. Object storage: The future building block for storage systems: A position paper. In *the Second International IEEE Symposium on Emergence of Globally Distributed Data*, page 119-123, 2005.

- [11] Michael Factor, Dalit Naor, Eran Rom, Julian Satran, and Sivan Tal. Capability based secure access control to networked storage devices. In *24th IEEE Conference on Mass Storage Systems and Technologies*, pages 114–128. IEEE Computer Society, 2007.
- [12] Erel Geron and Avishai Wool. Crust: Cryptographic remote untrusted storage without public keys. In *the Fourth International IEEE Security in Storage Workshop*, page 3Ü14, 2007.
- [13] Garth A. Gibson, David F. Nagle, Khalil Amiri, Fay W. Chang, Eugene M. Feinberg, Howard Gobiuff, Chen Lee, Berend Ozceri, Erik Riedel, David Rochberg, and Jim Zelenka. File server scaling with network-attached secure disks. In *the 1997 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 1997.
- [14] H. Gobio. Security for a high performance commodity storage subsystem. Master's thesis, Carnegie Mellon University, 1999.
- [15] Howard Gobiuff, Garth Gibson, and Doug Tygar. Security for network attached storage devices. 1997.
- [16] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Computer and Communications Security - CCS*, 2006.
- [17] Hasan, Myagmar, Lee, and Yurcik. Toward a threat model for storage systems. In *Workshop On Storage Security And Survivability*, pages 94 – 102, 2005.
- [18] <http://en.wikipedia.org/wiki/FreeNAS>. Freenas.
- [19] <http://h18006.www1.hp.com/storage/networking/index.html>. Storage networking.
- [20] [http://sourceforge.net/projects/disc\\_osd/](http://sourceforge.net/projects/disc_osd/). Disc osd t10 implementation.
- [21] [http://whatis.bitpipe.com/plist/Network Attached-Storage.html](http://whatis.bitpipe.com/plist/Network%20Attached-Storage.html). Nas products.
- [22] <http://www.dtc.umn.edu/disc/resources/CoverstonISW5.pdf>. Opensolaris t10 osd reference implementation.
- [23] Sami Iren and Rich Ramos. Object-based storage (osd) architecture and systems. <http://www.snia.org>, Sept 2007.
- [24] Eu jin Goh, Hovav Shacham, Nagendra Modadugu, and Dan Boneh. Sirius: Securing remote untrusted storage. In *Tenth Network and Distributed Systems Security (NDSS) Symposium*, pages 131–145, Feb 2003.
- [25] A. Juels and Jr. B. S. Kaliski. Pors: proofs of retrievability for large files. In *14th ACM conference on Computer and communications security*, page 584Ü597. ACM, 2007.
- [26] Vishal Kher and Yongdae Kim. Decentralized authentication mechanisms for object-based storage devices. In *Security in Storage Workshop, International IEEE*, volume 0, page 1, 2003.
- [27] Vishal Kher and Yongdae Kim. Securing distributed storage: Challenges, techniques, and systems. In *StorageSS*, pages 9 – 25. ACM, 2005.
- [28] S. Lakshmanan, M. Ahamad, and H Venkateswaran. Responsive security for stored data. In *the 23rd International Conference on Distributed Computing Systems*, page 146, 2003.
- [29] Frank Thomson Leighton and Silvio Micali. Secret-key agreement without public-key cryptography. In *the 13th Annual International Cryptology Conference on Advances in Cryptology*, pages 456 – 479, 1993.
- [30] Andrew Leung, Ethan L. Miller, and Stephanie Jones. Scalable security for petascale parallel file systems. In *ACM/IEEE conference on Supercomputing*. ACM, 2007.
- [31] Jin Li, Kui Ren, Bo Zhu, and Zhiguo Wan. Privacy-aware attribute-based encryption with user accountability. In *the 12th International Conference on Information Security*, pages 347 – 362, 2009.
- [32] Jed Liu, Michael D. George, K. Vikram, Xin Qi, Lucas Wayne, and Andrew C. Myers. Fabric: A platform for secure distributed computation and storage. In *ACM 2009 Symposium on Operating Systems Principles and Implementation*, page 321Ü334, 2009.
- [33] Peter Mell and Tim Grance. Effectively and securely using the cloud computing paradigm. NIST, Information Technology Laboratory, Oct 2009.
- [34] E. Miller, W. Freeman, D. Long, and B. Reed. Strong security for network-attached storage. In *the 2002 Conference on File and Storage Technologies (FAST)*, pages 1–13, Jan 2002.
- [35] D. Nagle, M. E. Factor, S. Iren, D. Naor, E. Riedel, O. Rodeh, and J. Satran. The ansi t10 object-based storage standard and current implementations. In *IBM Journal of Research and Development*, volume 52 , Issue 4, pages 401–411. IBM Corp., July 2008.
- [36] Takashi Nishide, Kazuki Yoneyama, and Kazuo Ohta. Abe with partially hidden encryptor-specified access structure. In *ACNS*, pages 111–129. Springer, 2008.
- [37] G. Wang Q. Liu and J. Wu. An efficient privacy preserving keyword search scheme in cloud computing. In *International Conference on Computational Science and Engineering*, 2009.
- [38] A. Sahai and B. Waters. Fuzzy identity based encryption. In *Theory and Application of Cryptographic Techniques - EUROCRYPT*, 2005.
- [39] Gopalan Sivathanu, Charles P. Wright, and Erez Zadok. Ensuring data integrity in storage: techniques and applications. In *In The 1st International Workshop on Storage Security and Survivability*, pages 26–36. ACM, 2005.
- [40] Ken Smith, Len Seligman, and Vipin Swarup. Everybody share: The challenge of data-sharing systems. In *Computer*, pages 54 – 61. the IEEE Computer Society, August 2008.
- [41] STORER, GREENAN, MILLER, and VORUGANTI. PotshardsÜa secure, recoverable, long-term archival storage system. In *ACM Transactions on Storage (TOS)*, volume 5. ACM, Jun 2009.
- [42] Storer, Greenan, and Ethan L. Miller. Longterm threats to secure archives. In *StorageSS, Storage Security And Survivability*, pages 9–16. ACM, October 2006.
- [43] Qian Wang, Cong Wang, Jin Li, Kui Ren, and Wenjing Lou. Enabling public verifiability and data dynamics for storage security in cloud computing. In *ESORICS*, pages 355–370, 2009.
- [44] Sage Weil, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long, and Carlos Maltzahn. Ceph: A scalable, high-performance distributed file system. In *7th Conference on Operating Systems Design and Implementation (OSDI '06)*, pages 307–320, Nov 2006.
- [45] Brent Welch, Marc Unangst, Zainul Abbasi, Garth Gibson, Brian Mueller, Jason Small, Jim Zelenka, and Bin Zhou. Scalable performance of the panasas parallel file system. In *FAST '08: 6th Usenix Conference on File and Storage Technologies*, page 17Ü33, 2008.
- [46] Wenying Zeng, Yuelong Zhao, Wenfeng Wang, and Wei Song. Intelligent storage architecture and key technologies research. In *CSIE 2009, 2009 WRI World Congress on Computer Science and Information Engineering*, volume 7, pages 778–782. IEEE Computer Society, 2009.

# Defining and Assessing Quantitative Security Risk Measures Using Vulnerability Lifecycle and CVSS Metrics

HyunChul Joh<sup>1</sup>, and Yashwant K. Malaiya<sup>1</sup>

<sup>1</sup>Computer Science Department, Colorado State University, Fort Collins, CO 80523, USA

**Abstract** - *Known vulnerabilities which have been discovered but not patched represents a security risk which can lead to considerable financial damage or loss of reputation. They include vulnerabilities that have either no patches available or for which patches are applied after some delay. Exploitation is even possible before public disclosure of a vulnerability. This paper formally defines risk measures and examines possible approaches for assessing risk using actual data. We explore the use of CVSS vulnerability metrics which are publically available and are being used for ranking vulnerabilities. Then, a general stochastic risk evaluation approach is proposed which considers the vulnerability lifecycle starting with discovery. A conditional risk measure and assessment approach is also presented when only known vulnerabilities are considered. The proposed approach bridges formal risk theory with industrial approaches currently being used, allowing IT risk assessment in an organization, and a comparison of potential alternatives for optimizing remediation. These actual data driven methods will assist managers with software selection and patch application decisions in quantitative manner.*

**Keywords** - Security vulnerabilities; Software Risk Evaluation; CVSS; Vulnerability lifecycle

## 1 Introduction

To ensure that the overall security risk stays within acceptable limits, managers need to measure risks in their organization. As Lord Calvin stated “If you cannot measure it, you cannot improve it,” quantitative methods are needed to ensure that the decisions are not based on subjective perceptions only.

Quantitative measures have been commonly used to measure some attributes of computing such as performance and reliability. While quantitative risk evaluation is common in some fields such as finance [1], attempts to quantitatively assess security are relatively new. There has been criticism of the quantitative attempts of risk evaluation [2] due to the lack of data for validating the methods. Related data has now begun to become available. Security vulnerabilities that have been discovered but remain unpatched for a while represent considerable risk for an organization. Today online banking, stock market trading, transportation, even military and governmental exchanges depend on the Internet based computing and communications. Thus the risk to the society due to the exploitation of vulnerabilities is massive. Yet peo-

ple are willing to take the risk since the Internet has made the markets and the transactions much more efficient [3]. In spite of the recent advances in secure coding, it is unlikely that completely secure systems will become possible anytime soon [4]. Thus, it is necessary to assess and contain risk using precautionary measures that are commensurate.

While sometimes risk is informally stated as the possibility of a harm to occur [5], formally, risk is defined to be a weighted measure depending on the consequence. For a potential adverse event, the risk is stated as [6]:

$$\text{Risk} = \text{Likelihood of an adverse event} \times \text{Impact of the adverse event} \quad (1)$$

This presumes a specific time period for the evaluated likelihood. For example, a year is the time period for which *annual loss expectancy* is evaluated. Equation (1) evaluates risk due to a single specific cause. When statistically independent multiple causes are considered, the individual risks need to be added to obtain the overall risk. A *risk matrix* is often constructed that divides both likelihood and impact values into discrete ranges that can be used to classify applicable causes [7] by the degree of risk they represent.

In the equation above, the likelihood of an adverse event is sometimes represented as the product of two probabilities: probability that an exploitable weakness is present, and the probability that such a weakness is exploited [7]. The first is an attribute of the targeted system itself whereas the second probability depends on external factors, such as the motivation of potential attackers. In some cases, the *Impact of the adverse event* can be split into two factors, the technical impact and the business impact [8]. Risk is often measured conditionally, by assuming that some of the factors are equal to unity and thus can be dropped from consideration. For example, sometimes the external factors or the business impact is not considered. If we would replace the impact factor in Equation (1) by unity, the conditional risk simply becomes equal to the probability of the adverse event, as considered in the traditional reliability theory. The conditional risk measures are popular because it can alleviate the formidable data collections and analysis requirements. As discussed in section 4 below, different conditional risk measures have been used by different researchers.

A *vulnerability* is a software defect or weakness in the security system which might be exploited by a malicious user causing loss or harm [5]. A stochastic model [9] of the vulnerability lifecycle could be used for calculating the *Likelihood of an adverse event* in Equation (1) whereas impact related metrics from the Common Vulnerability Scoring Sys-

tem (CVSS) [10] can be utilized for estimating *Impact of the adverse event*. While a preliminary examination of some of the vulnerability lifecycle transitions has recently been done by researchers [11][12], risk evaluation based on them have been received little attention. The proposed quantitative approach for evaluating the risk associated with software systems will allow comparison of alternative software systems and optimization of risk mitigation strategies.

The paper is organized as follows. Section 2 introduces the risk matrix. Section 3 discusses the CVSS metrics that are now being widely used. The interpretation of the CVSS metrics in terms of the formal risk theory is discussed in section 4. Section 5 introduces the software vulnerability lifecycle and the next section gives a stochastic method for risk level measurement. Section 7 presents a conditional risk assessing method utilizing CVSS base scores which is illustrated by simulated data. Finally, conclusions are presented and the future research needs are identified.

## 2 Risk matrix: scales & Discretization

In general, a system has multiple weaknesses. The risk of exploitation in each weakness  $i$  is given by Equation (1). Assuming that the potential exploitation of a weakness is statistically independent of others, the system risk is given by the summation of individual risk values:

$$\text{System Risk} = \sum_i L_i \times I_i \quad (2)$$

where  $L_i$  is the likelihood of exploitation of weakness  $i$  and  $I_i$  is the corresponding impact. A risk matrix provides a visual distribution of potential risks [13][14]. In many risk evaluation situations, a risk matrix is used, where both impact and likelihood are divided into a set of discrete intervals, and each risk is assigned to likelihood level and an impact level. Impact can be used for the x-axis and likelihood can be represented using then y-axis, allowing a visual representation of the risk distribution. For example, the ENISA Cloud Computing report [15] defines five impact levels from *Very Low* to *Very High*, and five likelihood levels from *Very Unlikely* to *Frequent*. Each level is associated with a *rating*. A risk matrix can bridge quantitative and qualitative analyses. Tables have been compiled that allow on to assign a likelihood and an impact level to a risk, often using qualitative judgment or a rough quantitative estimation.

The scales used for likelihood and impact can be linear, or more often non-linear. In Homeland Security's RAMCAP (Risk Analysis and Management for Critical Asset Protection) approach, a logarithmic scale is used for both. Thus, 0-25 fatalities is assigned a rating "0", while 25-50 is assigned a rating of "1", etc. For the likelihood scale, probabilities between 0.5-1.0 is assigned the highest rating of "5", between 0.25-0.5 is assigned rating "4", etc.

Using a logarithmic scale for both has a distinct advantage. Sometimes the overall rating for a specific risk is found by simply adding its likelihood and impact ratings. Thus, it would be easily explainable if the *rating* is proportional to the logarithm of the absolute value. Consequently, Equation (1) can be re-written as:

$$\begin{aligned} \log(\text{risk}_i) &= \log(\text{likelihood}_i) + \log(\text{impact}_i) \\ \text{rating\_Risk}_i &= \text{rating\_likelihood}_i + \text{rating\_impact}_i \end{aligned} \quad (3)$$

When a normalized value of the likelihood, impact or the risk is used, it will result in a positive or negative constant added to the right hand side. In some cases, higher resolution is desired in the very high as well as very low regions; in such cases a suitable non-linear scale such as using the *logit* or *log-odds* function [16] can be used.

The main use of risk matrices is to rank the risks so that higher risks can be identified and mitigated. For determining ranking, the rating can be used instead of the raw value. Cox [7] has pointed out that the discretization in a risk matrix can potentially result in incorrect ranking, but risk matrices are often used for convenient visualization. It should be noted that the risk ratings are not additive.

We will next examine the CVSS metrics that has emerged recently for software security vulnerabilities, and inspect the relationship (likelihood, impact) in risk and (exploitability, impact) in CVSS vulnerability metric system.

## 3 CVSS metrics and Related Works

Common Vulnerability Scoring System (CVSS) [10] has now become almost an industrial standard for assessing the security vulnerabilities although some alternatives are sometimes used. It attempts to evaluate the degree of risks posed by vulnerabilities, so mitigation efforts can be prioritized. The measures termed *scores* are computed using assessments (called *metrics*) of vulnerability attributes based on the opinions of experts in the field. Initiated in 2004, now it is in its second version released in 2007.

The CVSS scores for known vulnerabilities are readily available on the majority of public vulnerability databases on the Web. The CVSS score system provides vendor independent framework for communicating the characteristics and impacts of the known vulnerabilities [10]. A few researchers have started to use some of the CVSS metrics for their security risk models.

CVSS defines a number of metrics that can be used to characterize a vulnerability. For each metric, a few qualitative levels are defined and a numerical value is associated with each level. CVSS is composed of three major metric groups: Base, Temporal and Environmental. The Base metric represents the intrinsic characteristics of a vulnerability, and is the only mandatory metric. The optional Environmental and Temporal metrics are used to augment the Base metrics, and depend on the target system and changing circumstances. The Base metrics include two sub-scores termed *exploitability* and *impact*. The Base score formula [10], as shown in Equation (4), is chosen and adjusted such that a score is a decimal number in the range [0.0, 10.0]. The value for  $f(\text{Impact})$  is zero when *Impact* is zero otherwise it has the value of 1.176.

$$\begin{aligned} \text{Base score} &= \text{Round to 1 decimal}\{ \\ &[(0.6 \times \text{Impact}) + (0.4 \times \text{Exploitability}) - 1.5] \times f(\text{Impact}) \} \end{aligned} \quad (4)$$

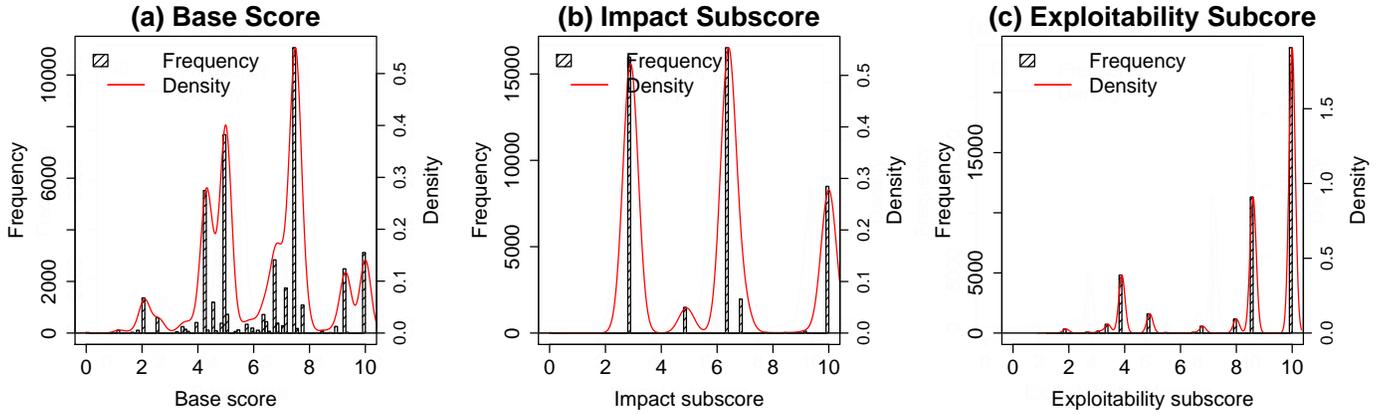


Figure 1. Distributions for CVSS base metric scores (100 bins); NVD [17] on JAN 2011 (44615 vuln.)

The formula for Base score in Equation (4) has not been formally derived but has emerged as a result of discussions in a committee of experts. It is primarily intended for ranking of vulnerabilities based on the risk posed by them. It is notable that the Exploitability and Impact sub-scores are added rather than multiplied. One possible interpretation can be that the two sub-scores effectively use a logarithmic scale, as given in Equation (3). Then possible interpretation is that since the Impact and Exploitability sub-scores have a fairly discrete distribution as shown in Fig. 1 (b) and (c), addition yields the distribution, Fig 1 (a), which would not be greatly different if we had used a multiplication. We have indeed verified that using  $Impact \times Exploitability$  yields a distribution extremely similar to that in Fig. 1 (a). We have also found that multiplication generates about twice as many combinations with wider distribution, and it is intuitive since it is based on the definition of risk given in Equation (1).

The Impact sub-score measures how a vulnerability will impact an IT asset in terms of the degree of losses in confidentiality, integrity, and availability which constitute three of the metrics. Below, in our proposed method, we also use these metrics. The Exploitability sub-score uses metrics that attempt to measure how easy it is to exploit the vulnerability. The Temporal metrics measure impact of developments such as release of patches or code for exploitation. The Environmental metrics allow assessment of impact by taking into account the potential loss based on the expectations for the target system. Temporal and Environmental metrics can add additional information to the two sub-scores used for the Base metric for estimating the overall software risk.

A few researchers have started to use the CVSS scores in their proposed methods. Mkpong-Ruffin et al. [17] use CVSS scores to calculate the loss expectancy. The average CVSS scores are calculated with the average growth rate for each month for the selected functional groups of vulnerabilities. Then, using the growth rate with the average CVSS score, the predicted impact value is calculated for each functional group. Houmb et al. [19] have discussed a model for the quantitative estimation of the security risk level of a system by combining the frequency and impact of a potential unwanted event and is modeled as a Markov process. They estimate frequency and impact of vulnerabilities using reorganized original CVSS metrics. And, finally, the two estimated measures are combined to calculate risk levels.

## 4 Defining conditional risk measures

Researchers have often investigated measures of risk that seem to be defined very differently. Here we show that they are conditional measures of risk and can be potentially combined into a single measure of total risk. The likelihood of the exploitation of a vulnerability depends not only on the nature of the vulnerability but also how easy it is to access the vulnerability, the motivation and the capabilities of a potential intruder.

The likelihood  $L_i$ , in Equation (2), can be expressed in more detail by considering factors such as probability of presence of a vulnerability  $v_i$  and how much exploitation is expected as shown below:

$$\begin{aligned} L_i &= \Pr\{v_i\} \times \Pr\{Exploitation \mid v_i\} \\ &= \Pr\{v_i\} \times \Pr\{V_i \text{ is exploitable} \mid v_i\} \times \\ &\quad \Pr\{v_i \text{ is accessible} \mid v_i \text{ exploitable}\} \times \\ &\quad \Pr\{v_i \text{ externally exploited} \mid v_i \text{ accessible \& exploitable}\} \\ &= L_{Ai} \times L_{Bi} \times L_{Ci} \times L_{Di} \end{aligned}$$

where  $L_{Bi}$  represents the inherent exploitability of the vulnerability,  $L_{Ci}$  is the probability of accessing the vulnerability, and  $L_{Di}$  represents the external factors. The impact factor,  $I_i$ , from Equation (1) can be given as:

$$\begin{aligned} I_i &= \sum_j \Pr\{Security \text{ attribute } j \text{ compromised for } v_i\} \times \\ &\quad \{Expected \text{ cost of } j \text{ compromised due to } v_i\} \\ &= \sum_j \Pr(\text{attribute}_j, v_i) \times C_{ji} \\ &= \sum_j I_{ji} \times C_{ji} \\ &= I_{iA} \times C_{ji} \end{aligned}$$

where the security attribute  $j=1,2,3$  represents confidentiality, integrity and availability.  $I_{iA}$  is the CVSS Base Impact sub-score whereas  $C_{ji}$  is the CVSS Environmental ConfReq, IntegReq or AvailReq metric.

The two detailed expressions for likelihood and impact above in terms of constituent factors, allow defining conditional risk measures. Often risk measures used by different authors differ because they are effectively conditional risks which consider only some of the risk components. The components ignored are then effectively equal to one.

As mentioned above, for a weakness  $i$ , risk is defined as  $L_i \times I_i$ . The conditional risk measures  $\{R_1, R_2, R_3, R_4\}$  can

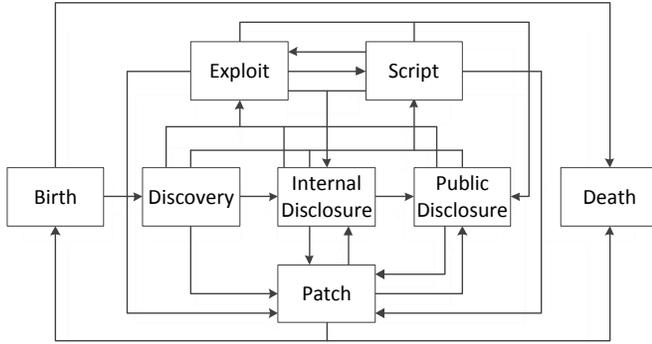


Figure 2. Possible vulnerability lifecycle journey

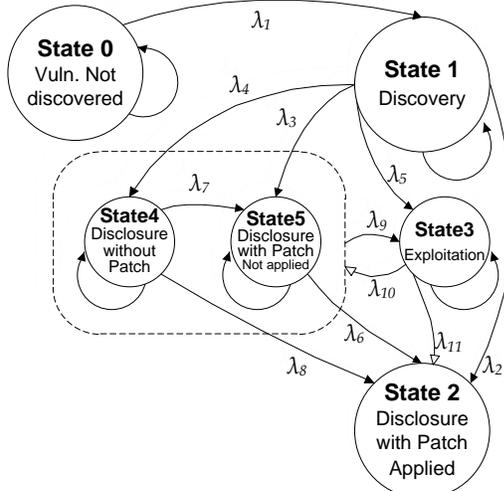


Figure 3. Stochastic model for a single vulnerability

be defined by setting some of the factors in the above equations to unity:

- $R_1$ : by setting  $\{L_{Ci}, L_{Di}, C_{ji}\}$  as unity. The CVSS Base score is a  $R_1$  type risk measure.
- $R_2$ : by setting  $\{L_{Di}, C_{ji}\}$  as unity. The CVSS temporal score is a  $R_2$  type risk measure.
- $R_3$ : by setting  $L_{Di}$  as unity. The CVSS temporal score is a  $R_3$  type risk measure.
- $R_4$ : is the total risk considering all the factors.

In the next two sections, we examine a risk measure that is more general compared with other perspectives in the sense that we consider the discovery of hitherto unknown vulnerabilities. This would permit us to consider 0-day attacks within our risk framework. In the following section a simplified perspective is presented which considers only the known vulnerabilities.

## 5 Software vulnerability Lifecycle

A vulnerability is created as a result of a coding or specification mistake. Fig. 2 shows possible vulnerability lifecycle journeys. After the birth, the first event is discovery. A discovery may be followed by any of these: internal disclosure, patch, exploit or public disclosure. The discovery rate can be described by vulnerability discovery models (VDM) [20]. It

has been shown that VDMs are also applicable when the vulnerabilities are partitioned according to severity levels [21]. It is expected that some of the CVSS base and temporal metrics impact the probability of a vulnerability exploitation [10], although no empirical studies have yet been conducted.

When a white hat researcher discovers a vulnerability, the next transition is likely to be the internal disclosure leading to patch development. After being notified of a discovery by a white hat researcher, software vendors are given a few days, typically 30 or 45 days, for developing patches [22]. On the other hand, if the disclosure event occurred within a black hat community, the next possible transition may be an exploitation or a script to automate exploitation. Informally, the term *zero day vulnerability* generally refers to an unpublished vulnerability that is exploited in the wild [23]. Studies show that the time gap between the public disclosure and the exploit is getting smaller [24]. Norwegian Honeynet Project [25] found that from the public disclosure to the exploit event takes a median of 5 days (the distribution is highly asymmetric).

When a script is available, it enhances the probability of exploitations. It could be disclosed to a small group of people or to the public. Alternatively, the vulnerability could be patched. Usually, public disclosure is the next transition right after the patch availability. When the patch is flawless, applying it causes the death of the vulnerability although sometimes a patch can inject a new fault [26].

Frei has [11] found that 78% of the examined exploitations occur within a day, and 94% by 30 days from the public disclosure day. In addition, he has analyzed the distribution of discovery, exploit, and patch time with respect to the public disclosure date, using a very large dataset.

## 6 Evaluating lifecycle risk

We first consider evaluation of the risk due to a single vulnerability using stochastic modeling [9]. Fig. 3 presents a simplified model of the lifecycle of a single vulnerability, described by six distinct states. Initially, the vulnerability starts in State 0 where it has not been found yet. When the discovery leading to State 1 is made by white hats, there is no immediate risk, whereas if it is found by a black hat, there is a chance it could be soon exploited. State 2 represents the situation when the vulnerability is disclosed along with the patch release and the patch is applied right away. Hence, State 2 is a safe state and is an absorbing state. In State 5, the vulnerability is disclosed with a patch but the patch has not been applied, whereas State 4 represents the situation when the vulnerability is disclosed without a patch. Both State 4 and State 5 expose the system to a potential exploitation which leads to State 3. The two white head arrows ( $\lambda_{10}$  and  $\lambda_{11}$ ) are backward transitions representing a recovery which might be considered when multiple exploitations within the period of interest need to be considered. In the discussion below we assume that State 3 is an absorbing state.

In the figure, for a single vulnerability, the cumulative risk in a specific system at time  $t$  can be expressed as proba-

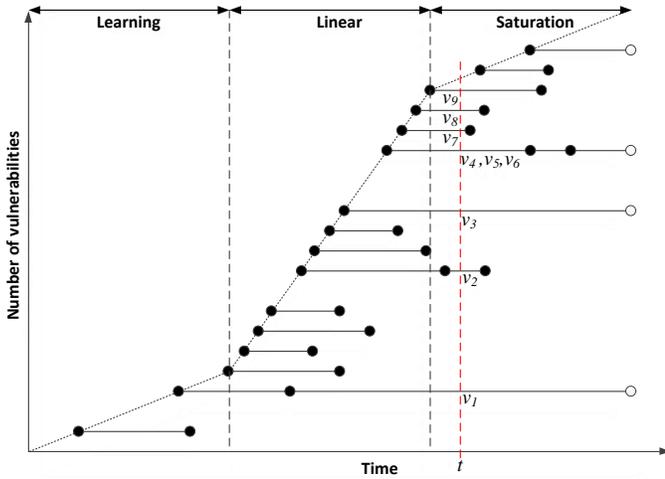


Figure 4. Example of the vulnerability discovery and patch in a system with simplified three phase vulnerability lifecycle

bility of the vulnerability being in State 3 at time  $t$  multiplied by the consequence of the vulnerability exploitation.

$$Risk_i(t) = \Pr\{Vulnerability_i \text{ in State 3 at time } t\} \times exploitation\_impact_i$$

If the system behavior can be approximated using a Markov process, the probability that a system is in a specific state at  $t$  could be obtained by using Markov modeling. Computational methods for semi-Markov [27] and non-Markov [28] processes exist, however, since they are complex, we illustrate the approach using the Markov assumption. Since the process starts at State 0, the vector giving the initial probabilities is  $\alpha = (P_0(0) P_1(0) P_2(0) P_3(0) P_4(0) P_5(0)) = (1 \ 0 \ 0 \ 0 \ 0 \ 0)$ , where  $P_i(t)$  represents the probability that a system is in State  $i$  at time  $t$ . Let  $\mathbb{P}(t)$  be as the state transition matrix for a single vulnerability where  $t$  is a discrete point in time. Let the  $x^{\text{th}}$  element in a row vector of  $v$  as  $v_x$ , then the probability that a system is in State 3 at time  $n$  is  $(\alpha \prod_{i=1}^n \mathbb{P}(t))_3$ . Therefore, according to the Equation (1), the risk for a vulnerability  $i$  for time window  $(0, t)$  is:

$$Risk_i(t) = (\alpha \prod_{k=1}^t \mathbb{P}_i(k))_3 \times impact_i \quad (5)$$

The impact may be estimated from the CVSS scores for Confidentiality Impact ( $I_C$ ), Integrity Impact ( $I_I$ ) and Availability Impact ( $I_A$ ) of the specific vulnerability, along with the weighting factors specific to the system being compromised. It can be expressed as:

$$impact_i = f_c(I_C R_C, I_I R_I, I_A R_A)$$

where  $f_c$  is a suitably chosen function. CVSS defines environmental metrics termed *Confidentiality Requirement*, *Integrity Requirement* and *Availability Requirement* that can be used for  $R_C$ ,  $R_I$  and  $R_A$ . The function  $f_c$  may be chosen to be additive or multiplicative. CVSS also defines a somewhat complex measure termed *AdjustedImpact*, although no justification is explicitly provided. A suitable choice of the impact function needs further research.

We now generalize the above discussion to the general case when there are multiple potential vulnerabilities in a software system. If we assume statistical independence of the vulnerabilities (occurrence of an event for one vulnerability is not influenced by the state of other vulnerabilities), the total risk in a software system can be obtained by the risk due to each single vulnerability given by Equation (5). We can measure risk level as given below for a specific software system.

$$Risk(t) = \sum_i (\alpha \prod_{k=1}^t \mathbb{P}_i(k))_3 \times impact_i$$

The method proposed here could be utilized to measure risks for various units, from single software on a machine to an organization-wide risk due to a specific software. Estimating the organizational risk would involve evaluating the vulnerability risk levels for systems installed in the organizations. The projected organizational risk values can be used for optimization of remediation within the organization.

## 7 Risk from known unpatches vulnerabilities

It can take considerable effort to estimate the transition rates among the states as described in the previous section. A conditional risk measure for a software system could be defined in terms of the intervals between the disclosure and patch availability dates that represent the gaps during which the vulnerabilities are exposed.

We can use CVSS metrics to assess the threat posed by a vulnerability. Let us make a preliminary assumption that the relationships between the Likelihood ( $L$ ) and the Exploitability sub-score ( $ES$ ), as well as the Impact ( $I$ ) and the Impact sub-score ( $IS$ ) for a vulnerability  $i$  are linear:

$$ES_i = a_0 + a_1 \times L_i \quad \text{and} \quad IS_i = b_0 + b_1 \times I_i$$

Because the minimum values of  $ES$  and  $IS$  are zero,  $a_0$  and  $b_0$  are zero. That permits us to define normalized risk values, as can be seen below.

Now, a conditional risk,  $Risk\_c_i$ , for a vulnerability  $i$  can be stated as:

$$Risk\_c_i = L_i \times I_i = \frac{ES_i IS_i}{a_1 b_1}$$

For the aggregated conditional risk is:

$$Risk\_c = \frac{1}{a_1 b_1} \sum_i ES_i IS_i$$

A normalized risk measure  $Risk'_c(t)$  can be defined by multiplying the constant  $a_1 b_1$ , expressed as:

$$Risk'_c(t) = \sum_i ES_i(t) IS_i(t) \quad (6)$$

This serves as an aggregated risk measure for known and exposed vulnerabilities. Its estimation is illustrated below using numerical data.

Fig. 4 is a conceptual diagram to illustrate the risk gap between vulnerability discoveries and patch releases on top of the simplified three phase vulnerability lifecycle in AML model [20]. In the initial learning phase, the software is gaining market share gradually. In the linear phase, the discovery rate reaches the maximum due to the peak popularity of the

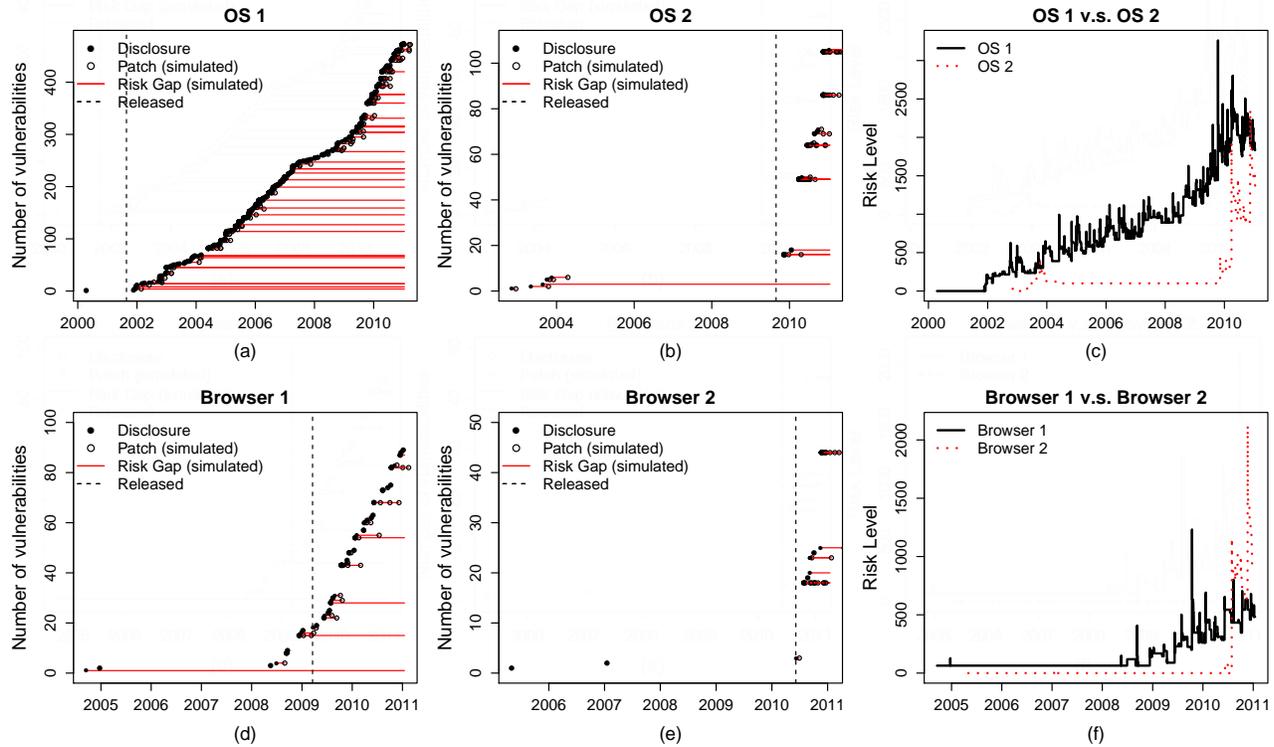


Figure 5. Evaluated risk gaps (a, b, d, e) and normalized risk level (c, f)

Table 1. Average patch time [11]

	0-day	30-day	90-day	180-day
Microsoft	61%	75%	88%	94%
Apple	32%	49%	71%	88%

Table 2. Simulated datasets for patch date

	OS 1	OS 2	Browser 1	Browser 2	
Simulated # of vuln.	0 day	289	33	54	14
	1-30	66	18	12	7
	31-90	61	23	11	9
	91-180	28	18	5	7
	No patch	30	14	7	7
Total [17]	474	106	89	44	

software, and finally, in the saturation phase, vulnerability discovery rate slows down.

In the figure, each horizontal line represents the duration for an individual vulnerability from discovery date to patch availability date. When there are multiple dots at the right, the horizontal line represents multiple vulnerabilities discovered at the same time, but with different patch dates. A white dot is used when a patch is not hitherto available. For example, in Fig 4, at time  $t$  marked with the vertical red dashed line, there are nine known vulnerabilities with no patches. To calculate the conditional risk level at that time point, each single vulnerability risk level need to be calculated first and then added as shown in Equation (6).

We illustrate the approach using simulated data that has been synthesized using real data. Actual vulnerability disclosure dates [17] are used but the patch dates are simulated. XP is currently (Jan. 2011 [29]) the most popular OS with 55.26% share. Also, Snow Leopard is the most popular among non-Windows OSes. IE 8 and Safari 5 are the most adopted Web browsers for the two OSes. Considerable effort

and time would be needed for gathering the actual patch release dates [22], thus simulated patch dates are used here for the four systems. The patch dates are simulated using the aggregate data [11] representing the fraction of vulnerabilities patched, on average, within 0, 30, 90 and 180 days as shown in Table 1. Note that 6% and 12% of the vulnerabilities for Microsoft and Apple respectively are not patched by 180 days. Many of them are patched later, however because of lack of data, the simulated data treats them as unpatched vulnerabilities which would cause the data to differ from real data.

The simulated data sets are listed in Table 2; note that while OS 1, OS 2, Browser 1 and Browser 2 are based on XP, Snow Leopard, IE 8 and Safari 5 respectively, they are used here only to illustrate the procedure and not for evaluation the risk levels of the actual software.

Fig. 5 (a, b, d, e) give the risk gaps for the four datasets. The linear trend observed arises as special cases of the logistic process [30]. Fig. 5 (c, f) give the normalized risk levels calculated daily. As shown in the plots, OS 1 risk level has started to decline while OS 2 risk level is still rising. For the browsers, Browser 2 risk level rises sharply right after the release due to the two sets of vulnerability clusters with no available immediate patches. The long term rising trend observed might be caused by vulnerabilities we have presumed to be unpatched after 180 days. Since the data sets are simulated, the results only serve as an illustration of the approach and do not represent any actual products.

## 8 Conclusions

This paper presents formal measures of security risk that are amenable to evaluation using actual vulnerability data. It

also explores the relationship of CVSS metrics and scores with formal expressions of risk.

While a preliminary examination of some of the software lifecycle transitions has recently been done by some researchers [11][12], risk evaluation considering the vulnerability lifecycle has so far received very little attention. In this paper, a formal quantitative approach for software risk evaluation is presented which uses a stochastic model for the vulnerability lifecycle and the CVSS metrics. The model incorporates vulnerability discovery and potential 0-day attacks. The risk values for individual vulnerabilities can be combined to evaluate risk for an entire software system, which can in turn be used for evaluating the risk for an entire organization. A simplified approach for risks due to known but unpatched vulnerabilities is also given.

While some data has started to become available, further research is needed to develop methods for estimating the applicable transition rates [11][19][31]. In general, the computational approaches need to consider the governing probability distributions for the state sojourn times. Since the impact related scores may reflect a specific non-linear scale, formulation of the impact function also needs further research.

The proposed approach provides a systematic approach for software risk evaluation. It can be used for comparing the risk levels for alternative systems. The approach can be incorporated into a methodology for allocating resources optimally by both software developers and end users.

## 9 References

- [1] C. Alexander, *Market Risk Analysis: Quantitative Methods in Finance*, Wiley, 2008.
- [2] V. Verendel, Quantified security is a weak hypothesis: a critical survey of results and assumptions, Proc. 2009 workshop on New security paradigms workshop, Sept.08-11, 2009, Oxford, UK. pp. 37-49.
- [3] R. L. V. Scoy, Software development risk: Opportunity, not problem (cmu/sei-92-tr-030), Software Engineering Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania, Tech. Rep., 1992.
- [4] S. Farrell, Why Didn't We Spot That?, *IEEE Internet Computing*, 14(1), 2010, pp. 84-87.
- [5] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, 3rd ed. Prentice Hall PTR, 2003.
- [6] National Institute of Standards and Technology (NIST), Risk management guide for information technology systems, 2001. Special Publication 800-30.
- [7] L. A. (Tony) Cox, Jr, Some Limitations of Risk = Threat  $\times$  Vulnerability  $\times$  Consequence for Risk Analysis of Terrorist Attacks, *Risk Analysis*, 28(6), 2008, pp. 1749-1761.
- [8] Open Web Application Security Project (OWASP) Top 10 2010 - The Ten Most Critical Web Application Security Risks, [http://www.owasp.org/index.php/Top\\_10\\_2010-Main](http://www.owasp.org/index.php/Top_10_2010-Main)
- [9] H. Joh and Y. K. Malaiya, A Framework for Software Security Risk Evaluation using the Vulnerability Lifecycle and CVSS Metrics, Proc. International Workshop on Risk and Trust in Extended Enterprises, November 2010, pp. 430-434.
- [10] P. Mell, K. Scarfone, and S. Romanosky, CVSS: A complete Guide to the Common Vulnerability Scoring System Version 2.0, Forum of Incident Response and Security Teams (FIRST), 2007.
- [11] S. Frei, *Security Econometrics: The Dynamics of (IN)Security*, Ph.D. dissertation at ETH Zurich, 2009.
- [12] W. A. Arbaugh, W. L. Fithen, and J. McHugh, Windows of vulnerability: A case study analysis, *Computer*, 33(12), 2000, pp. 52-59.
- [13] P. A. Engert, Z. F. Lansdowne, Risk Matrix 2.20 User's Guide, November 1999, <http://www.mitre.org/work/sepo/toolkits/risk/ToolsTechniques/files/UserGuide220.pdf>
- [14] J. P. Brashear, J. W. Jones, Risk Analysis and Management for Critical Asset Protection (RAMCAP Plus), Wiley Handbook of Science and Technology for Homeland Security, 2008.
- [15] European Network and Information Security Agency (ENISA), Cloud Computing - Benefits, risks and recommendations for information security, Ed. Daniele Catteddu and Giles Hogben, Nov 2009.
- [16] L. Cobb, A Scale for Measuring Very Rare Events, April, 1998, <http://www.aetheling.com/docs/Rarity.htm>
- [17] NIST, National Vulnerability Database (NVD), <http://nvd.nist.gov/>, Accessed on Feb. 2011
- [18] I. Mkpong-Ruffin, D. Umphress, J. Hamilton, and J. Gilbert, Quantitative software security risk assessment model, ACM workshop on Quality of protection, 2007, pp. 31-33.
- [19] S. H. Houmb and V. N. L. Franqueira, Estimating ToE Risk Level Using CVSS, International Conference on Availability, Reliability and Security, 2009, pp.718-725.
- [20] O. H. Alhazmi and Y. K. Malaiya, Application of vulnerability discovery models to major operating systems, Reliability, *IEEE Transactions on*, 57(1), 2008, pp. 14-22.
- [21] S.-W. Woo, H. Joh, O. H. Alhazmi and Y. K. Malaiya, Modeling Vulnerability Discovery Process in Apache and IIS HTTP Servers, *Computers & Security*, Vol 30(1), pp. 50-62, Jan. 2011
- [22] A. Arora, R. Krishnan, R. Telang, and Y. Yang, An Empirical Analysis of Software Vendors' Patch Release Behavior: Impact of Vulnerability Disclosure, *Information Systems Research*, 21(1), 2010, pp. 115-132.
- [23] E. Levy, Approaching Zero, *IEEE Security and Privacy*, 2(4), 2004, pp. 65-66.
- [24] R. Ayoub. An analysis of vulnerability discovery and disclosure: Keeping one step ahead of the enemy, Tech. Report, Frost & Sullivan, 2007.
- [25] Norwegian Honeynet Project, Time to Exploit, <http://www.honeynor.no/research/time2exploit/>, Accessed on Feb. 2011
- [26] S. Beattie, S. Arnold, C. Cowan, P. Wagle, and C. Wright, Timing the application of security patches for optimal uptime, Proceedings of the 16th USENIX conference on System administration, Berkeley, CA, 2002, pp. 233-242.
- [27] V. S. Barbu, and N. Limnios, *Semi-Markov Chains and Hidden Semi-Markov Models Toward Applications: Their Use in Reliability and DNS Analysis*, Springer, New York, 2008.
- [28] Y. K. Malaiya and S. Y. H. Su, Analysis of an Important Class of Non-Markov Systems, *IEEE Transactions on Reliability*, R-31(1), April 1982, pp. 64 - 68.
- [29] NetMarketShare, Operating System Market Share, <http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=10>, Accessed on Feb. 2011
- [30] G. Schryen, Security of open source and closed source software: An empirical comparison of published vulnerabilities. Proceedings of the 15th Americas Conference on Information Systems. 2009.
- [31] M. D. Penta, L. Cerulo, and L. Aversano, The life and death of statically detected vulnerabilities: An empirical study, *Information and Software Technology*, 51(10), 2009, pp. 1469-1484.

# Study of Information Security Pre-Diagnosis Model for New IT Services

Wan s. Yi<sup>1</sup>, Kwangwoo Lee<sup>1</sup>, and Dongho Won<sup>1</sup>

<sup>1</sup>Information & Communication Security Lab.

School of Information and Communication Engineering, Sungkyunkwan University,  
300 Cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do, 440-746, Korea

**Abstract** - Along with fast development of IT, conventional industries are converging with information technologies thus creating bigger IT market. Fast changing environment is an opportunity for many businesses and at the same time, it is a threat which needs to be overcome to survive in the competitive business world. This paper introduces information security pre-diagnosis methodology and procedure. It provides a method to develop information security measures by analyzing threats and vulnerabilities covering planning phase to testing phase prior to initiating its service to their customers. This paper deals with information security pre-diagnosis implementation methodology, trial run result and its effectiveness.

**Keywords:** pre-diagnosis, risk analysis, vulnerability analysis, security measures, threat scenario, on-site review.

## 1. Introduction

In order to initiate a new information communication service, which requires a lot of initial investment, service provider must go through feasibility analysis at the beginning of planning stage to assure business success. For new IT services, customers, who are willing to pay the price, and diversity of customers will determine the new service market. Therefore service developer must consider customers' range and service price for feasibility study. Therefore to minimize security risks, there is increased need for systematic security vulnerability analysis starting from planning phase till initiation of new service.

Implementing security measures through security vulnerability analysis in early stages will dramatically reduce costs than implementing security measures

during operation phases. Especially, finding vulnerabilities and security measures, including technical, managerial and physical measures, in development phase is so much more cost effective than finding them in operational phase.

## 2. Risk Analysis Methodologies

Risk analysis methodology is a very popular subject to study. And in many countries, Governments have taken the initiatives to research and develop risk analysis methodologies. Governments developed information security management guidelines and distributed to public and private sectors to actively promote information security consulting businesses. Currently, more than 100 risk analysis automation tools have been developed and are used at various areas at the moment. Leading risk analysis methodologies include NIST, GMITS, GAO and OCTAVE of the US, BS7799 of the UK, and CSE of Canada. In Korea, KISA (Korea Internet and Security Agency) and TTA (Telecommunications Technology Association) have announced the risk analysis methodology. [Table 1] below compares assets, weaknesses, threats and level of risk analysis methodologies from different countries.

Comparing other methodologies indicated that major steps were similar while there were some differences in classification methodology and how to calculate risk level based on the vulnerability and threat analysis. Since such existing risk analysis methodology is based on what kind and how much information assets are being used during the operation phase, its major focus is on the security incidences response rather than minimizing threats and vulnerabilities especially in the development phase. Therefore, existing risk analysis methodologies are inadequate to apply against future oriented u-IT services.

\* Corresponding author: Dongho Won (dhwon@security.re.kr)

Table 1. Comparison of Risk Analysis Models of Different Countries

Methodology	NIST	GMITS	BS7799	CSE	OCTAVE	KISA
Classification of assets	<ul style="list-style-type: none"> <li>hardware</li> <li>software</li> <li>system</li> <li>interphase</li> <li>information &amp; data</li> <li>human</li> <li>system</li> </ul>	<ul style="list-style-type: none"> <li>information &amp; data</li> <li>hardware</li> <li>software</li> <li>telecommunication</li> <li>equipment</li> <li>documents</li> <li>palmsware</li> <li>documents</li> <li>capital</li> <li>manufactured products</li> <li>service</li> <li>confidence and trust in service</li> <li>environmental equipment</li> <li>manpower</li> <li>organization image</li> </ul>	<ul style="list-style-type: none"> <li>information</li> <li>software</li> <li>physical equipment</li> <li>service</li> <li>documents</li> <li>human</li> <li>company image, reputation</li> </ul>	<ul style="list-style-type: none"> <li>information</li> <li>process</li> <li>platform</li> <li>interface</li> <li>human</li> <li>environment</li> <li>material asset</li> <li>immaterial asset</li> </ul>	<ul style="list-style-type: none"> <li>information</li> <li>system</li> <li>software</li> <li>hardware</li> <li>human</li> </ul>	<ul style="list-style-type: none"> <li>information &amp; data</li> <li>documents</li> <li>hardware</li> <li>software</li> </ul>
Classification of weaknesses	-	<ul style="list-style-type: none"> <li>environment and basic facilities</li> <li>hardware</li> <li>software</li> <li>telecommunications</li> <li>documents</li> <li>human</li> <li>general weaknesses</li> </ul>	<ul style="list-style-type: none"> <li>employee security</li> <li>physical environment</li> <li>security</li> <li>management of computer &amp; networks</li> <li>Maintain system access control &amp; development</li> </ul>	<ul style="list-style-type: none"> <li>external</li> <li>technical</li> <li>process</li> <li>system</li> <li>technical</li> <li>process</li> <li>Object</li> <li>technical</li> <li>process</li> <li>manpower</li> <li>accessibility</li> <li>knowledge</li> <li>training</li> <li>process</li> </ul>	<ul style="list-style-type: none"> <li>server</li> <li>network</li> <li>security</li> <li>system</li> <li>desktop</li> <li>PC</li> <li>notebook</li> <li>storage device</li> <li>wireless LAN, mobile phone</li> <li>etc</li> </ul>	<ul style="list-style-type: none"> <li>management</li> <li>policy, organization, human resources</li> <li>building, facilities, etc.</li> <li>technical</li> <li>network system, application, database, pc</li> </ul>
Classification of threats	<ul style="list-style-type: none"> <li>threat from nature</li> <li>threat from humans</li> <li>consideration of intention of threat</li> <li>threat from environment</li> </ul>	<ul style="list-style-type: none"> <li>planned</li> <li>environmental</li> <li>human</li> </ul>	<ul style="list-style-type: none"> <li>infected/had not allowed to access the system or network</li> <li>software operation malfunction</li> <li>Sending of not allowed message</li> <li>re-sending of message by 3<sup>rd</sup> party</li> <li>fire</li> <li>burglar</li> <li>employee mistake</li> </ul>	<ul style="list-style-type: none"> <li>non-human</li> <li>random (nature)</li> <li>planned (human)</li> <li>artificial</li> <li>internal</li> <li>external</li> </ul>	<ul style="list-style-type: none"> <li>human</li> <li>system</li> <li>hardware</li> <li>software</li> <li>etc</li> <li>natural disaster</li> <li>communication obstacle</li> <li>physical environmental obstacle</li> </ul>	<ul style="list-style-type: none"> <li>executor</li> <li>human</li> <li>non-human</li> <li>access route</li> <li>network</li> <li>optical</li> <li>intention</li> <li>coincidence</li> <li>intentional result of damage</li> <li>change</li> <li>vulnerability</li> <li>destruction</li> <li>interruption</li> </ul>
Calculation method of degree of risk	<ul style="list-style-type: none"> <li>standard matrix for calculating degree of risk</li> <li>Asset</li> <li>frequency of threat</li> <li>severity of threat</li> <li>level of threat</li> </ul>	<ul style="list-style-type: none"> <li>standard matrix for calculating degree of risk</li> <li>Asset</li> <li>weakness</li> <li>threat</li> <li>degree of risk</li> </ul>	<ul style="list-style-type: none"> <li>standard matrix for calculating degree of risk</li> <li>Asset</li> <li>weakness</li> <li>threat</li> <li>degree of risk</li> </ul>	<ul style="list-style-type: none"> <li>scenario of threat</li> <li>Asset -&gt; threat (motive, ability to execute) -&gt; weaknesses (severity, vulnerability) -&gt; degree of risk</li> </ul>	<ul style="list-style-type: none"> <li>risk evaluation standard established by situation</li> <li>Important assets -&gt; threat profile -&gt; weakness -&gt; threat (degree of damage, frequency of threat)</li> </ul>	<ul style="list-style-type: none"> <li>standard matrix for calculating degree of risk</li> <li>Asset</li> <li>weakness</li> <li>threat</li> <li>degree of risk</li> </ul>

Costs needed to find the vulnerabilities and fix them before the initiation of a service is much more cost effective than finding and fixing them in the operational phase. According to IBM system laboratory, implementation of security measures in the implementation phase is about 60 to 100 times cost effective than implementing them in the operational phase. Also according to Soo Hoo, return of investment of analyzing vulnerabilities in the development phase increases by 21%. [1]

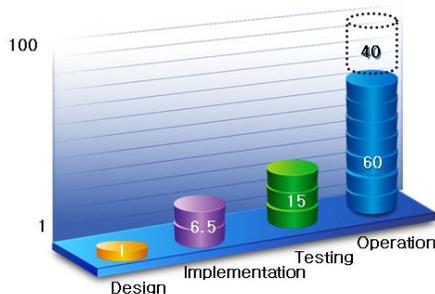


Figure 1. Cost for implementing security measures in each development phase

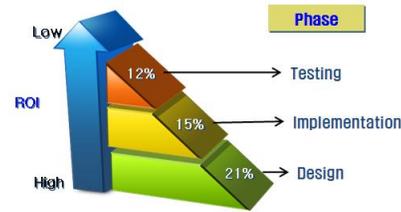


Figure 2. ROI for implementing security measures in each development phase

### 3. IT Security Pre-Diagnosis Model

#### 3.1 Objective

By using IT security pre-diagnosis methodology, risk from information threats and vulnerabilities can be identified and mitigated in advance, the trial-and-error at development and operation phase can be prevented and effective countermeasures can be prepared.

#### 3.2 Planning for IT Security Pre-diagnosis

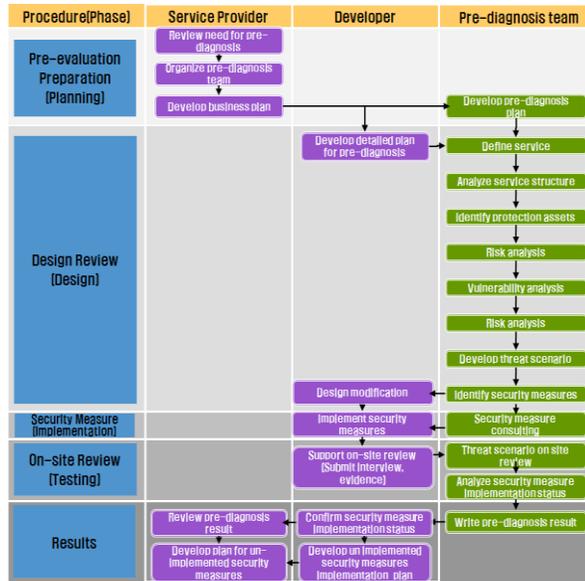
IT security pre-diagnosis is a security analysis methodology to find information security threats and vulnerabilities that are embedded while a new system is being developed. A plan for pre-diagnosis is generally prepared during the business planning stage which is carried out in the beginning of planning phase. However, IT security pre-diagnosis plan can be generated at every phase of service life cycle. If there is a problem or there is a high possibility of serious intrusion threat, a plan can be generated even at service operation phase.

#### 3.3 IT Security Pre-diagnosis Team

The purpose of IT security pre-diagnosis is to analyze and evaluate the threats and vulnerabilities of target business and then identify security measures to mitigate them. A separate team is organized composed of business manager, security manager and service provider. If needed, professionals from outside of the organization can join the team for assistance.

#### 3.4 Structure of Pre-diagnosis Methodology

Detailed procedure for IT security pre-diagnosis is shown below.



### 3.4.1 Design Information Security Requirements

#### 3.4.1.1 Develop Information Security Requirements

The information security requirements, which the business planner must refer to during the business planning step, are developed to be used as a checklist to develop information security measures.

#### 3.4.1.2 Review Information Security Requirements

In this stage information security requirements defined in '3.4.1.1 Develop Information Security Requirements' is reviewed to see whether requirements are reflected in business plan and business execution plan.

### 3.4.2 Define IT Service

#### 3.4.2.1 Define Service Concept

The purpose, business target of audience and major business functions to be pre-diagnosed are defined in detail. As a result, business definition and concept is better understood.

### 3.4.3 Analyze Service Structure

#### 3.4.3.1 Develop Service Structural Diagram

It defines service structure and relationship with various application developers so that as many application as possible can be provided to customers.

#### 3.4.3.2 Identify Components

It identifies service components for the target business and detailed information for each service components is gathered.

#### 3.4.3.3 Analyze Communication Protocol

It identifies used communication protocols to be used to transfer data and to control applications. And also identifies when and at which service process, each network protocol is used.

### 3.4.4 Application Service Analysis

#### 3.4.4.1 Prepare Application List

An 'application security list' is prepared to be used to identify which application and data to protect. It is also used as one of a input data for risk analysis.

#### 3.4.4.2 Design Data Flow Diagram

List of data and what kind of control is needed for each application is identified. Also, data flow is analyzed for entire data life cycle including creation, ownership, use and deletion. And it is mapped with which application is involved at which stage.

#### 3.4.4.3 Analyze Work Flow

At first, service customers, operator, administrator, security officer, chief security officer and chief personal information officer is identified and their roles and responsibilities are defined. Also define which system is involved at which stage of the service and what kind of data is transmitted, used, created, and destroyed. At this stage, working environment should also be looked into.

### 3.4.5 Analyze Security Management Status

#### 3.4.5.1 Check Security Management Status

The security management and operation status of the service is studied to analyze the information security threats and vulnerabilities.

### 3.4.6. Identify Protection Target List

#### 3.4.6.1 Develop Criteria to Measure Importance

After making a list of assets to be protected, evaluation criteria is developed to make a priority list

depending on which one plays vital role and is important for the service continuity.

#### 3.4.6.2 Measure Importance of each element

Criteria, developed in '3.4.6.1 Develop Criteria to Measure Importance', is used to measure the importance of each assets need to be protected make a priority list.

### 3.4.7 Threat Analysis

#### 3.4.7.1 Develop Threat Analysis Criteria

Criteria for evaluating the level of threat are developed with consideration on the likely hood to occur and what kind of effect if incident occurs.

#### 3.4.7.2 Evaluate Threats

The level of threat of each component is evaluated in terms of the impact to the organization.

### 3.4.8 Vulnerability Analysis

Vulnerability analysis identifies vulnerabilities at each business process, systems and data. And it evaluates the impact on managerial, physical and technical aspect in order to classify the levels of vulnerabilities.

### 3.4.9 Risk Analysis

#### 3.4.9.1 Develop Risk Measurement Criteria

At this stage, risk measurement criteria are developed to measure weights on each risk.

#### 3.4.9.2 Risk Analysis

Risk analysis is to find out the potential risk hidden in the new business model. The result of the risk analysis will be used as one of input data to select security measures.

### 3.4.10 Generate Threat Scenario

#### 3.4.10.1 Develop Threat Scenario

The threat scenario is generated on the bases of the service structure, threat and vulnerability analysis results. The results will be used to develop security measures in the security requirement later.

#### 3.4.10.2 Measure Importance of Scenario

Looking at the developed scenarios, prioritize them so that the ones with the highest possibilities to occur and one with the highest impact will receive most attention.

### 3.4.11 Develop Security Measure

#### 3.4.11.1 Security Measures for each Protection Target

Security measures, from the result of risk analysis, are planned in managerial, technical and physical perspectives for each protection target.

### 3.4.12 Analyze Implemented Security Measures

#### 3.4.12.1 List Implemented Security Measures

Inspection of whether the security measures identified in '3.4.11 Develop Security Measures' are actually implemented and executed.

#### 3.4.12.2 Analyze Security Measures Effectiveness

Appropriateness of the security measures implemented and executed to mitigate the risks of the target service is analyzed whether they are sufficient or is left over risk are manageable. And detailed execution status of the security measures is analyzed to identify the improvement opportunity.

### 3.4.13. Inspect Information Security Threat

#### 3.4.13.1 On-Site Review

On-site review using prepared threat scenarios during '3.4.10.1 Develop Threat Scenarios' is performed to check the feasibility of the scenario, and check if the security measures identified is sufficient to protect against identified threats and vulnerabilities.

### 3.4.14 Identify Information Security Requirement

#### 3.4.14.1 Identify Security Requirement for each Protection Target

Out of all the security measures, there are the ones that are not yet implemented due to lack of cost or manpower. Executives, managers and system operators are to define organization's security requirements. Security requirements should be able to mitigate vulnerabilities identified in vulnerability analysis step or on-site review.

### 3.4.15. Develop Security Measures Implementation Plan

3.4.15.1 Write Security Measures Implementation Plan

Security measures implementation plan should include information on how to mitigate vulnerabilities that are not fully covered by current implemented measures.

3.4.15.2 Write Security Measures Implementation Schedule

Implementation schedule for each security measures, that are not sufficient or not implemented yet, should be included as a result of '3.4.15.1 Write Security Measures Implementation Plan'.

4. Effectiveness of Security pre-diagnosis

First trial was done on a following service where RFID tag is used to determine whether whisky is genuine or not.

<Trial information security pre-diagnosis operation environment>  
 ·Target : RFID tag installation target – 21 and 17 years old Imperial whisky  
 ·# of target: 15,000 RFID tags, 135 RFID readers  
 ·price per equipment: 30 cents for a tag, \$230 for RFID reader that can determine genuine liquor

<Equation to Calculate Effectiveness >

$$TC = \sum_{i=1}^l \alpha_i + (\sum_{i=1}^m \beta_i + K) + \sum_{i=1}^n \gamma_i$$

$\alpha_i$  = I the equipment replacement cost  
 l = # of replacement equipment  
 $\beta_i$  = I th equipment replacement and collection labor cost  
 m = # of replacement equipment  
 K = Compensation cost for delay in service connection  
 $\gamma$  = Compensation cost for disclosure of secret information  
 n = Service enroll and # of users

o Trial implementation result (without information security pre-diagnosis)

: During the test phase, found a critical vulnerability in RFID readers' firmware and those readers needed to be replaced

- Security target: mobile RFID reader (\$230 per reader)

- Security measure implementation phase: Test phase

- Security measures: RFID readers without data encryption mechanism needed to be replaced

Table 2. Calculating cost

Category	Equation	Calculation	Cost
Initial damage cost	# of replaced device × Cost per device	135 readers × \$ 250 price for each	\$ 33,750
2nd damage cost	Cost for 1 man/day1) × # of replaced device	\$ 136 × 135 readers	\$ 18,339
<b>Total cost</b>			<b>\$ 52,149</b>

If liquor market is expected to grow up to 1 billion dollars then effectiveness of information security pre-diagnosis is expected to be 100 times.

5. Conclusion

Since the existing risk analysis methodology for security management is to reduce the threat against information asset, it is inadequate to apply them to the new future oriented u-IT services that are being developed. Furthermore, its procedure is too complex and the framework of classification criteria is not appropriate in many cases, making it difficult to be used by the field operation personnel for new IT service. Therefore, this paper proposes the information security pre-diagnosis methodology for u-IT service which analyzes service structure and operation procedure. Then it makes a plan to cope with the threats so that the security will be considered in advance before the service is initiated and provided to customers. By executing the information security pre-diagnosis in little segments such as design review, define and implement security measures, site review and security measures review, the complexity is minimized by clarifying the outcome at each and every step.

During the information security pre-diagnosis, we were able to find vulnerabilities such as data encryption functions not being implemented which could lead to exploitation. By finding these vulnerabilities and implementing security measures before service is initiated, we were able to calculate that it is cost effective since there would not be any more cost involved with implementing additional security measures.

## 6. Acknowledgement

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the "ITRC" support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2011-C1090-1001-0004)  
- Corresponding Author: Dongho Won

## 7. References

- [1] Tangible ROI through Secure Software Engineering  
by Kevin Soo Hoo, Andrew W. Sudbury and Andrew R. Jaquith
- [2] NIST, "Risk Management Guide for Information Technology Systems" 2001
- [3] BSI, BS7799-Code of Practice for Information Security Management, British Standards Institute, 1999
- [4] Kenneth R. van Wyk, and Gary McGraw. 2005. Bridging the Gap between Software Development and Information Security. IEEE Security and Privacy, pp. 75-79.
- [5] Elaine Fedchak, Thomas McGibbon and Robert Vienneau, 2007. Software Project Management for Software Assurance. DACS Report Number 347617.
- [6] Noopur Davis, 2005. Secure Software Development Life Cycle Processes: A Technology Scouting Report. CMU/SEI, USA
- [7] Ron Moritz and Scott Char etc, 2004. Improving Security Across the Software Development LifeCycle. NCSS, USA  
CMU S/W Engineering Institute, The Team S/W Process and Security. <http://www.sei.cmu.edu/tsp/tsp-security.html>.
- [8] Department of Homeland Security, 2006. Security in the software lifecycle-Draft version 1.2. DHS Report, USA.
- [9] GAO, 2004, Knowledge of Software Suppliers Needed to Manage Risks, Report to Congressional Requesters, USA  
NIST, 2008, Security Considerations in the System Development Lifecycle, Special Publication 800-64 Revision 2, USA
- [10] Kwangwoo Lee, Yunho Lee, Dongho Won and Seungjoo Kim, "Protection Profile for Secure E-Voting Systems", Proc. of ISPEC 2010, Information Security Practice and Experience Conference 2010, Springer-Verlag, LNCS 6047, Seoul, Korea, March 12-13, 2010, pp.386-397.
- [11] Heasuk Jo, Seungjoo Kim, and Dongho Won, "A Study on Comparative Analysis of the Information Security Management Systems", Proc. of ICCSA 2010, The International Conference on Computational Science and Its Applications 2010, Springer-Verlag, LNCS 6019, Fukuoka, Japan, March 23-26, 2010, pp.510-519.

[12] Hyunsang Park, Kwangwoo Lee, Yunho Lee, Seungjoo Kim and Dongho Won, "Security Analysis on the Online Office and Proposal of the Evaluation Criteria", International Conference on Computer Networks and Security(ICCNS 2009), Bali, Indonesia, November 25-27, 2009, pp.198-204.

# Common Network Security Threats and Counter Measures

A. Mahmoud Haidar<sup>1</sup>, B. Nizar Al-Holou, Ph.D.<sup>2</sup>

<sup>1</sup>Dialexa LLC, Dallas, Texas, U.S.A

<sup>2</sup>Electrical and Computer Engineering, University of Detroit Mercy, Detroit, Michigan, U.S.A

**Abstract** - *Despite the growing level of interest in the field of Network Security, there is still little knowledge about the actual issues involved in securing networks. The real dangers of cyber crime are of serious consequence. Individuals with sufficient technical knowledge of Information Technology (IT), networks, and networking devices can steal sensitive information and may exploit vulnerable network systems. In this paper we illustrate some of those threats and learn how to protect our network from attacks and exploits. For this purpose we also propose a set of developed theoretical and practical laboratory sessions that can serve as a complement to academic/professional introductory network security classes.*

**Keywords:** Network, Security, Labs, Software

## 1 Introduction

A computer crime is rarely detected by a victim, which makes it very hard to precisely determine the rate of its occurrence. The General Accounting Office reported approximately 160,000 successful attacks out of 250,000 attempts annually [1]. Moreover, the Defense Information Systems Agency found that 65% of attempted attacks were successful [2].

The cost of computer fraud and abuse in the US is over \$3 billion each year, which is significant considering that over 90% of computer fraud cases are not reported. A study was conducted by BYTE magazine showed that 53% of its readers have experienced data losses that cost \$14,000 on average. A survey of over 600 governmental agencies and companies in the United States and Canada revealed that around 63% reported that their computers were infected by at least one virus. In fact, there exist over 2,500 various viruses are spreading around global at any given instance of time. Moreover, the style and behavior of those viruses are evolving rapidly. That's why hackers with extensive experience are more able to automate the exploitation of networks and systems than any other day.

With the increasing security threats, protecting our networks and systems from unauthorized access and exploits has become an urgent necessity. Hence, arming Information Technology students and professionals with knowledge and

experience necessary to identify and resolve these threats becomes imperative. For this reason, we propose in this paper an educational framework that employs freeware off the shelf tools to learn network security. In section 2, the methodologies to penetrate and exploit a network are presented, while the labs developed to learn those methods and how to address them are discussed in section 3.

## 2 Network Penetration and Exploitation Methodologies

Attacking a network is generally a two steps procedure. The first step is to penetrate the network and find a weakness you can take advantage of to gain access to the network. The second step would be attacking and exploiting computers in that network. For each step, there is a common set of tools and methodologies.

### 2.1 Network Penetration

#### 2.1.1 Scanners

Scanners can be classified into different categories based on the software application they run, which are designed to either probe server-side or client-side. Examples include TCP/IP/UDP port Scanner, Shared Scanner on a network, and NetBIOS Scanner. These scanners are capable to scan standalone workstation, a group of computers that are connected to network, domains or sub-domains, providing detailed information regarding the scanned area such as open ports, active services, shared resources and the active directory information [5].

IP scanner is used mainly to identify if an IP address is active or not, the IP scanner should be able to scan range of IP addresses, C or B range or even sub range to give first report for the active IP(s). The IP scanner is the first step in gathering the information a target network or system.

A port scanner, on the other hand, scans the host's Ports. It looks for open service ports on the target. Each port is associated with a service that may be exploitable or contain vulnerabilities. Port scanners can be used to scan specific ports or they can be used to scan every port on each host and is used as a next step after knowing if the system alive or not using the IP scanner.

### 2.1.2 Sniffers

Sniffers are programs that passively monitor and capture traffic. Almost any laptop or PC can be turned into a sniffer by installing sniffer software, much of which is freely available on the Internet. The system running the sniffer should have a network interface card that can be used in promiscuous mode. Promiscuous mode enables the sniffer to view but not respond to network traffic, thereby making the sniffer essentially invisible to the network.

Sniffers are very useful tools during penetration testing and network troubleshooting; we commonly use them to capture user names and passwords from FTP and telnet sessions. In addition, sniffers can be used to capture any network traffic that is not encrypted, such as e-mail, HTTP, and other clear text services. Sniffers are generally able to intercept network traffic only on their local network segment. For instance, if a sniffer is located on a shared network that uses hubs, it can view all traffic on the entire network. If a sniffer is located on a switched network (one that uses switches versus hubs), the sniffer can see only broadcast traffic and traffic directed to it. To sniff a switched network, the sniffer would have to be located on a switch port that mirrored the traffic to other ports. Emerging sniffers, such as Dsniff by Dug Song, can sniff switched networks. The thought that switched networks are safe from sniffers are no longer true. Hence, encrypting sensitive information is always recommended to eliminate the affect of sniffing. Data encryption will be discussed in more detail in later sections of this paper.

### 2.1.3 Denial of Service (DoS)

Denial-of-Service (DoS) attacks are well known attacks and have been the bane of the security professionals. The objective of a DoS attack is to exhaust the resources (memory, buffer space, or CPU) to make its service unavailable to legitimate users.

Denial-of-Service (DoS) attacks can be accomplished by two methods. The first and most commonly used method is to flood the target to exhaust its resources. The second method is to create multiple attacks to confuse and crash the target.

Denial-of-Service (DoS) attacks are considered one of the most marketed hacker attacks since its tools have been the destruction of many powerful security structures. The main objective of this kind of attacks is to prevent access to a service or a resource located on the server, and makes it unavailable by the end-user. This is often performed by using flooding techniques against the target server in order to exhaust its specific resources (CPU, memory or buffer) depending on the main service that it provides. Also several attacks aim to stop these services by sending confusing packets to the target, resulting in system crashes while processing these packets due to some bug in the target.

Avoiding penetration is accomplished using Intrusion Detection Systems (IDS) and Firewalls. The main function of IDS is similar to that of burglar alarms. While the firewall keeps its organization safe from any external spiteful attacks through the Internet [3], the IDS detects any illegal attempt to break in the firewall security or any plans to access the trusted side, and once one of the previous actions occurred, an alert is sent to the system administrator warning him/her of a breach security existence [4].

## 2.2 Network Attacks and Exploits

### 2.2.1 Viruses and Worms

Day after day, the reliance on computer applications and programs increases. But what most of us don't know, that programs by themselves often expose a security threat. The work done by a program is hidden from the user. We only know what input we gave to the program and the output displayed by the program. Hence, most of the programs are treated as a black box. A malicious code could be hidden in that black box causing an intentional harm to the computer or the network. The most common form of such codes are Viruses and Worms. The origin of the word "virus" is Latin; which means a poison. In biology, it is defined as an infectious agent that is unable to grow or reproduce outside a host cell. A computer virus, on the other hand, is a set of code instructions encapsulated within an executable file, made to cause damage on the host machine, in such a way it is executed when the host executable is executed. By June 2005, there had been 103,000 different computer viruses created. Viruses are distinguished either by their function or category/type. The function is the harm that the virus creates. Whereas the category/type defines the characteristics of the virus. The following defines some of the important computer viruses' categories [5]:

**Polymorphic Viruses-** Are viruses that can change themselves after every infection to avoid identification by virus scanners. They are considered by many the most dangerous type of viruses [6].

**Stealth Viruses-** Are viruses that hides the harm they have caused. This is mainly done by taking control over the system module responsible of reporting or detecting the harm caused by the virus. When the stealth virus takes over this module, it will report the correct information before infection and hides the damage done.

**Fast and Slow Infectors-** A slow infector virus is the traditional virus which infects the programs when they are created. As for the fast infector virus, it infects the programs when they run in the memory. The main reason behind creating a fast infecting virus, is to infect the anti virus when it runs in the memory so it would infect all the files being scanned.

**Sparse Infectors-** Is the virus that uses a certain technique to decrease the probability of its detection. For example, it might only infect files after being executed 12 times, infect files with a certain size range, etc..

**Armored Viruses-** This virus is made in a way that it is very hard for anti-virus engineers to reverse engineer it. Usually the reverse engineering of a virus is done by disassembling its code. For this purpose, virus writers write thousands and thousands of unnecessary assembly code to make the virus code look like a maze and confusing.

**Virus Droppers-** It is a regular program that doesn't cause any harm to your computer other than dropping or installing a virus.

Regardless of the virus category or type, every virus generally has four main parts as shown in *Figure 1*.

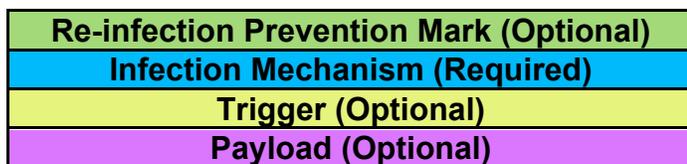


Figure 1: General Virus Stack

**Re-infection Prevention Mark-** It is a mark that the virus leaves on the infected file or system so it won't infect again. This part of the virus is optional.

**Infection Mechanism-** It is the method used by a virus to spread or infect other files on the computer. This part is required for any virus.

**Trigger-** Is an optional property. It defines the condition required to execute the virus's payload.

**Payload-** Is the damage that the virus causes to the infected computer besides spreading into other files.

A computer worm on the other hand, is a malicious program that copies itself over a network without the user consent, intervention, or knowledge [7]. It is very similar to computer viruses in design. However, it doesn't need a host file to replicate itself over the network. It takes advantage of the holes or weak spots in your system to travel across the network. It could send itself through your email, local network, or internet. When it is copied to another computer, it will copy itself to all the computers in that network and so forth. So, the worm can spread among computers exponentially. One the dangers is if the worm creates many copies of itself on the same computer, then each copy will send a copy of itself through the network. This will exploit the network and affect the bandwidth considerably.

A common method to avoid viruses and worms is to use an anti virus program. It is a software used to search for,

identify, and recover a computer from hosted viruses. An anti virus software uses three main approaches to detect a virus:

**Virus Dictionary-** When a virus is first reported by a user to an anti virus software company, an anti-virus researcher will examine the virus, reverse engineer it, and create an anti virus to recover from it. The virus identified will be added to a virus dictionary, which contains records of all the viruses previously identified. Each record will contain the virus identifier, function, type, and actions need to be taken to recover from it. The dictionary can be later used by the anti virus software to detect viruses and recover from them. This is done by going through all the files in the computer and comparing each file by the records in the dictionary. If it matches a record, then the action defined will be executed.

**Programs' Behavior Monitoring-** Using this approach, the anti virus software will monitor the programs being executed in the computer at real time. If it detects a suspicious behavior, an alert will be popped to the user. A suspicious behavior could be a program trying to write data to another executable. This approach is used to detect newly created viruses.

**Program Execution Emulation-** An anti virus software may emulate a program execution to check if it is a virus or infected file. This is done by executing the beginning of the program to check if it will try to infect other files or execute a damaging payload.

### 2.2.2 Password Cracking

A password is a piece of information needed to access a private resource such as: emails, programs, bank accounts, computers, routers, house security systems, etc. Because of their importance and purpose, passwords should not be guessed, recovered, or bypassed by those who are not allowed to access the password-protected resource.

Since it must be known to the program or application it is protecting, a password is usually stored in a database or file to be later used for comparison with the password provided by the user to grant/revoke access. Securing this file is imperative. Moreover, precautionary measures should be taken to make it less probable for a person to crack passwords stored in that file, if the network or system was penetrated. One of the most commonly used approaches in this regard, is to prevent storing the passwords in clear text format. This can be done by applying an encryption function on the password. However, this encryption function must be a "one-way function"; which means if you have a password, you can get its encryption, but if you have the password encryption, you can't get the password. This type of functions is called a cryptographic hash function where its encrypted output is called a hashed password.

Most of the operating systems being used nowadays use hashed passwords. Linux-based systems, for example,

currently use MD5 hashed passwords. As for Windows systems, prior to Vista, uses LM hashed passwords [8]. Hashed passwords add more security to our system. However, attacks shown in the following list, can be used to recover a hashed password. The main idea behind those attacks is that they keep generating hashed passwords from a clear text password and they compare the result with the hashed passwords in the password file. If, they match, then the clear text password is the password. You should note that unless you are trying to guess the password, the attacks presented will not work without knowing what algorithm was used to produce the hashed password.

*Dictionary-* This attack uses a dictionary file which contains most of the words that we use. Of course, the more words are there in the dictionary the higher the probability to crack a password. This attack is only effective if the password was a single or a combination of alphabetic words. Using this attack, you will have a fair chance to crack a password. A study that was conducted lately shows that 3.8% of the passwords are a single word passwords and 12% are a single word plus one numeric digit in which 66% of the time is "1" [9].

*Brute Force-* In this attack, the cracking program generates every possible combination of a password. Theoretically, this attack will always work and eventually crack any password. However, the larger the password the less practical the attack will be. For example, if we had a three digit alphanumeric password, we would need to generate 46,656 passwords ( $36 * 36 * 36$ ) because in each digit we have 36 possibilities (26 letters and 10 numbers). Imagine now if the password used one of the other characters on the keyboard like `~!@#%&^&*()-_+=~:~;~?~/.>,<{}|[]``. That's another 32 possibilities for each digit, this will exponentially increase the number of needed passwords to generate. In real life, the average length of a password is between 8 -12. So, we need  $9.77477912 \times 1021$  (6812) trials to be certain that we will crack a 12 character length password. This is true only if we know what is the length of the password. In case we don't know, we will need to generate all possible passwords with lengths between the minimum and maximum allowed password length. If we had a system that allows password lengths between 1 and 12, we will need  $6812 + 6811 + 6810 + 689 + \dots + 681$  trials. It is obvious to see how unpractical it is to use the brute force attack when the password length is long or not known. A more practical approach is a hybrid attack between dictionary and brute force where we use a dictionary word and start generating a prefix and suffix. Of course, this will work only if the password had a dictionary password in part of it.

*Pre-computation-* It is similar to brute force attack except that it is done before attempting to crack a password. In this approach, we generate all possible passwords to create a hash lookup table containing the clear text passwords paired with their hash. The table will later be used for hash lookup. This will only take the searching time to crack the password.

Although, it took the same time as brute force attack while generating the hash lookup table, the pre-computation attack is more effective if we want to crack many passwords on different systems.

### 2.2.3 Buffer Overflow and Shell Coding

Programming an application is not a straight forward task. Sometimes, an unintentional mistake, weak programming, or a bug in the program's code can be used as a back door to penetrate, take control over, and exploit the system. The most common example of such threat is Buffer Overflow.

Buffer overflow is the state where you write beyond the memory boundary specified for a buffer in your program. The overwritten memory could belong to a variable, return addresses, pointers, or other important data in the program [10]. Overwriting it, will lead to wrong results, errors, crashes, or exploits in the program. Buffer overflow is common on all platforms especially in applications written in C/C++ programming language. The reason behind that is C/C++ doesn't provide boundary check for allocated memory arrays. Take the following code as an example:

```
void BufferOverflowFunction(char *String)
{
    //Buffer allocated of size 3 bytes
    char BufferToOverfLow[3];
    //Copies the contents of String buffer to another buffer until a
    //null character is reached.
    strcpy(BufferToOverfLow, String);
}

int main() {
    //Buffer allocated of size 34 bytes
    char String[34];
    //Add data to the buffer
    String = "Network security class is awesome";
    BufferOverflowFunction(String);
    return 1;
}
```

As you can see, even though BufferOverflow buffer is only three bytes long, C/C++ will allow copying data of larger size. What is of concern, however, is what happens when you overflow the buffer? Before we dig deeper into this, we need to know the exact memory layout of a process or a program to predict the behavior of a program after a buffer overflow.

When a function is called, the CPU will store the data of that function (parameters, local variables, and address of where to go after the function is executed) in a Stack Frame (SF). The SF will also contain a Stack Frame Pointer (SFP) which contains the address of a fixed location within the stack so local variables can be located relative to that location.

In *Figure 2*, we show the stack frame for the above `BufferOverflowFunction(char * String)` function:

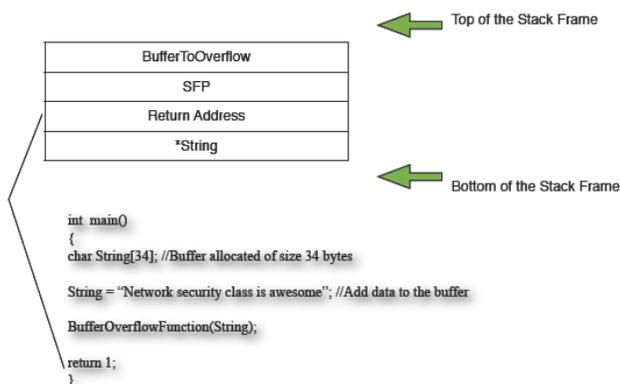


Figure 2: Stack Frame for the `BufferOverflowFunction()`

If you look at the above Stack Frame, overflowing `BufferToOverFlow` buffer will overwrite the SFP, Return Address, and `*String`. This will cause a segmentation fault error and will crash your program. Moreover, the return address can be overwritten in a way the program would jump to execute a malicious code after the execution of the program.

Here is where Shell Coding jumps in as a security threat since it can be used to exploit such mistakes to take over your system. A shell code is a piece of code that is injected into a vulnerable program to exploit your system [11]. Originally, it was called a shell code because the injected code was supposed to take over the operating system's shell command, which would give access to the kernel's functions. However, many of the current "shell codes" don't take over the shell command. Many attempts to change the naming to a more conceptually fitting name failed to succeed. Shell codes are written in machine code.

To exploit a system using the Buffer Overflow vulnerability, all what the hacker needs to do is to put the shell code in memory and overwrite the return address in the SF to point to the address of the shell code and then the shell code will automatically be executed. That's why buffer overflow is considered to be one of the most dangerous threats being faced in computer and network security.

## 2.2.4 Attacks on Encrypted Information

Sniffing network data is not a hard task to do. Hence, sharing sensitive information of a high security application in plain text will give the system information to an eavesdropper on a silver plate. Hiding information by encryption is crucial for such applications in this case. Encrypting shared information will enhance the security of the system by making it less probable for attackers to recover shared information. This doesn't mean that the system is totally

secure. Cryptography fails to claim that it is unbreakable [12]. In general, there are three methods of encryption:

*Alphabetic substitution*- is an encryption method which applies a two way one-to-one mapping on every character or ordered set of characters in the alphabet to a different character or set of characters in the same alphabet [13]. This method is hard to break using brute force since the mapping is a permutation of a different alphabetic sequence, which is very hard to guess. However, there is another way that will make breaking it a very easy task. A study was made on the average utilization relative frequency for every letter in the English alphabet. The results are shown in *Figure 3* [14]:

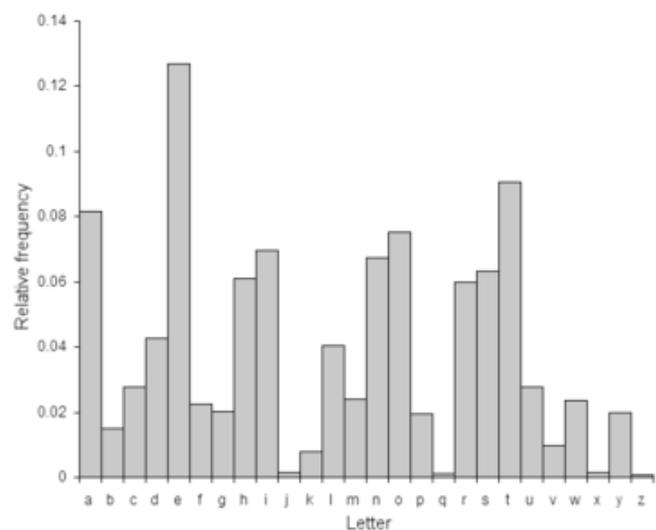


Figure 3: Letter Occurrence Frequency in English Alphabet

This information can be used to attack the Alphabetic Substitution Encryption method. The letter relative utilization frequency analysis can be applied in the same manner for the encrypted text and then use the result to guess the key. For example if we saw that the letter "t" is the most frequently used in the encrypted text then "t" is most probably "e" in the plain text. This can be applied on all other letters.

*Symmetric Key Ciphers*- is a branch of encryption algorithms that uses the same or trivially related key to encrypt and decrypt data [15]. A good example on Symmetric Key Ciphers would be the Data Encryption Standard (DES). DES is a widely used algorithm that was developed by IBM . The details of DES algorithm are out of the scope of this paper. However, we will introduce the concept to better understand Symmetric Key Ciphers. The key concept of the DES algorithm is that it divides the plain text into equally sized blocks and apply its encryption algorithm on every block [16]. It has two main block encryption mechanisms:

- 1- Electronic Code Book (ECB)- In ECB mode, DES will divide the plain text to blocks and encrypt each

block with the same key as shown in *Figure 4*. The main advantage of this algorithm mode is that it can be processed in parallel whereas the disadvantage is that identical blocks will produce the same encrypted block.

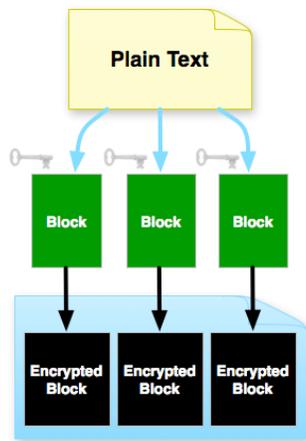


Figure 4: ECB Block Generation

- 2- Cipher Block Chaining (CBC): Similarly, CBC will divide the text into blocks. However, it XOR's each block with the encryption result of the previous block as shown in *Figure 5*. As for the first block, CBC uses an Initialization Vector (IV), which could be a pseudo randomly generated block or specified by the user. The main advantage of this algorithm mode is that looking at the encrypted text will not give any information to the attacker since the encrypted block depends on entire previous input. However, it has a disadvantage of that it is a sequential algorithm and can't be processed in parallel which makes it slower and less convenient for real time applications.

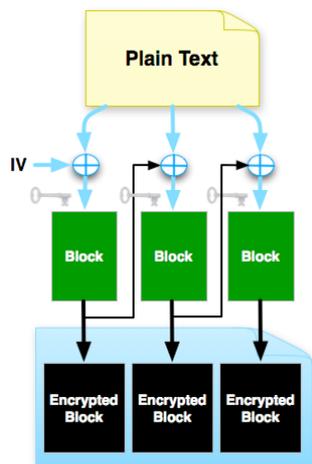


Figure 5: CBC Block Generation

After the encrypted text is transmitted, the receiving end will use the same key to decrypt the cipher. Language analysis can't be applied on this type of algorithms. However, the brute force attack is effective in breaking symmetric key Ciphers especially when the key size is small.

*Asymmetric Key Ciphers*- is another branch of encryption algorithms where the key used to encrypt data is different from the key that is later used to decrypt it [17]. The key used to encrypt data is called public key and it can be known to anyone whereas the key that is used to decrypt data is called private key and is only known to the host that is receiving the encrypted data. Asymmetric Key Ciphers are considered to be much more secure than Symmetric Key Ciphers and are the most commonly used nowadays.

### 3 Developed Laboratories

The principle of this study was initiated on the idea that says the fastest way to learn network security is by trying to break the network's security. This study was prepared to form the class note's core and lab session's experiments for introduction to network security class. The students for this class should have a background in Networking fundamentals and TCP/IP protocol. The students should also have basic C programming experience.

The laboratories were developed to accomplish several objectives. The first objective is to let students have the general background about hacking and network attacking methods. The second objective is to help students understand different methods on how to penetrate a network and detect its backdoors, open ports and any unsecured services or holes. The last objective is to teach the students how to secure the network from the discussed attacks by using a variety of defense measures.

Every lab session is ninety minutes long. At the beginning of each lab, students will be required to present chosen topics in network security that could form an introduction/support to the lab. Each lab will be directed toward a general network security problem. It will be composed of a theoretical background section and a set of exercises. The objective of the exercises is to allow students to learn the network security problem by practice. Each exercise will reveal a specific aspect of this problem. After every lab exercise, students are required to solve problems related to the given exercise and then write an explanation about this aspect of the problem and conclude how it could be resolved. In *Table 1*, we show the developed labs along with their objectives, and tools utilized in them.

Lab Name	Lab Objectives	Tools Used
<b>Introduction to labs tools and using SMTP, POP3 and FTP</b>	<ul style="list-style-type: none"> <li>• Introduction to the Vmware environment.</li> <li>• Working with the Terminal Console through Windows and Unix</li> <li>• Sending anonymous emails using the Simple Mail Transfer Protocol (SMTP) and use its command lines.</li> <li>• Receiving emails using the Post Office Protocol v3 (POP3) and use its command lines.</li> <li>• Downloading, uploading and manipulating files using a File Transfer Protocol (FTP).</li> </ul>	NS lookup (host) Sam Spade Pinger Traceroute Telnet Finger Whois/nicname Knowbot Netfind Archie Gopher

<b>Penetration Testing</b>	<ul style="list-style-type: none"> <li>• Deal with some scanner tools on windows</li> <li>• Be familiar with scanner tools that require Unix.</li> <li>• Learn how to install .rpm and .tar files in Linux.</li> <li>• Be familiar with the command line in Linux/Unix.</li> <li>• Learn how to scan targets without being traced or detected.</li> <li>• How to block ports</li> </ul>	Superscan Nmap
<b>Sniffing</b>	<ul style="list-style-type: none"> <li>• Using Sniffers</li> <li>• Learn how to apply Monkey in the Middle Attack</li> <li>• Encryption as precaution</li> </ul>	-Session Hijacking - Dsniff
<b>Denial of Service (DoS)</b>	<ul style="list-style-type: none"> <li>• Understanding the Denial of Service principle.</li> <li>• Gain knowledge of Resource Exhaustion Attacks.</li> <li>• Knowledge in IP Fragmentation Attacks method.</li> <li>• Knowledge in the Distributed Denial-of-Service (DDoS) method.</li> </ul>	Distributed DOS Syn Flood Win Nuke Smuf, snork

<p><b>Viruses and Worms</b></p>	<ul style="list-style-type: none"> <li>• Learn computer viruses types and structure.</li> <li>• Program a simple virus.</li> <li>• Learn how does an anti virus work.</li> <li>• Program an anti virus for the previously created virus.</li> <li>• Differentiate between a computer virus and worm.</li> <li>• Learn how does a computer worm work using the famous “msblaster” worm example.</li> </ul>	<p>- Borland C compiler</p>
<p><b>Password Cracking</b></p>	<ul style="list-style-type: none"> <li>• Learn password cracking techniques.</li> <li>• Learn how to attack Windows LM hash passwords using dictionary attack.</li> <li>• Learn how to attack Windows LM hash passwords using brute force attack.</li> <li>• Learn how to attack Windows LM hash passwords using hybrid attack.</li> <li>• Learn how to choose a strong password.</li> </ul>	<p>- LCP password cracker.</p>

<p><b>Buffer Overflow and Shell Coding</b></p>	<ul style="list-style-type: none"> <li>• Buffer Overflow description and exploits.</li> <li>• Buffer Overflow programming tutorial (Practice).</li> <li>• Shell Coding in theory.</li> <li>• Shell Coding programming tutorial (Practice).</li> <li>• Take home exercise: Buffer Overflow exploit using Shell Coding</li> </ul>	<p>Borland compiler or other IDA Pro</p>
<p><b>Cryptography</b></p>	<ul style="list-style-type: none"> <li>• Practice and learn Simple Encryption Using classic Caesar Algorithm</li> <li>• Practice and learn Alphabetic Substitution Encryption and Attack Using Language Analysis.</li> <li>• Practice and learn Data Encryption Standard “DES” and develop attacks on it.</li> <li>• Practice RSA Crypto systems.</li> <li>• Practice message signature generation.</li> </ul>	<p>CrypTool</p>

Table 1: Developed Labs, Objectives and Tools

By the end of the class, the students are expected to have basic understanding of general issues in network security and the approaches that can be followed to resolve them.

## 4 References

- [1] Rodger, Will. Cybercops Face Net Crime Wave. (1996, June 17). [Online]. Available at <http://www.zdnet.com/intweek/print/960617/politics/doc1.html>.
- [2] Flick, Anthony R.. Crime and the Internet. (1997, October 3). [Online]. Available at <http://www.rwc.uc.edu/bezemek/PaperW97/Flick.htm>.
- [3]. Guide to Firewalls and Network Security: Intrusion Detection and VPNs, Course Technology, by Greg Holden ISBN: 0-619-13039-3
- [4]. Intrusion Detection Systems; Definition, Need and Challenges, Sans Institute 2001.
- [5]. Computer Virus Tutorial, 2005, Computer Knowledge, <http://www.cknow.com/VirusTutorial.htm>
- [6]. Péter Ször , Peter Ferrie, Hunting For Metamorphic, Symantec Security Response
- [7]. Computer Worms Information: <http://virusall.com/worms.shtml>
- [8]. How to prevent Windows from storing a LAN manager hash of your password in Active Directory and local SAM databases, <http://support.microsoft.com/default.aspx?scid=KB;EN-US;q299656&>
- [9]. Net users picking safer passwords, [http://news.zdnet.com/2100-1009\\_22-150640.html](http://news.zdnet.com/2100-1009_22-150640.html)
- [10]. Crispin Cowan, Perry Wagle, Calton Pu, Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade\*
- [11]. Yong-Joon Park, Gyungho Lee, Repairing Return Address Stack for Buffer Overflow Protection, CF'04 April 14-16, 2004, Ischia, Italy, Copyright 2004 ACM 1-58113-741-9/04/0004
- [12]. Ross Anderson, Why Cryptosystems Fail, <http://www.cl.cam.ac.uk/~rja14/wcf.html>
- [13]. Substitution Cipher: <http://www.nationmaster.com/encyclopedia/Substitution-cipher>
- [14]. Cryptograms and English Language Letter Frequencies, <http://www.cryptograms.org/letter-frequencies.php>
- [15]. Cook, D. L. and Keromytis, A. D. 2005. Conversion and Proxy Functions for Symmetric Key Ciphers. In Proceedings of the international Conference on information Technology: Coding and Computing (Itcc'05) - Volume I - Volume 01 (April 04 - 06, 2005). ITCC. IEEE Computer Society, Washington, DC, 662-667. DOI= <http://dx.doi.org/10.1109/ITCC.2005.115>
- [16]. Walter Tuchman (1997). "A brief history of the data encryption standard". Internet besieged: countering cyberspace scofflaws: 275-280, ACM Press/Addison-Wesley Publishing Co. New York, NY, USA.
- [17]. J. Katz; Y. Lindell (2007). Introduction to Modern Cryptography. CRC Press. ISBN 1-58488-551-3.

# Formal Verification of the Security of a Free-Space Quantum Key Distribution System

Verónica Fernández, María-José García-Martínez, Luis Hernández-Encinas, and Agustín Martín

Department of Information Processing and Coding  
Instituto de Física Aplicada, (IFA-CSIC), Serrano 144, 28006- Madrid, Spain

**Abstract** - *The security of a free-space Quantum Key Distribution (QKD) system is analyzed by using PRISM, a probabilistic model checker. Disturbances and misalignments causing an imperfect channel are considered. Results show that as the channel becomes noisier the probability of Eve's detection increases. The security of the system is formally demonstrated against intercept-resend and random substitution eavesdropping attacks for a particular range of transmitted photons.*

**Keywords:** Cryptography, formal verification, probabilistic model checking, quantum key distribution.

## 1 Introduction

Security protocols are specifications of communication patterns which are intended to let agents share secrets over a public network. They are required to perform correctly even in the presence of malicious intruders who listen to the message exchanges over the network and also manipulate the system (by blocking or forging messages, for instance). Obvious desirable requirements include secrecy and authenticity. The presence of possible intruders imposes the use of symmetric and asymmetric cryptographic primitives to encrypt the communications [1].

Nevertheless, it has been widely acknowledged that even the use of the most perfect cryptographic tools does not always ensure the desired security goals. This could be either for efficiency reasons or because frequent use of certain long-term keys might increase the chance of those keys being broken by means of cryptanalysis.

Secure key agreement where the output key is entirely independent from any input value is offered by Quantum Key Distribution (QKD). Although this technique does not eliminate the need for other cryptographic protocols, such as authentication, it can be used to build systems with new security properties.

The aim of this work is to analyze the security of BB84 protocol [2] against two kinds of eavesdropping attacks (intercept-resend and random substitution attacks) when implemented in an experimental QKD system. We will consider the influence of possible disturbances in the free-space between Alice and Bob, and misalignments in the optics to calculate the probability of detection of the eavesdropper as

a function of the number of photons transmitted (or equivalently, the length of the bit sequence generated by Alice).

The rest of the paper is organized as follows. Section II includes some preliminaries and definitions. Section III briefly outlines the BB84 protocol, describes the actual free-space QKD system under development in our labs, and exposes the model checking methodology used to analyze its security. The calculated results are presented and discussed in section IV and, finally, conclusions are derived in section V.

## 2 Preliminaries

In this section, we include a short explanation about the security of QKD systems and the usefulness of formal methods to verify its security, and a description of the verification software used in this work.

### 2.1 Quantum Key Distribution security

QKD protocols provide a way for two parties, a sender, Alice, and a receiver, Bob, to share a key through a quantum communication channel (by means of optical fiber or free-space links), and detect the presence of an eavesdropper, Eve. The first complete protocol for QKD, widely used today, was BB84, which uses two non-orthogonal bases, each one with two orthogonal and linearly polarized states ( $0^\circ/90^\circ$  and  $45^\circ/-45^\circ$ , respectively) that encrypt each photon to be transmitted [2]. Later on, a simplified version, the B92 protocol, was also introduced [3].

QKD allows two distant partners to communicate with absolute security. Unlike conventional cryptography, QKD promises perfect, unconditional security based on the fundamental laws of physics, the non-cloning theorem and the uncertainty principle. The security of QKD has been rigorously proven in several papers [4]–[6], given some assumptions as can be the physical security of encoding/decoding devices, a true source of random bits, authenticated classical channel to compare bits, and reliable single photon emitters and detectors.

Unfortunately, building a practical QKD system that is absolutely secure is a substantial research challenge. The first prototype of a QKD system leaked key information over a side channel (it made different noises depending on the photon polarization) [7], and more sophisticated side channel

attacks continue to be proposed against particular implementations of existing systems [8]. Furthermore, experiments can be insecure because QKD systems in real life are generally based on attenuated laser pulses, which occasionally give out more than one photon [9].

Those multi-photon pulses enable powerful eavesdropping attacks including the Beam-Splitting (BS) attack [10], or the Photon Number Splitting (PNS) attack [11], [12]. Information leakage caused by BS attacks can be extinguished by privacy amplification [13]. To counter the PNS attack several schemes have been proposed: The non-orthogonal encoding protocol SARG04 [14], the decoy state method [15], [16], or the differential phase shift QKD [17]. Other device-independent security proofs aim to minimize the security assumptions on physical devices [18]–[20]. Very recently, several methods have been presented to blind or control the detection events in QKD distribution systems that use gated single-photon detectors [21], [22], allowing for attacks eavesdropping the full raw and secret key without increasing the Quantum Bit Error Rate (QBER).

## 2.2 Formal methods

Thus, despite the existence of a mathematical proof of the security of a given protocol, it is necessary to verify that the implementation of that protocol in a real system is secure. *Formal methods* allow this task to be developed.

Formal methods provide a mathematical representation of the security functions and the expected behavior of a given protocol or system. The two main aspects of formal methods are the language that is used to formally express the characteristics of the protocol or system (specification language), and the way to proof the correct behavior of the system according to the formal specification (formal verification). The most widely used technique to verify security protocols is *model checking* [23].

The basic idea of model checking security protocols is to build a relatively small model of a system running the protocol of interest together with a general intruder model that interacts with the protocol [24]. The model checking technique explores all possible system states to automatically test whether the system model meets the specification. The automated software tool is called a model checker.

Since quantum phenomena are inherently described by random processes, an entirely appropriate technique for verification of quantum protocols is *probabilistic* model checking [25]. Probabilistic model checking is a formal verification technique for the modeling and analysis of systems that exhibit stochastic behavior. It can be applied to several different types of probabilistic models. The three most commonly used are: Discrete Time Markov Chains (DTMCs), in which time is modeled as discrete steps, and randomness as discrete probabilistic choices; Markov Decision Processes (MDPs), which extend DTMCs with the ability to represent nondeterministic behavior; and Continuous Time Markov Chains (CTMCs) which does not permit nondeterminism but

allows specification of real (continuous) time behavior, through the use of exponential distributions [26].

## 2.3 PRISM model checker

In this work we use PRISM [27], [28] to verify the security of a free-space QKD system under development in our labs [29]. PRISM is a free and open source probabilistic model checker for formal modeling and analysis of systems which exhibit random or probabilistic behavior. It was initially developed at the University of Birmingham and now at the University of Oxford, and supports the three types of probabilistic models mentioned above, DTMCs, CTMCs, and MDPs, plus extensions of these models with costs and rewards. Models are described using the PRISM language, a simple, state-based language which subsumes several well-known probabilistic temporal logics, including Probabilistic Computational Tree Logic (PCTL), used for specifying properties of DTMCs and MDPs, and Continuous Stochastic Logic (CSL), an extension of PCTL for CTMCs. The model checker provides support for automated analysis of a wide range of quantitative properties of these models, as can be, for example, the calculation of the worst-case probability of a given protocol terminating in error, over all possible initial configurations or the probability that an enemy obtains information data on a key in a QKD protocol as a function of several parameters. It incorporates state-of-the art symbolic data structures and algorithms, based on Binary Decision Diagrams (BDDs) and Multi-Terminal Binary Decision Diagrams (MTBDDs) [30], [31]. It also features discrete-event simulation functionality for generating approximate results to quantitative analysis.

PRISM has been used to analyze systems from a wide range of application domains, including quantum protocols. BB84, assuming a perfect quantum channel, was examined using this method in [32] and [33]. Very recently the security of B92 and BB84 quantum protocols have been analyzed in [34] and [35], respectively, by considering an intercept-resend attack and by calculating the probability that an eavesdropper measures more than half the photons transmitted from Alice to Bob, taking into account the influence of quantum channel efficiency and Eve's power on the information obtained about the key. Similar approaches are presented in [36] and [37] for a standard man in the middle attack, showing results about the probability to detect the eavesdropper. The same tool has been used in [38] to study the security of BB84 protocol in the same attacking scenarios analyzed in present paper but calculating, for different key lengths, the probability of detection of the eavesdropper as a function of a parameter which represents the probability of flipping the transmitted bit in its own basis. The results of this work predict a lower chance to detect the eavesdropper in a noisy channel.

## 3 System description and methodology

The aim of this work is to verify the security of BB84 QKD protocol when implemented in a practical system. In this section we first outline the basics of the protocol. Then

we describe the experimental setup and the formal models used to simulate it.

### 3.1 BB84 protocol description

The basic BB84 protocol consists in a first phase, where quantum transmissions take place over a quantum channel and a second one, where Alice and Bob discuss over a classical channel, assumed public, which may be passively monitored (but not tampered with) by an enemy [2]. QKD uses polarized photons as information carriers. BB84 protocol uses four polarizations for the photons:  $|0\rangle$ ,  $|1\rangle$ ,  $|+\rangle$ , and  $|-\rangle$ , grouped in two non-orthogonal basis,  $\oplus$  for horizontal and vertical polarizations, and  $\otimes$ , also known as *Hadamard* basis, for diagonal polarizations. The first state of each base corresponds to the 0 classical bit value, while the second one corresponds to the 1.

During the first phase:

- a) Alice generates a random string of bits  $\mathbf{d} \in \{0,1\}^n$ , where  $n$  is the number of transmitted photons, and a random string of bases  $\mathbf{b} \in \{\oplus, \otimes\}^n$ , with  $n > K$ , where  $K$  is the length of the key.
- b) Alice sends, over the quantum channel, a photon to Bob for each bit  $d_i$  in  $\mathbf{d}$ . For each photon she randomly selects a basis  $b_i$  in  $\mathbf{b}$  with equal probability so that those photons are codified in one of the four above mentioned polarizations.
- c) Bob measures each quantum state received with respect of each one of the orthogonal basis, chosen at random. The choices of bases generate a string  $\mathbf{b}' \in \{\oplus, \otimes\}^n$  and the measurements generate the string  $\mathbf{d}' \in \{0,1\}^n$ .

During the second phase:

- a) For each bit  $d_i$  in  $\mathbf{d}$ :
  - i. Alice sends the value of  $b_i$  to Bob over a public classical channel (an asymmetric channel, for example).
  - ii. Bob responds by stating whether he used the same basis for measurements. If  $b_i' \neq b_i$ , both  $d_i$  and  $d_i'$  are discarded.
- b) Alice chooses a subset of the remaining bits in  $\mathbf{d}$  and discloses their values to Bob over the classic channel. If the results of Bob's measurements for any of these bits do not match the values disclosed, eavesdropping is detected and communication is aborted.
- c) Once the bits disclosed in previous step are removed, the remaining bits in  $\mathbf{d}$  form the final secret key.

### 3.2 Description of our QKD system

Our experimental free-space QKD setup is currently designed to implement B92 protocol at 1 GHz clock rate, and we are improving the system to also implement BB84

protocol. The transmitter in Alice's module (Fig. 1) is mounted on an aluminium base plate. It has two 850nm channels, used for the transmission of the key, and a 1550nm channel for the synchronizing signal. Those channels are combined by means of two pellicles, and the resulting beam is expanded with an output telescope, formed by lenses  $L_1$  and  $L_2$ , so that it produces a 40mm-diameter diffraction limited spot. The expansion of the beam is made to allow a long-distance transmission without large beam divergences.

The receiver module, Bob, is placed at a distance of 40 m from Alice during the preliminary tests (3 km in the final system is expected) and, therefore, it receives a diverging beam. To efficiently detect the beam a Schmidt-Cassegrain telescope of 25.4 cm diameter, 2.5 m equivalent focal distance and fine-pointing capability is used. Bob's optics has been designed to be coupled to the output of the telescope by using lightweight and compact mounts (see Fig. 2). The output of the telescope is connected to Bob's optics and the outputs of Bob's channels are connected to two single-photon detectors by using optical fiber. The optical synchronization pulse is detected by an avalanche photodiode. The outputs of all three detectors are connected to an electronic card which is able to measure the time of arrival of the photons with high temporal precision. This information is then sent to Alice from which she can infer which key bits have been received by Bob.

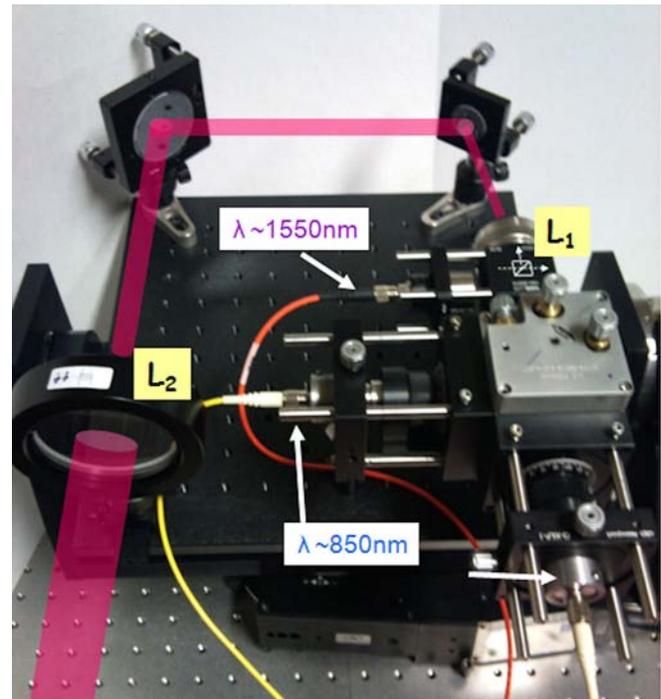


Fig. 1. Current Alice's setup (implementing B92).

Especially care must be paid to one of the most critical parts of the system, the filtering of the solar background radiation. For this purpose, a combination of spectral, spatial, and software filtering are used. The spatial filtering is carried out by optical fiber (Fig. 2). A good compromise of the

diameter of this fiber must be found, as small diameters improve the filtering of the solar radiation at the expense of higher signal losses. In addition, if the diameter is too small the signal could be lost due to the beam wandering caused by the fluctuations of the index of refraction of the air.

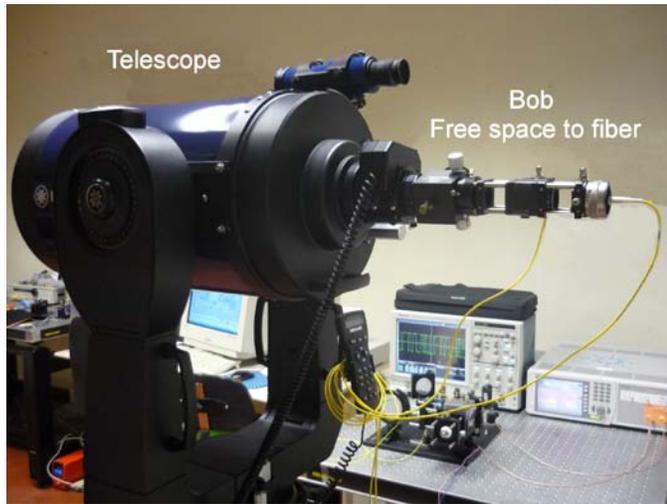


Fig. 2. Bob's optics at the output of the receiver telescope, coupling input beam to optical fiber.

A non-optimal filtering of the solar radiation can be a typical source of noise. In addition, a not optimal alignment between Alice and Bob, variations in the atmospheric conditions and/or difficulties in the coupling losses can make the channel imperfect, and should be considered in order to formally verify the security of the whole system.

Although Fig. 1 and Fig. 2 show our current experimental setup implementing the B92 protocol, all comments in previous paragraph about sources of noise and imperfections are also valid for the BB84 protocol which will also be implemented as an improvement to our system. For this reason this is the protocol we simulate in this work.

### 3.3 Formal models

In order to verify the security of the system described above we have to model it in a description language and express its desired properties by means of a formula written in a given logic. The model and the formula are the input to PRISM, that will compute the probability with which that particular formula is satisfied by the simulated model.

According to the experimental setup and the protocol described previously in subsection B, we have simulated the QKD system in PRISM language. The modeling is probabilistic, DTMC, and we have analyzed the probability to detect Eve as a function of the channel efficiency and the number of transmitted photons (which are assumed to be linearly related to the length of the key).

1) System model ( $M$ ): Four modules have been built to consider Alice, Bob, Eve, and a communication channel which

can be imperfect due to disturbances and misalignment losses. All those modules have three local variables, corresponding respectively to the computational state, the basis with respect to which a photon is encoded, and the bit value which is being encoded. A fourth variable is added to Alice module to simulate the transmission of  $N$  photons (each one encoding a bit value) as the iteration for  $N$  times of the transmission of a single photon in a given state.

2) Desired property: The presence of an eavesdropper must be detected by the protocol users. If  $\Phi$  is a formula corresponding to the event that an eavesdropper is detected, the probability of this event in our model  $M$  is:

$$P_{\text{detection}} = P_r \{M(N, P_C) \text{ satisfy } \Phi\} \quad (1)$$

where  $P_C$  is the probability that Eve obtains the correct bit value although an incorrect basis is chosen for her measurement, and  $\Phi = \text{true} \cup \text{Bobstate} = V$ ,  $V$  being the value assigned in the program to the state of Bob when Eve is detected.

3) Attacks: Two different attacks are considered: A typical intercept-resend attack [32] and a random substitution attack [33]. In the first one, which is the most widely simulated eavesdropping attack, we have introduced nondeterminism for Alice, Bob and Eve, and we have simulated Bob's behavior so that a comparison is made between his variable of basis and that of the channel before Alice reveals her basis; if both values are different, then the value of the bit variable in Bob module is updated with the value of the bit variable in channel module (0 or 1) with a probability  $P_C$ , and with the other bit value (1 or 0) with a probability  $1-P_C$ . In the same way, Eve's behavior is simulated so that if the value of her variable of basis coincides with that of the channel, she gets the right bit. Otherwise the result she gets is random, as predicted by quantum theory.

In the random substitution attack, the eavesdropper chooses a basis  $b_i$  at random, and also a random data bit  $d_i$ ; she substitutes the  $i$ -th photon (which encodes bit  $d_i$  in  $b_i$  basis) with a new photon which represents  $d_i$  bit in  $b_i$  basis. In our program, Eve replaces a 0 bit on the channel with a probability defined by a variable called SUBS, and a 1 bit with a probability  $1-\text{SUBS}$ . The same probabilities are used to replace channel bases.

## 4 Results

We have computed the probability of detection of an eavesdropping while performing the two above mentioned attacks. For each one, we have studied the variation of  $P_{\text{detection}}$  as a function of the number of transmitted photons. Several calculations have been made, varying the value of  $P_C$  (we have considered values from  $P_C = 0$  to  $P_C = 0.9$  in steps of 0.15), and simulating possible channel inefficiencies by the inclusion of a noise parameter in the channel module.

## 4.1 Intercept-resend attack

Fig. 3 shows the probability of detection of an eavesdropper in the BB84 protocol as a function of the number of photons transmitted. The channel is assumed without noise and a comparison is made between the plots obtained for different values of the parameter  $P_C$ .

As can be observed, the value of  $P_C$  highly influences the probability of detection of the eavesdropper when there is no noise in the channel. In fact, if the number of photons transmitted is greater than 25, the probability of detecting the eavesdropper is higher than 0.9, except if  $P_C = 0.9$ .

Channel module in PRISM is modified in order to simulate a noisy channel so that the probability of the information sent by Alice (base and bit) remain unchanged before being received by Eve is 40%. Calculations are repeated and results are shown in Fig. 4.

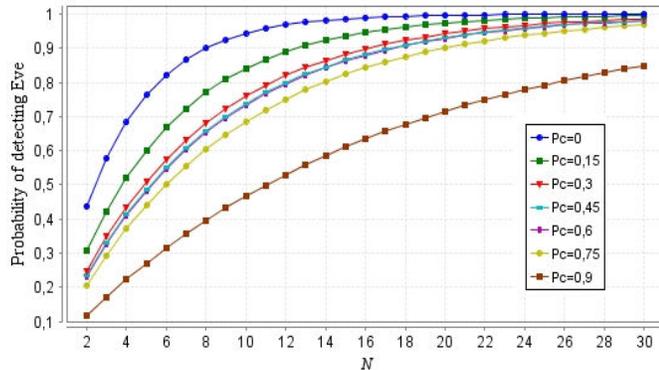


Fig. 3. Probability of detection of Eve as a function of the number of photons emitted for different values of  $P_C$ , when a noiseless channel is considered.

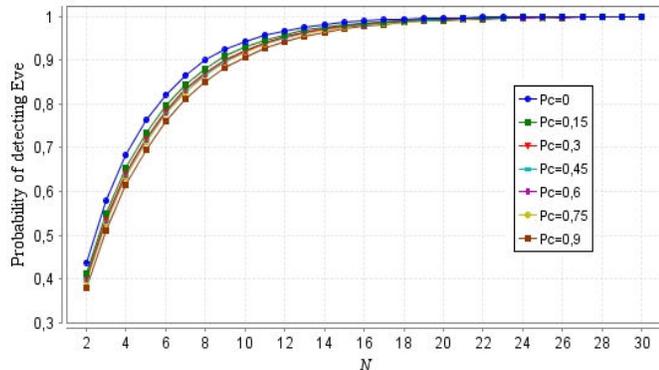


Fig. 4. Probability of detection of Eve as a function of the number of photons emitted for different values of  $P_C$ , when a noisy channel is considered.

In this case, i.e., if the channel is noisy, the eavesdropper is detected with a probability higher than 0.9 if only 10 photons are transmitted, for all values of  $P_C$ .

A comparison of Fig. 3 and Fig. 4 reveals that in a noisy channel the value of the probability that Eve obtains the correct bit value although an incorrect basis is chosen for her

measurement has almost negligible influence in the probability of detection of the attack. Moreover, in the presence of noise the probability of detection of Eve increases. This result is similar to that presented in [37], although it differs from what is concluded in a very recent paper [38].

## 4.2 Random substitution attack

As for the previous attack, the probability of detection of Eve as a function of the number of photons transmitted in a channel without noise is shown in Fig. 5 for different values of the  $P_C$  parameter.

It can be noted that, in this case, there is almost no difference between the calculated probabilities for different values of  $P_C$ . When calculations were repeated considering a noisy channel the values obtained were the same (shown as a wide green line in Fig. 5). In this simulation, if the number of transmitted photons is greater than 10, the probability that Eve is detected is higher than 0.9, for each value of  $P_C$  considered.

This result indicates that the random substitution attack produces a high probability of Eve's detection regardless the channel noise (as could be expected, because in this scenario Eve's behavior is similar to the way how noise, at random, modifies the transmitted bits).

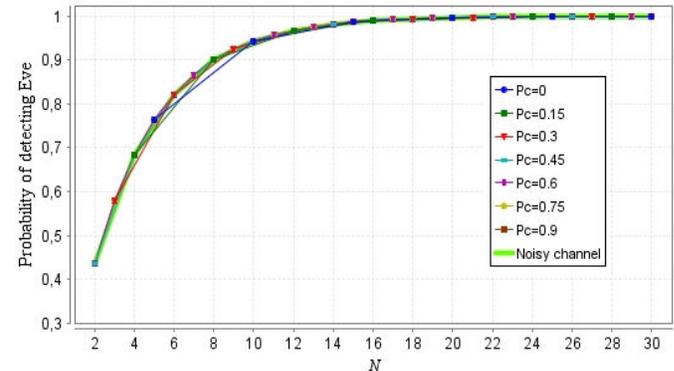


Fig. 5. Probability of detection of Eve as a function of the number of photons emitted for different values of  $P_C$  in a channel without noise. Results for a noisy channel are also shown.

By comparing Fig. 3 with Fig. 5 it can be observed that, in a perfect channel, Eve is more likely to be detected if she uses a random substitution attack, even with small values of  $N$ . In presence of noise or imperfections in the operating devices the probability of detecting the eavesdropper is quite similar for both attacks.

## 5 Conclusions

In this work, the interest of formally verifying the security of an experimental QKD system, by describing possible problems which can cause imperfections in the quantum channel, has been pointed out. By using a probabilistic model checker, the probability of detecting an eavesdropper is calculated for both an intercept-resend attack

and a random substitution attack. Results show that as the channel becomes noisier the probability of Eve's detection increases.

**Acknowledgments:** Manuscript received March 30, 2011. This work was supported in part by the Ministerio de Ciencia e Innovación (Spain), project no. MTM2008-02194 and Ministerio de Industria, Turismo y Comercio (Spain), project no. TSI-020100-2009-44.

## 6 References

- [1] A. S. Khan, M. Mukund, and S. P. Suresh, "Generic verification of security protocols," *Lecture Notes in Comput. Sci.*, vol. 3639, pp. 221–235, 2005.
- [2] C. H. Bennett and G. Brassard, "Quantum Cryptography: Public key distribution and coin tossing," in *Proc. IEEE Int. Conf. Comput. Syst. Signal Process.*, pp. 175–179, 1984.
- [3] C. H. Bennet, "Quantum Cryptography using any two nonorthogonal states," *Phys. Rev. Lett.*, vol. 68, pp. 3121–3124, 1992.
- [4] P. W. Shor and J. Preskill, "Simple Proof of Security of the BB84 Quantum Key Distribution Protocol," *Phys. Rev. Lett.*, vol. 85, pp. 441–444, 2000.
- [5] D. Mayers, "Unconditional security in quantum cryptography," *J. ACM*, vol. 48, pp. 351–406, 2001.
- [6] H. K. Lo and H. F. Chau, "Unconditional security of quantum key distribution over arbitrarily long distances," *Science*, vol. 283, no. 5410, pp. 2050–2056, 1999. Available: doi:10.1126/science.283.5410.2050. eprint arXiv:quant-ph/9803006.
- [7] G. Brassard, "Brief history of quantum cryptography: A personal perspective," eprint arXiv:quant-ph/0604072, 2006.
- [8] Y. Zhao, C. H. F. Fung, B. Qi, Ch. Chen, and H.K. Lo, "Experimental demonstration of time-shift attack against practical quantum key distribution systems," eprint arXiv:0704.3253v2, March 2008.
- [9] M. Jing-Long, W. Fa-Qiang, L. Qing-Qun, and L. Rui-Sheng, "Practical non-orthogonal decoy state quantum key distribution with heralded single photon source," *Chinese Physics B*, vol. 17, no. 4, pp. 1178–06, April 2008.
- [10] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental quantum cryptography," *J. Cryptol.*, vol. 5, pp. 3–28, 1992.
- [11] G. Brassard, N. Lütkenhaus, T. Mor, and B. C. Sanders, "Limitations on practical quantum cryptography," *Phys. Rev. Lett.*, vol. 85, no. 6, pp. 1330–1333, 2000.
- [12] N. Lütkenhaus and M. Jarma, "Quantum key distribution with realistic states: photon-number statistics in the photon-number splitting attack," *New J. Phys.*, vol. 4, pp. 44-1–44-9, 2002.
- [13] C. H. Bennett, G. Brassard, C. Crepeau, and U. M. Maurer, "Generalized privacy amplification," *IEEE Trans. Inf. Theory*, vol. 41, no. 6, part 2, pp. 1915–1923, November 1995.
- [14] V. Scarani, A. Acín, G. Ribordy, and N. Gisin, "Quantum Cryptography Protocols Robust Against Photon Number Splitting Attacks For Weak Laser Pulse Implementations," *Phys. Rev. Lett.*, vol. 92, no. 5, 057901, 2004.
- [15] W. Y. Hwang, "Quantum key distribution with high loss: Toward global secure communication," *Phys. Rev. Lett.*, vol. 91, no. 5, 057901, 2003. eprint arXiv:quant-ph/0211153.
- [16] J. W. Harrington, J. M. Ettinger, R. J. Hughes, and J. E. Nordholt, "Enhancing practical security of quantum key distribution with a few decoy states," 2005. eprint arXiv: quant-ph/0503002.
- [17] Z. Feng, F. Ming-Xing, L. Yi-Qun and L. Song-Hao, "Differential-phase-shift quantum key distribution," *Chinese Physics*, vol. 16, pp. 3402–3406, November 2007.
- [18] D. Mayers and A. C. Yao, "Quantum cryptography with imperfect Apparatus," in *Proc. 39th Ann. IEEE Symp. Foundations of Comp. Sci.*, pp. 503–509. IEEE Press, 1998. Available: doi:10.1109/SFCS.1998.743501. eprint arXiv:quant-ph/9809039.
- [19] A. Acin, N. Brunner, N. Gisin, S. Massar, S. Pironio, and V. Scarani, "Device-independent security of quantum cryptography against collective attacks," *Phys. Rev. Lett.*, vol. 98, no. 2, 230501, 2007. Available: doi:10.1103/PhysRevLett.98.230501. eprint arXiv:quant-ph/0702152.
- [20] D. Gottesman, H.K. Lo, N. Lütkenhaus, and J. Preskill, "Security of quantum key distribution with imperfect devices," *Quantum Inform. Comput.*, vol. 4, no. 5, pp. 325–360, September 2004. Available: <http://www.rinton.net/xqic4/qic-4-5/325-360.pdf>. eprint arXiv:quant-ph/0212066.
- [21] L. Lydersen, C. Wiechers, C. Wittmann, D. Elser, J. Skaar and V. Makarov, "Thermal blinding of gated detectors in quantum cryptography," *Opt. Exp.*, vol. 18, no. 26, pp. 27938–27954, 2010.
- [22] C. Wiechers, L. Lydersen, C. Wittmann, D. Elser, J. Skaar, Ch. Marquardt, V. Makarov, and G. Leuchs, "After-gate attack on a quantum cryptosystem," *New Journal of Physics*, vol. 13, 013043, 14 pp. 2011.
- [23] C. Baier and J. P. Katoen, *Principles of model checking*. The MIT Press, Cambridge, MA, USA, 2008.
- [24] S. Basagiannis, P. Katsaros and A. Pombortsis, "Synthesis of attack actions using model checking for the verification of security protocols," *Secur. Comm. Networks*, vol. 4, no. 2, pp. 147–161, February 2011.
- [25] S. J. Gay, R. Nagarajan, and N. Papanikolaou, "Probabilistic Model-Checking of Quantum Protocols", arXiv:quant-ph/0504007, April 2005.
- [26] M. Duflot, M. Kwiatkowska, G. Norman, D. Parker, S. Peyronnet, C. Picaronny, and J. Sproston, "Practical Applications of Probabilistic Model Checking to Communication Protocols," In S. Gnesi and T. Margaria (eds.), *FMICS Handbook on Industrial Critical Systems*, IEEE Computer Society Press. Ch. 7, 2010. Available: <http://eprints.gla.ac.uk/39594/1/fmics-chapter.pdf>
- [27] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: A tool for automatic verification of probabilistic systems," *Lecture Notes in Comput. Sci.*, vol 3920, pp 441–444, 2006.
- [28] PRISM web site. [www.prismmodelchecker.org](http://www.prismmodelchecker.org).

- [29] M. J. García Martínez, D. Arroyo, N. Denisenko, D. Soto, A. Orúe, and V. Fernández, "High-speed free-space quantum key distribution system for urban applications," in *Proceedings of Photon10*, pp. 276, Southampton, UK, August, 2010.
- [30] M. Kwiatkowska, G. Norman, and D. Parker, "Probabilistic Symbolic Model Checking with PRISM: A Hybrid Approach," *Int. J. Soft. Tools Techn. Transfer*, vol. 6, no. 2, pp. 128–142, September 2004.
- [31] D. Parker, "Implementation of Symbolic Model Checking for Probabilistic Systems", Ph.D. thesis, University of Birmingham. August 2002. Available: <http://www.prismmodelchecker.org/papers/davesthesis.pdf>
- [32] R. Nagarajan, N. Papanikolaou, G. Bowen, and S. Gay. "An automated analysis of the security of quantum key distribution," In Proc. Third International Workshop on Security Issues in Concurrency (SECCO'05), San Francisco, USA, August, 2005. Available: <http://www.dcs.warwick.ac.uk/~nikos/downloads/nrgsecco05.pdf>
- [33] N. Papanikolaou, *Techniques for design and validation of quantum protocols*, M. Sc. Thesis, University of Warwick (UK), 2004. Available: <http://www.dcs.warwick.ac.uk/~nikos/downloads/npmsthesis.pdf>.
- [34] M. Elboukhari, M. Azizi, and A. Azizi, "Applying Model Checking Technique for the Analysis of B92 Security," *J. Computing*, vol. 2, no. 9, pp. 50–56, September 2010.
- [35] M. Elboukhari, M. Azizi, and A. Azizi, "Verification of Quantum Cryptography Protocols by Model Checking," *Int. J. Network Security & Appl.*, vol 2, no 4, pp. 43–53, October 2010. Available: <http://aircse.org/journal/nsa/1010ijnsa04.pdf>
- [36] M. Elboukhari, M. Azizi, and A. Azizi, "Analysis of Quantum Cryptography Protocols by Model Checking," *Int. J. Universal Comput. Sci.*, vol 1, pp. 34–40, 2010. Available: <http://www.hypersciences.org/IJUCS/Iss.1-2010/IJUCS-4-1-2010.pdf>
- [37] M. Elboukhari, M. Azizi, and A. Azizi, "Analysis of the Security of BB84 by Model Checking," *Int. J. Network Security & Appl.*, vol 2, no 2, pp. 87–98, April 2010. Available: <http://aircse.org/journal/nsa/0410ijnsa7.pdf>.
- [38] A. M. Tavala, S. Nazem, and A. A. Babaei-Brojeny, "Verification of Quantum Protocols with a Probabilistic Model-Checker," *Elect. Notes Theor. Comput. Sci.*, vol. 270, no 1, pp. 175–182, 2011.

# Cyber Security Considerations in the Development of I&C Systems for Nuclear Power Plants

Jung-Woon Lee, Cheol-Kwon Lee, Jae-Gu Song, and Dong-Young Lee  
I&C and HF Research Division, Korea Atomic Energy Research Institute,  
Daejeon, The Republic of Korea

**Abstract** - Digital technologies have been applied recently to the I&C systems of nuclear power plants. According to this application, cyber security concerns are increasing in nuclear facilities as in IT industries and other process industries. Many reports and standards are issued for cyber security in industrial control systems. Nuclear regulatory requirements based on the standards for industrial control systems have also been announced. However, it does not clearly indicate what I&C system developers should consider in their development. It is suggested that developers consider 1) maintaining a secure development environment during the development of I&C systems and 2) developing the systems to have security features necessary for a secure operation within the operation environment of NPPs in accordance with a secure development process.

**Keywords:** Nuclear Power Plant, I&C system, Cyber Security, Development Environment

parameters, integrate sensor information, monitor plant performance, and generate signals to control plant devices for NPP operation and protection. Although the application of digital technology to industrial control systems started a few decades before, the I&C system in NPPs have utilized analog technology longer than any other industries. The reason for this stems from NPPs requiring strong assurance for safety and reliability. In recent years, however, digital I&C systems have been developed and applied to the construction of new NPPs and the upgrades of operating NPPs. Fig. 1 shows a typical configuration of the digital I&C system. The safety systems are placed on the left half in Fig.1 and the non-safety systems on the right half. The NPP I&C system has similar constituents and structure to those of control systems in other industries except the safety systems. The safety systems function to shutdown the reactor safely and maintain it in a shutdown condition. The safety systems require higher reliability, functionality, and availability than the non-safety systems.

## 1 Introduction

The instrumentation and control (I&C) systems in nuclear power plants (NPPs) collect sensor signals of plant

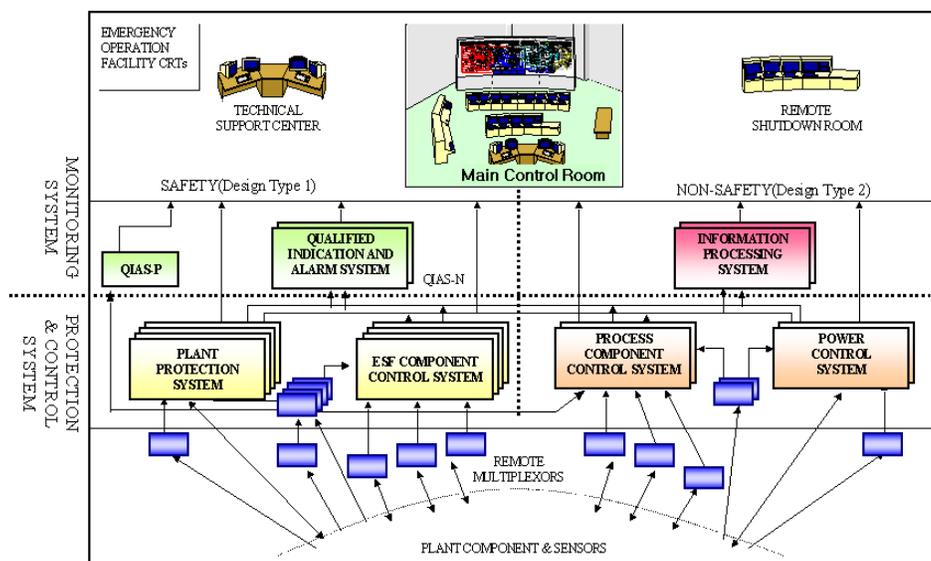


Fig.1 A typical configuration of I&C system in NPPs

Digital I&C systems in NPPs possess cyber security problems as industry control systems do. Reports by Idaho National Laboratory (INL) [1] and the U. S. Department of Homeland Security (DHS) [2] point out the following security problems arising by introducing IT component into control systems;

- Increasing dependency on automation and control systems
- Insecure connectivity to external networks
- Usage of technologies with known vulnerabilities, creating previously unseen cyber risk in the control domain
- Lack of a qualified cyber security business case for industrial control system environments
- Some control system technologies have limited security and are often only enabled if the administrator is aware of the capability (or the security does not impede the process)
- Control system communications protocols are absent of basic security functionality (i.e., authentication, authorization)
- Considerable amount of open source information is available regarding control system configuration and operations.

The following issues need more consideration as the INL report [1] suggests and the DHS report [2] added some more issues;

- Backdoors and holes in the network perimeter
- Devices with little or no security features (modems, legacy control devices, etc.)
- Vulnerabilities in common protocols
- Attacks on field devices
- Database attacks
- Communications hijacking and 'Man-in-the-middle' attacks
- Improper or nonexistent patching of software and firmware
- Insecure coding techniques
- Improper cyber security procedures for internal and external personnel
- Lack of control systems specific mitigation technologies

Among these, 'Man-in-the-middle' attack is exceptionally dangerous since attackers may do the following;

- Stop operations
- Capture, modify, and replay control data
- Inject inaccurate data to falsify information in key databases, timing clocks, and historians
- Replay normal operational data to the operator HMI while executing a malicious attack on the field device (while preventing the HMI from issuing alarms).

The report prepared by the U.S. General Accounting Office (GAO) [3] states that the following factors have contributed to the increment of risks by cyber threats specific to control systems;

- adoption of standardized technologies with known vulnerabilities,
- connectivity of control systems with other networks,
- insecure remote connections, and
- widespread availability of technical information about control systems.

And possible actions by cyber attacks may include;

- disrupting the operation of control systems by delaying or blocking the flow of information through control networks, thereby denying availability of the networks to control system operators;
- making unauthorized changes to programmed instructions in PLCs, RTUs, or DCS controllers, changing alarm thresholds, or issuing unauthorized commands to control equipment, which could potentially result in damage to equipment (if tolerances are exceeded), premature shutdown of processes (such as prematurely shutting down transmission lines), or even disabling control equipment;
- sending false information to control system operators either to disguise unauthorized changes or to initiate inappropriate actions by system operators;
- modifying the control system software, producing unpredictable results; and
- interfering with the operation of safety systems.

The North American Electric Reliability Council (NERC) listed top 10 vulnerabilities of control systems and recommended mitigation strategies [4]. The top 10 vulnerabilities are quoted as follows;

1. Inadequate policies, procedures, and culture that govern control system security,
2. Inadequately designed control system networks that lack sufficient defense-in-depth mechanisms,
3. Remote access to the control system without appropriate access control,
4. System administration mechanisms and software used in control systems are not adequately scrutinized or maintained,
5. Use of inadequately secured wireless communication for control,
6. Use of a non-dedicated communications channel for command and control and/or inappropriate use of control system network bandwidth for non-control purposes,
7. Insufficient application of tools to detect and report on anomalous or inappropriate activity,
8. Unauthorized or inappropriate applications or devices on control system networks,
9. Control systems command and control data not authenticated, and
10. Inadequately managed, designed, or implemented critical support infrastructure.

These vulnerabilities contain both managerial and technical ones. Among these vulnerabilities, item 5 'wireless communication for control' is seldom used in NPPs, but other vulnerabilities are very common to NPPs.

DHS assessed industrial control systems and listed common cyber security vulnerabilities categorized by software, configuration, and network in technical detail [5]. In Special Publication 800-82 of the National Institute of Standards and Technology (NIST), "Guide to Industrial Control Systems (ICS) Security [6]," vulnerabilities in industrial control systems are well identified in the categories of policy and

procedure, platform configuration, platform hardware, platform software, platform malware protection, network configuration, network hardware, network perimeter, network monitoring and logging, communication, and wireless connection.

There are many standards and guidelines available for mitigating the cyber security vulnerabilities of industrial control systems. Nuclear regulation requirements are established based on these standards and guidelines for industrial control systems. In this paper, nuclear regulation requirements are discussed for cyber security considerations when developing the I&C systems in NPPs.

## 2 Nuclear regulatory requirements

As cyber security has been an emerging concern in nuclear industries, the U.S. NRC issued the regulatory guide (RG) 1.152 revision 2, "Criteria for Use of Computers in Safety Systems of Nuclear Power Plants," in 2006 [7]. This regulatory guide addresses cyber security for the use of digital computers in the safety systems of NPPs. It describes regulatory position by using the waterfall lifecycle phases which consist of the following phases:

- 1) Concepts;
- 2) Requirements
- 3) Design
- 4) Implementation
- 5) Test
- 6) Installation, Checkout, and Acceptance Testing
- 7) Operation
- 8) Maintenance
- 9) Retirement.

It is required that the digital safety system development process should address potential security vulnerabilities in each phase of the digital safety system lifecycle.

In 2009, 10 CFR 73.54, "Protection of Digital Computer and Communication Systems and Networks [8]," requires NPP licensees in U. S. to submit a cyber security plan for protecting critical digital assets (CDAs) associated with the following categories of functions, from cyber attacks: 1) safety-related and important-to-safety functions, 2) security functions, 3) emergency preparedness functions, including offsite communications, and 4) support systems and equipment which, if compromised, would adversely impact safety, security, or emergency preparedness functions.

The RG 5.71 [9] was issued in 2010 for applicants and licensees to comply with the requirements of 10 CFR 73.54. This regulatory guide applies to operating NPPs and to an application for a combined operating license. RG 5.71 provides a framework to aid in the identification of CDAs categorized in 10 CFR 73.54, the application of a defensive architecture, and the collection of security controls for the protection of CDAs from cyber threats. Guidance in RG 5.71

on a defensive architecture and a set of security controls based on standards provided in NIST SP 800-53 [10] and NIST SP 800-82 [6].

The issuance of RG 5.71 brought a need for the revision of RG 1.152 due to the duplication on cyber security matters. Draft regulatory guide DG-1249 [11] for RG 1.152 revision 3 was issued for review in 2010. This regulatory guide was introduced in the NPIC&HMIT conference on November 2010 [12]. RG 1.152 revision 3 aims to eliminate reference to cyber-security and also gives directions to evaluate systems against intentional malicious actions or attacks. RG 1.152 revision 3 is clarifying its focus on: 1) Protection of the development environment from inclusion of undocumented and unwanted code, 2) Protection against undesirable behavior of connected systems, and 3) Controls to prevent inadvertent access to the system. In other words, the conference paper [12] describes these as 1) Secure Development Environment, 2) Secure Operational Environment - Independence from Undesirable Behavior of Connected Systems, and 3) Secure Operational Environment - Control of Access.

RG 1.152 revision 3 also contains a regulatory position regarding the 5 waterfall lifecycle phases from 1) Concepts to 5) Test, which are narrowed from the 9 phases in RG 1.152 revision 2. The phases after 6) Installation, Checkout, and Acceptance Testing, regulations are handed over to RG 5.71.

## 3 Considerations for secure I&C system development

Most cyber security suggestions are focused on the protection of control systems against cyber attacks in an operational environment. Articles on cyber security in a development environment can hardly be found. RG 1.152 revision 3 specifies the importance of a secure development environment in the development of safety systems in NPPs.

Cyber attacks may target the development environment too. For instance, attackers may try to insert malicious codes into the systems under development which will later be installed in NPPs or collect design information on the critical systems for later cyber attacks.

It could be argued that tests for the end products would be enough to achieve acceptable security without maintaining a secure development environment. This argument seems right since maintaining a secure development environment may cause more development expenses. However, tests may not detect all the residual weaknesses or cannot cover all the possible events, which may be triggered by one or combinations of the residual weaknesses. Considering defense-in-depth concepts in the development, maintaining a secure development environment is necessary together with performing the tests.

As shown in Fig. 2, the system to be securely developed and protected from a cyber attack is placed in a development environment during the development phase and in the operational environment after site installation.

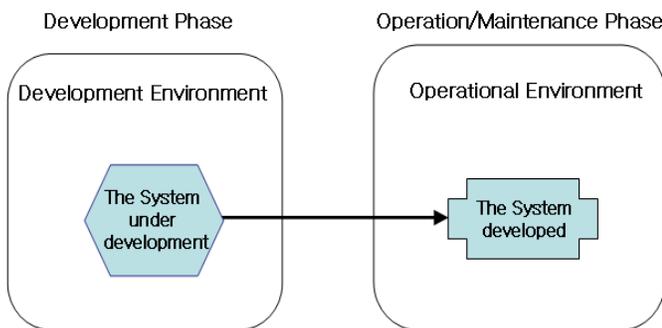


Fig. 2 System and environment in the development phase and the operation phase

Developing the I&C systems which are secure up to an acceptable level can be achieved by considering the following two matters; 1) maintaining a secure development environment and 2) the development of right security features of the I&C systems.

### 3.1 Maintaining a secure development environment

In this discussion, the environment includes hardware such as computers, networks and other digital elements, and software such as the operating system, application program, software libraries, software development tools, etc.

In order to maintain a secure development environment, an ordinary loop of cyber security activities, which is consisted of 'assessment,' 'implementation,' and 'maintenance,' should be applied.

#### 3.1.1 Assessment of the development environment

During the concept phase of the development process, developers review the digital assets of their development environment and their impact on the system to be developed (STD) within the development environment. Connectivities between digital assets in the development environment and relations of digital assets to the STD should be analyzed. Vulnerabilities in any links are assessed in terms of risk levels that may be imposed to the STD. During this assessment, the STD becomes a critical asset to be protected by the security program. The STD can be also a system containing many digital assets. In this assessment, the digital assets of the STD are also reviewed in accordance with their criticality.

#### 3.1.2 Implementation of security measures

Developers may determine security measures suitable to mitigate vulnerabilities identified in the assessment and implement them to make the development environment and

STD secure. After the implementation, the security measures should be validated and tested to ensure the measures increase the security levels at vulnerable points up to intended levels.

#### 3.1.3 Maintenance of security

The configuration of STD may change in accordance with the various development phases, such as planning, requirement development, design, implementation, and testing, and also the configuration of development environment changes. In many cases, the testing environment may have differences in the configuration from that of the development environment.

If there would be any changes in the development environment or in the configuration of STD, the change may affect the assessment results obtained previously. Hence, the assessment results may be analyzed again by focusing on the changes to find new vulnerable points within the development environment or in the STD. Security measures being implemented into the STD during the development phase can be temporal in some cases.

It is important to keep the assessment for the current status of the development environment. For this purpose, continuous monitoring of the development environment or a carefully designed monitoring program may be required.

#### 3.1.4 Security policy and plan

I&C system developers should prepare cyber security policies, plans, procedures, and organizations to perform the activities described above appropriately so that they can achieve the goals of maintaining a secure development environment.

### 3.2 Development of right security features of the I&C systems

#### 3.2.1 General cyber security considerations

Draft regulatory guide DG-1249 distinguished secure system development from cyber security provisions in the development. Security as part of safety review under 10 CFR 50 refers to protective actions taken against a predictable set of non-malicious acts that could challenge the integrity, reliability, or functionality of digital safety systems. Cyber security refers to those measures and controls, implemented to comply with 10 CFR 73.54, to protect digital systems against malicious cyber threats. DG-1249 specifies regulatory requirements for the safety systems during the development phase, and RG 5.71 in compliance with 10 CFR 73.54 describes the guides for the operation and maintenance phase in NPP sites. Cyber security features should be designed and implemented during the development phase before a site application of the system, because any later treatment on the systems for cyber security after the development may cause

other defects in the systems or may be implemented with less effective security measures.

DG-1249 requires independence from undesirable behavior of connected systems and control of access for the establishment of a secure operational environment. Undesirable behavior of connected systems can occur by either non-malicious or malicious acts and control of access is a common measure in the cyber security domain. From system developers' point of view, no significant differences have been assumed in the process, methods, and measures for handling the vulnerabilities when the system confronts either non-malicious behavior or malicious acts. Discriminating between non-malicious behavior and malicious acts may double the system developers' efforts. This paper suggests that the developers address cyber security in their development in parallel with considering protection of the system from non-malicious acts. IEEE Std. 7-4.3.2-2010[13], which is recently updated from the 2003 version, also mentions that the digital safety systems/equipment development process shall address potential security vulnerabilities in each phase of the digital safety system lifecycle and system security features should be addressed appropriately in the lifecycle phases.

Cyber security design features included in a STD should be determined based on the assessment on the system in the operational environment of NPP sites. RG 5.71 requires an analysis of critical digital assets (CDA) in the digital environment of NPP sites. The developed system will be integrated with other systems and installed at the site. I&C system developers can estimate a position of their system within the site's digital environment. When the developers perform the asset analysis, a scope of the analysis includes the system and the interfaces with other digital assets of the plant. Based on the vulnerabilities identified by the analysis, the developers can design, implement, and test cyber security design features needed for the system.

RG 5.71 provides a reference practice for a cyber security program. The developers can use this guide document to establish their cyber security policy, plan, procedures, and appropriate measures, selecting items described in RG 5.71 that corresponds to the system they develop.

### 3.2.2 Recommended cyber security activities in the design process

Fig. 3, which is redrawn from NUREG-0800 Ch. 7.0 [14], shows a general lifecycle process of I&C systems in NPPs. Although many development activities are presented in Fig. 3, they can be grouped into three stages, according to the involved organizations, which are (1) system design(SD), (2) component design and equipment supply(CD/ES), and (3) operation and maintenance. An SD company produces system design documents to hand over to a CD/ES company. Then, the CD/ES company implements hardware, software, and user interface things, integrates them, and installs the system in an NPP. A utility company who owns the NPP operates the

system in its NPP site. This paper concentrates on recommending cyber security activities for the SD and CD/ES stages.

Cyber security features should be incorporated in a system design in the SD stage. Hence, cyber security activities should be performed in the SD stage. System design specifications produced in an SD stage are translated into hardware design specifications for purchase and/or manufacturing and software design specifications for implementation during the CD/ES stage. The design in the CD/ES stage become more concrete and detail than the design in the SD stage. It would be better to assess again cyber security characteristics in the hardware and software design during the CD/ES stage. Also in the CD/ES stage, decisions can be made on which 3rd party products or commercial off-the-shelf(COTS) items are utilized in the development. Cyber security characteristics of these 3rd party products or COTS items should be assessed in the CD/ES stage. After the completion of hardware and software design, hardware is assembled and software coding is implemented, then these are integrated and tested. At this time, vulnerability scanning and security testing can be performed with the manufactured systems.

It is important that system functionality and reliability should not be adversely impacted by the inclusion of cyber security measures into the systems. This point should be assessed carefully, once cyber security measures are included.

The following sections list cyber security activities recommended for the SD and CD/ES stages. The cyber security activities are devised from those in RG 5.71 [9], NIST 800-30 [15], and NIST 800-82 [6].

There can be variations of the scheme of stages in the I&C system development process. In the case of variation, a slight modification to the sets of recommended activities may be applicable.

#### 3.2.2.1 Cyber security activities in the SD stage

Cyber security activities to be performed by system designers during the SD stage may include ;

- 1) Establishment of a cyber security program,
- 2) Analysis of the target operational environment,
- 3) Analysis of assets of the STD,
  - CDAs
  - Networks
  - Data flow
  - Connectivities
- 4) Design of baseline security controls to CDAs (Appendix B & C to RG 5.71),
- 5) Threat, vulnerability, and risk analyses,
- 6) Application of supplemental security measures to mitigate the vulnerabilities identified in 5),
- 7) Analysis of effects of security measures on functionality and reliability of the system, and
- 8) Iteration of 3), 5), 6), and 7), as needed.

### 3.2.2.2 Cyber security activities in the CD/ES stage

Cyber security activities during the CD/ES stage may include ;

- 1) Establishment of a cyber security program,
- 2) Maintaining a secure development environment,
- 3) Analysis of assets (with component design results including the 3rd party products and COTS items involved in the system development),

- 4) Threat, vulnerability, and risk analyses,
- 5) Application of supplemental security measures to mitigate the vulnerabilities identified in 4)
- 6) Analysis of effects of security measures on functionality and reliability of the system,
- 7) Vulnerability scanning and security testing, and
- 8) Iteration of 3), 4), 5), 6), and 7), as needed.

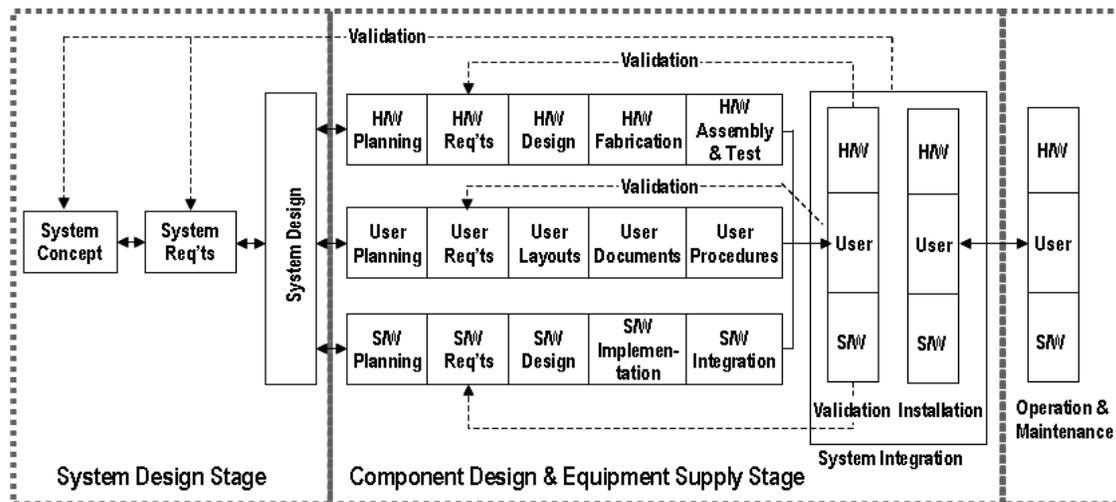


Fig. 3 General lifecycle process of I&C systems in NPPs (redrawn from NUREG-0800 Ch. 7.0 [14])

## 4 Conclusions

Cyber security becomes an important feature in the development of I&C systems in NPPs. This paper explores how to develop the I&C systems having appropriate cyber security features in a secure manner.

RG 1.152 revision 3 requires a secure development and operational environment for the safety systems and RG 5.71 requires the protection of digital systems from cyber attacks. The interpretation of these regulatory guides leads us to draw a conclusion on the policies in the development of I&C systems for NPPs in two points. First, the developers should maintain a secure development environment during their development of the systems based on their analysis of the development environment and the system itself within the development environment. Secondly, the system should be developed to have the security features necessary for a secure operation within the operation environment of NPPs in accordance with a secure development process. Cyber security activities in the SD and CD/ES stages are recommended for the developers.

## 5 References

[1] Control Systems Cyber Security: Defense in Depth Strategies, INL/EXT-06-11478, David Kuipers, Mark Fabro, Idaho National Laboratory, Idaho Falls, Idaho, May 2006.

<http://www.inl.gov/technicalpublications/Documents/3375141.pdf>

[2] Recommended Practice: Improving Industrial Control Systems Cyber security with Defense-In-Depth Strategies, Homeland Security, October 2009.

[http://www.us-cert.gov/control\\_systems/practices/documents/Defense\\_in\\_Depth\\_Oct09.pdf](http://www.us-cert.gov/control_systems/practices/documents/Defense_in_Depth_Oct09.pdf)

[3] Critical Infrastructure Protection, Challenges and Efforts to Secure Control Systems, GAO-04-354, United States General Accounting Office, March 2004.

<http://www.gao.gov/new.items/d04354.pdf>

[4] Top 10 Vulnerabilities of Control Systems and Their Associated Mitigations - 2007, North American Electric Reliability Council, December 7, 2006.

[http://www.us-cert.gov/control\\_systems/pdf/2007\\_Top\\_10\\_Formatted\\_12-07-06.pdf](http://www.us-cert.gov/control_systems/pdf/2007_Top_10_Formatted_12-07-06.pdf)

[5] Common Cyber Security Vulnerabilities Observed in DHS Industrial Control Systems Assessments, Homeland Security, July 2009.

[http://www.us-cert.gov/control\\_systems/pdf/DHS\\_Common\\_Vulnerabilities\\_R1\\_08-14750\\_Final\\_7-1-09.pdf](http://www.us-cert.gov/control_systems/pdf/DHS_Common_Vulnerabilities_R1_08-14750_Final_7-1-09.pdf)

[6] NIST Special Publication 800-82, Guide to Industrial Control Systems (ICS) Security, National Institute of Standards and Technology (NIST), September 2008.

[http://csrc.nist.gov/publications/drafts/800-82/draft\\_sp800-82-fpd.pdf](http://csrc.nist.gov/publications/drafts/800-82/draft_sp800-82-fpd.pdf)

[7] Regulatory Guide 1.152 revision 2, Criteria for Use of Computers in Safety Systems of Nuclear Power Plants, U.S. Nuclear Regulatory Commission, January 2006.

[8] 10 CFR Part 73.54, Protection of Digital Computer and Communication Systems and Networks, U.S. Nuclear Regulatory Commission, Washington, DC.

[9] Regulatory Guide 5.71, Cyber Security Programs for Nuclear Facilities, U.S. Nuclear Regulatory Commission, January 2010.

[10] NIST Special Publication 800-53 Rev 3, "Recommended Security Controls for Federal Information Systems", Aug. 2009.

[11] Draft Regulatory Guide DG-1249, Criteria for Use of Computers in Safety Systems of Nuclear Power Plants, U.S. Nuclear Regulatory Commission, June 2010.

[12] Tim Mossman, Security of Digital Safety Systems, NPIC&HMIT 2010, Las Vegas, Nevada, November 7-11, 2010.

[13] IEEE Standard 7-4.3.2-2010, Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations, August 2, 2010.

[14] NRC Standard Review Plan NUREG-0800 Chapter 7. 0 Instrumentation and Controls – Overview of Review Process, Revision 6, May 2010.

[15] NIST Special Publication 800-30, Risk Management Guide for Information Technology Systems, National Institute of Standards and Technology (NIST), July 2002.

# Security in Cloud Computing

Kazi Zunnurhain<sup>1</sup>, and Susan V. Vrbsky<sup>2</sup>

Department of Computer Science

The University of Alabama

[kzunnurhain@crimson.ua.edu](mailto:kzunnurhain@crimson.ua.edu); [vrbsky@cs.ua.edu](mailto:vrbsky@cs.ua.edu)

**Abstract** - *Cloud computing has been envisioned as the next generation architecture of IT Enterprises. It offers great potential to improve productivity and reduce costs. In contrast to traditional solutions, where the IT services are under proper physical, logical and personnel controls, cloud computing moves the application software and databases to large data centers, where the management of the data and services may not be fully trustworthy. This unique attribute, however, poses many new security challenges which have not been well understood yet. In this paper we investigate some prime security attacks on clouds: Wrapping attacks, Malware-Injection attacks and Flooding attacks, and the accountability needed due to these attacks. The focus of this paper is to identify and describe these prime attacks with the goal of providing theoretical solutions for individual problems and to integrate these solutions.*

**Keywords:** Cloud Security, Wrapping Attack, Flooding Attack, Hypervisor, Accountable cloud.

## 1 Introduction

In the field of computation, there have been many approaches for enhancing the parallelism and distribution of resources for the advancement and acceleration of data utilization. Data clusters, distributed database management systems, data grids, and many other mechanisms have been introduced. Cloud computing is currently emerging as a mechanism for high level computation, as well as serving as a storage system for resources. Clouds allow users to pay for whatever resources they use, allowing users to increase or decrease the amount of resources requested as needed. Cloud servers can be used to motivate the initiation of a business and ease its financial burden in terms of Capital Expenditure and Operational Expenditure [10].

Cloud computing has been introduced as providing a large framework that is beneficial for clients which utilize all or some aspects of it. Cloud computing can be thought of as composed of different layers, depending on the distribution of the resources. In this view, the CPU, memory and other hardware components reside at the bottom-most layer, called the Infrastructure as a Service (IaaS) layer. The layer which is responsible for hosting different environments for customer specific services is the middle layer, known as the Platform as the Service (PaaS) layer. Finally, the topmost layer is the Software as a Service (SaaS) layer, where cloud

service accessing takes place through the Web service and web browsers. Amazon EC2 is a well known example of IaaS, Google App engine is an example of PaaS and salesforce.com is an example of SaaS.

It is clear that cloud computing is the next step in the evolution of on-demand information technology services and products. The ancestors of cloud computing have existed for a long time now, such as the following distributed systems: AEC08, Con08, Fos04, Had08, IBM07c, Net06 and VCL04 etc. The term cloud computing became popular in October 2007 after the announcement of IBM's and Google's collaboration on Loh07 and IBM07a, which was followed by IBM's announcement of the "Blue Cloud" effort.

Cloud security is a complex issue, involving the different levels of the cloud, external and internal threats, and responsibilities that are divided between the user, the provider and even a third party. The focus of this paper is to identify and describe prime security attacks on clouds. The remainder of this paper is organized as follows. In Section II a short introduction about cloud security is provided followed by some specific cloud security issues and related work in Section III. Then Section IV focuses on the root causes of those security issues and approaches are presented in Section V to solve these problems. Finally the conclusions are presented with thoughts for our future work and improvements in Section VI.

## 2 Introduction to cloud security

Security threats on cloud users are both external and internal. Many of the external threats are similar to the threats that large data centers have already faced. This security concern responsibility is divided among the cloud users, the cloud vendors and the third party vendor involved in ensuring secure sensitive software or configurations.

If the application level security is the responsibility of the cloud user, then the provider is responsible for the physical security and also for enforcing external firewall policies. Security for intermediate layers of the software stack is shared between the user and the operator. The lower the level of abstraction exposed to the user, the more responsibility goes with it [18].

Besides the external security issues, the cloud does possess some internal security issues as well. Cloud providers must guard theft or denial-of-service attacks by users. In other words, users need to be protected from each other. Virtualization is the primary mechanism that today's clouds

have adapted because of its powerful defense and protection against most of the attempts by users to attack each other or the underlying cloud infrastructure. However, not all the resources are virtualized and not all virtualization environments are bug free. Virtualization software contains bugs that allow virtualized code to “break loose” to some extent. Incorrect network virtualization may allow user code access to sensitive portions of the provider’s infrastructure or to the resources of others.

The cloud should also be protected from the provider. By definition, the provider controls the bottom layer of the software stack, which effectively circumvents most known security techniques. The one important exception is the risk of inadvertent data loss.

In addition, if any kind of failure occurs, it is not clear who is the responsible party. A failure can occur for various reasons: 1) due to hardware, which is in the Infrastructure as a Service layer of the cloud; 2) due to malware in software, which is in the Software as a Service layer of the cloud; or 3) due to the customer’s application running some kind of malicious code, the malfunctioning of the customer’s applications or a third party invading a client’s application by injecting bogus data. Whatever the reason, a failure can result in a dispute between the provider and the clients.

## 2.1 Cloud security issues and related works

In this section we depict some prominent security issues in cloud computing today, along with the techniques applied by adversaries to intrude in the cloud. Also presented are the after effects when the intruder has made a successful compilation of his/her malicious program and hampered the regular functioning of the cloud system. We describe existing solutions and their pitfalls. Our observations in this paper will be specific to each issue rather than imposing security as a whole.

### 2.1.1 Soap Messages

As web service (WS) technology is the mostly used technology in the field of SOA in the Cloud, the WS-security system should be rigid enough to optimize the security attacks from different adversaries. Security attacks can involve SOAP messages. SOAP is an XML based messaging framework, used to exchange encoded information (e.g. web service request and response) over a variety of protocols (e.g. HTTP, SMTP, MIME). It allows a program running in one system to call a program running in another system and it is independent of any programming model[22].

As of now, two common attacks with SOAP messages are the Denial of Service and Wrapping attack. In the latter one, the wrapping element signature attack is the main picture for WS-security in large data centers like a cloud or grid system. So in that light, there are some IT companies who have accomplished some tasks thus far to prevent their system from such kinds of attacks. But even a company like Amazon had weaknesses in the SOAP request validation component

in their EC2 (Elastic Compute Cloud), and thus, allowed unprivileged actions to take place in the cloud on a victim’s account.

We now present an example using Amazon Web Service (AWS) [16] technology and its security. In the beginning, while registering, the customer has to provide a Self-Signed Certificate, and a randomly generated RSA to the AWS. If not, then a publicly defined certificate can be sent to the AWS with the signature. Here the AWS provides some command line tools to search the virtual machine images (AMI- Amazon Machine Images), to run those images, to monitor them and finally terminate some of the AMIs. These SOAP messages can be modified by the developers. The SOAP Header contains two elements. One is the BinarySecurityToken which contains the certificate mentioned above. The second is the TimeStamp which will contain the information of the creation and expiration of this SOAP. If the SOAP message is transferred through an unsecured layer, then the SOAP Body (inside the SOAP: Envelope) as well as the TimeStamp inside the SOAP Header needs to be signed. If the transport layer is secured then only the TimeStamp needs to be signed.

Since the channel is protected by means of SSL/TLS by default, this is largely an ineffective attack vector. Also, as the EC2 Web services allow access via simple HTTP as well, a passive attack would be sufficient to get in possession of such a request. For a wrapping attack to be successful, the only requirement here is that the bogus Body needs to have exactly the same ID as the original one.

### 2.1.2 Multi-core OS systems

Factored operating systems (*fos*) are designed to address the challenges found in systems, such as cloud computing and many core systems, and can provide a framework from which to consider cloud security. In reality there are several classes of systems having similarities to *fos*: traditional microkernels, distributed OS’s and cloud computing infrastructure. Traditional microkernels include Mach [2] and L4. Instead of simple exploitation of parallelism between servers, *fos* seeks to distribute and parallelize within a server for a single high level function [1]. The main motive of *fos* was to compel the scalability, elasticity of demand, fault tolerance and resolve difficulties in programming a large system. For a large system like a cloud, an OS such as *fos* is the perfect match to take care of all the above issues.

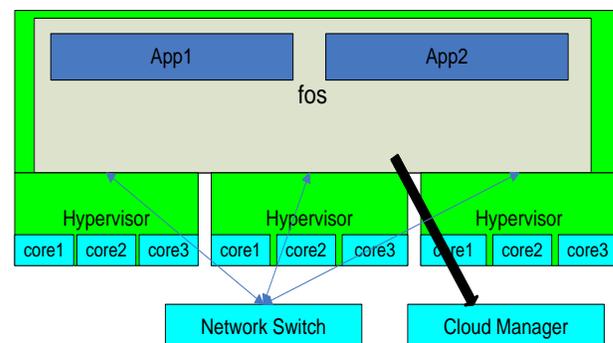


Figure 1: Hypervisor and Network Switch executes the scheduling where *fos* communicates with the Cloud Manager to send scheduling command

In many a core multi processor system, the OS manages and monitors the resources, and the scheduling task. So in the case of scalability, an application is factored into a service, then it is also factored in additional services to be distributed between service specific servers or a group of servers. Figure 1 illustrates the *fos* system functionality.

As mentioned previously, the resources are managed by the OS. So the cores are dynamically distributed and allocated for each of the services between the servers. A periodic message is monitored to verify if all the servers are working soundly or not. If one of the messages is missing, then a server fault is detected and a decision can be taken to appoint a new server for that specific task.

Unlike early cloud and cluster systems, *fos* provides a single system image to an application. This means that the application interface is that of a single machine while the OS implements this interface across several machines in the cloud. Aspects of *fos* can be used to secure cloud systems in and is discussed in Section V.

### 2.1.3 Securing Code, Control Flow and Image Repositories

Each user in the cloud is provided with an instance of a Virtual Machine (VM): an OS, application, etc. Virtual Machine Introspection (VMI) was proposed in [3] to monitor VMs together with Livewire, a prototype IDS that uses VMI to monitor VMs. A monitoring library named XenAccess is for a guest OS running on top of Xen that applies the VMI and virtual disk monitoring capabilities to access the memory state and disk activity of a target OS. These approaches require that the system must be clean when monitoring is started, which is a flaw and needs further investigation in VMI.

*Lares* [6] is a framework that can control an application running in an untrusted guest VM by inserting protected hooks into the execution flow of a process to be monitored. Since the guest OS needs to be modified on the fly to insert hooks, this technique may not be applicable in some customized OS.

All of these works have some flaws when security is considered in a cloud. So encapsulation of the cloud system in a secured environment is mandatory.

### 2.1.4 Accountability in clouds

Making the cloud accountable means that the cloud will be trustable, reliable and customers will be satisfied with their monthly or yearly charge for using the provider's cloud. In Section IV we discuss several types of attacks on clouds, all of which have an impact on the accountability of a cloud.

In this section we describe some of the work that has been done on accountability.

Trusted computing [19] is an approach to achieve some of the characteristics mentioned above to make a cloud accountable. Typically, it requires trusting the correctness of large and complex codebases.

A simple yet remarkably powerful tool of selfish and malicious participants in a distributed system is "equivocation": making conflicting statements to others. A small, trusted component is TrInc[20] which combats equivocation in large, distributed systems. TrInc provides a new primitive: unique, once-in-a-lifetime attestations. It is practical, versatile, and easily applicable to a wide range of distributed systems. Evaluation shows that TrInc eliminates most of the trusted storage needed to implement append-only memory and significantly reduces communication overhead in PeerReview. Small and simple primitives comparable to TrInc will be sufficient to make clouds more accountable.

## 2.2 Security issue causes

Next, we identify different kinds of attacks in a cloud: a) Wrapping attack, b) Malware-Injection attack, c) Flooding attack, and in the face of these attacks the need for Accountability checking. We describe each of these prime security issues in cloud systems and depict their root causes.

### 2.2.1 Wrapping attack problem

When a user makes a request from his VM through the browser, the request is first directed to the web server. In this server, a SOAP message is generated. This message contains the structural information that will be exchanged between the browser and server during the message passing.

Here the Body of the message contains the operation information and, it is supposedly signed by a legitimate user by appending the <KeyInfo> and reference signatures in the SOAP Body as well as the SOAP security <Header>. The SOAP header contains the SOAP Body. Figure 2 [10] shows an ideal case of a SOAP message where the user is requesting a file *me.jpg*.

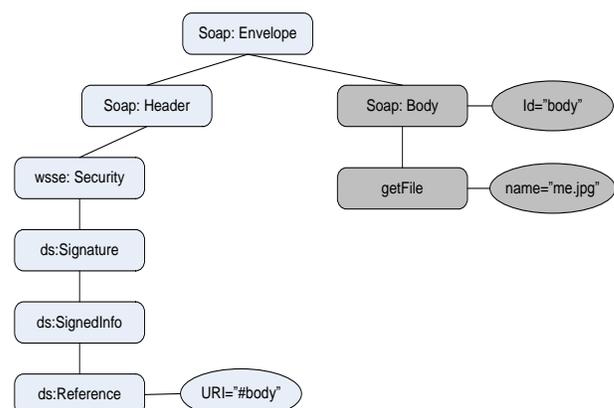


Figure 2: SOAP message before attack [10]

For a wrapping attack, the adversary does its deception before the translation of the SOAP message in the TLS (Transport Layer Service) layer. If the Body is included with a new Wrapper element inside the SOAP Header, then a simple validation can easily disclose the original SOAP message. Using this privilege an adversary makes a duplication of the message, as in Figure 3 [10], and sends it to the server as a legitimate user. The basic function for the attacker is to wrap the total message in a new header, the <Wrapper> element. Then the wrapper contains the original message body, which is the legitimate request from the user, and makes the <Security> as the new header element for that message. So when the validation session takes place, the server checks the authentication by the ID and integrity checking for the message. The Bogus elements and its contents are ignored by the recipient since this header is unknown, but the signature is still acceptable because the element at reference URI matches the same value as in the wrapper element.

There are other ways to detect a security breach through Wrapping. As discovered by Schaad and Rits, the *inline approach* [9] is one of them. There are some protected properties:

- Number of child elements of SOAP: Envelope
- Number of header elements inside Header
- Successor and Predecessor of each signed object

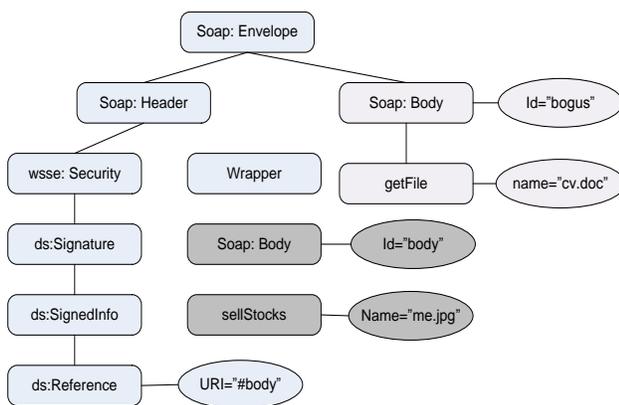


Figure 3: SOAP message after attack [5]

If an attacker changes the structure of the message and one of these properties is modified, the attack can be detected. This approach is known as schema validation. In this approach, the WS-Policy standardization will be adapted in the SOAP validation and the properties mentioned above will be injected as the SOAP header.

Since cloud computing is a new area in the field of SOA, it is anticipated these approaches to verify the SOAP message will experience many more obstacles.

## 2.2.2 Malware-injection attack problem

In the cloud system, as the client's request is executed based on authentication and authorization, there is a huge possibility of meta data exchange between the web server and web browser. An attacker can take advantage during this exchange of metadata. Either the adversary makes his own instance or the adversary may try to intrude with malicious code. In this case, either the injected malicious service or code appears as one of the valid instance services running in the cloud. If the attacker is successful, then the cloud service will suffer from eavesdropping and deadlocks, which forces a legitimate user to wait until the completion of a job which was not generated by the user. This type of attack is also known as a meta-data spoofing attack.

## 2.2.3 Flooding attack problem

In a cloud system, all the computational servers work in a service specific manner, with internal communication between them. Whenever a server is overloaded or has reached the threshold limit, it transfers some of its jobs to a nearest and similar service-specific server to offload itself. This sharing approach makes the cloud more efficient and faster executing requests.

When an adversary has achieved the authorization to make a request to the cloud, then he/she can easily create bogus data and pose these requests to the cloud server. When processing these requests, the server first checks the authenticity of the requested jobs. Non-legitimate requests must be checked to determine their authenticity, but checking consumes CPU utilization, memory and engages the IaaS to a great extent, and as a result the server will offload its services to another server. Again, the same thing will occur and the adversary is successful in engaging the whole cloud system just by interrupting the usual processing of one server, in essence flooding the system.

## 2.2.4 Accountability check problem

The payment method in a cloud System is "No use No bill". When a customer launches an instance, the duration of the instance, the amount of data transfer in the network and the number of CPU cycles per user are all recorded. Based on this recorded information, the customer is charged. So, when an attacker has engaged the cloud with a malicious service or runs malicious code, which consumes a lot of computational power and storage from the cloud server, then the legitimate account holder is charged for this kind of computation. As a result, a dispute arises and the provider's business reputation is hampered

## 2.3 Possible security approaches

In this section we discuss possible solutions for the three mostly probable attacks: wrapping attacks, malware-injection attacks and flooding attacks, as well as an accountability check for the Cloud system.

### 2.3.1 Wrapping attack solution

In this regard some additional precautions should be considered for the reliability of the SOAP message. Two approaches can be adapted by the registered users in this message passing:

- A Self signed Certificate and RSA key can be generated for convenience.
- Registering a public certificate with the provider.

These certificates will be authenticated by a trusted CA.

We propose that the Security Header must be signed while passing this message through an unsecured transport layer. When it is received in the destination, the validation is checked first. If the Timestamp (discussed in Section III.A) is not reasonable, then it can be assumed that security has been breached, actions can be taken accordingly and the SOAP message can be ignored.

### 2.3.2 Malware-injection attack solution

The client's VM is created and stored in the image repository system of the cloud. These applications are always considered with high integrity. We propose to consider the integrity in the hardware level, because it will be very difficult for an attacker to intrude in the IaaS level. Our proposal is to utilize a FAT-like (File Allocation Table) system architecture due to its straightforward technique which is supported by virtually all existing operating systems. From this FAT-like table we can find the application that a customer is running. A Hypervisor can be deployed in the provider's end. The Hypervisor is responsible for scheduling all the instances, but before scheduling it will check the integrity of the instance from the FAT-like table of the customer's VM.

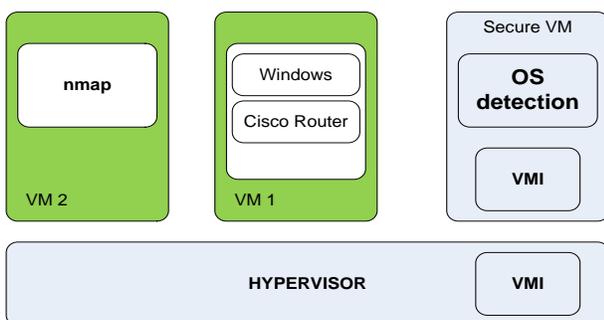


Figure 4: Guest OS Identification [4]

Now the question is how the FAT-like table will be utilized to do the integrity checking. The IDT (Interrupt Descriptor Table) can be used in the primary stage to detect. Firstly, the IDT location can be found from the CPU registers; then an analysis of the IDT contents and the hash values of in-memory code blocks can determine the running

OS in the VM. Finally, using the information of the running OS with the appropriate algorithms, all the running instances can be identified and then validated by the Hypervisor. So in Figure 4 (which is based on [4]), it is observed that the OS of the VM2 can be easily detected.

### 2.3.3 Flooding attack solution

For preventing a flooding attack, our proposed approach is to consider all the servers in the cloud system as a fleet of servers. Each fleet of servers will be designated for a specific type of job, like one fleet engaged for file system type requests, another for memory management and another for core computation related jobs, etc. In this approach, all the servers in the fleet will have internal communication among themselves through message passing, as in Figure 5. So when a server is overloaded, a new server will be deployed in the fleet and the name server, which has the complete records of the current states of the servers, will update the destination for the requests with the newly included server.

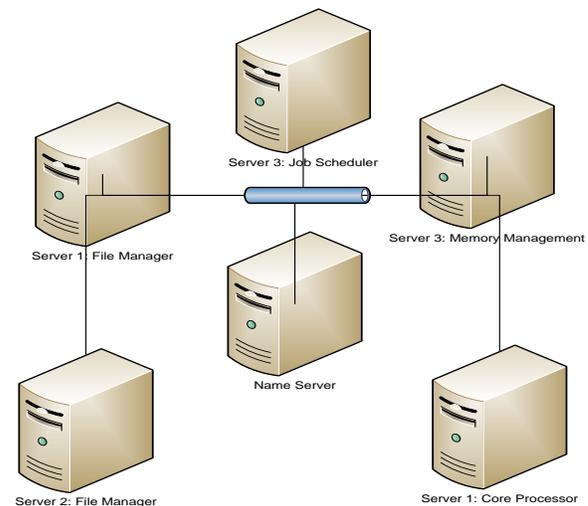


Figure 5: Messaging between servers

As mentioned in the previous section, a Hypervisor can also be utilized for the scheduling among these fleets. The Hypervisor will do the validity checking and if any unauthorized code is interrupting the usual computation in the cloud system, then the system will detect the instance by introspection. In this way, the flooding attack can be mitigated to an extent. If the Hypervisor is locally breached, which would require a misfeasor, then further analysis and efforts will be required to secure the Hypervisor.

Additionally, a PID can be appended in the messaging, which will justify the identity of the legitimate customer's request. The PID can be checked by the Hypervisor in the assignment of instances to the fleet of servers. This PID can be

encrypted with the help of various approaches, such as implementing hash values or by using the RSA.

### 2.3.4 Accountability check solution

The provider does not know the details of the customer's applications and it does not have the privilege to test the integrity of the application running in the cloud. On the other hand, customers do not know the infrastructure of the provider's cloud. If a customer is charged due to a malware attack or a failure, then the customer has no option to defend himself.

There can be unusual phenomenon, such as a dramatic increase in a current account usage balance all of a sudden or charges for instances at a specific time when the customer was away from the cloud. In this case, an investigation should take place before charging the customer, because an adversary may be responsible for these unusual activities. In our approach the following features will be ensured in the provider's end before launching any instance of a customer:

- Identities
- Secure Records
- Auditing
- Evidence

Firstly, before starting the instance, the identity of the legitimate customer should be checked by the Hypervisor. Secondly, all the message passing and data transfer in the network will be stored securely and uninterrupted in that specific node. Hence, when the auditing takes place, all the necessary information can be retrieved. Also, the evidence must be strong enough to clarify the recorded events, so the AUDIT will have the following properties: completeness, accuracy and verifiability. These properties ensure that when there is a security attack it is reported immediately, no false alarm will be reported and the evidence can be scrutinized by a trusted third party who will commit the task of AUDIT from a neutral point of view.

In some cases, there can be a conflict between privacy and accountability, since the latter produces a detailed record of the machines' actions that can be inspected by a third party. An accountable cloud can maintain separate logs for each of its customers and make it visible to only the customer who owns it. Also, the log available to customers will not have any confidential information about the infrastructure of the provider from which the IaaS can be inferred by the AUDITOR.

## 3 Conclusions

Cloud computing is revolutionizing how information technology resources and services are used and managed, but this revolution comes with new problems. We have depicted some crucial and well known security attacks and have

proposed some potential solutions in this paper, such as utilizing the FAT-like table and a Hypervisor.

In the future, we will extend our research by providing implementations and producing results to justify our concepts of security for cloud computing. The concepts we have discussed here will help to build a strong architecture for security in the field of cloud computation. This kind of structured security will also be able to improve customer satisfaction to a great extent and will attract more investors in this cloud computation concept for industrial as well as future research farms. Lastly, we propose to build strong theoretical concepts for security in order to build a more generalized architecture to prevent different kinds of attacks.

## 4 References

- [1] D. Wentzlaff, C. Gruenwald III, N. Beckmann, K. Modzelewski, A. Belay, L. Touseff, J. Miller, and A. Agarwal. Fos: A Unified Operating System for Clouds and Manycore. *Computer Science and Artificial Intelligence Laboratory TR*, Nov. 20, 2009.
- [2] M. Accetta, R. Baron, W. Bolosky, D. Golub, R. Rashid, A. Tevanian, and M. Young. Mach: A new kernel foundation for UNIX development. In *Proc. of the USENIX Summer Conference*, pp. 93-113, June 1986.
- [3] T. Garfinkel, M. Rosenblum, A virtual machine introspection based architecture for intrusion detection. *Proc. 2003 Network and Distributed Systems Symposium*, 2003.
- [4] M. Christodorescu, R. Sailer, D. L. Schales, D. Sgandurra, D. Zamboni. *Cloud Security is not (just) Virtualization Security*, CCSW'09, Nov. 13, 2009, Chicago, Illinois, USA.
- [5] L. Litty and D. Lie. Manitou: a layer-below approach to fighting malware. In *ASID '06: Proc. of the 1<sup>st</sup> workshop on Architectural and system support for improving gsoftware dependability*, pages 6-11, New York, NY, USA, 2006. ACM.
- [6] B.D. Payne, M. Carbone, M. Sharif, and W. Lee. Lares: An architecture for secure active monitoring using virtualization. *Security and Privacy, IEEE Symposium on*, 0:233-247, 2008.
- [7] VMware. Virtual Appliance Marketplace. <http://www.vmware.com/appliances/>.
- [8] Amazon Elastic Compute Cloud (Amazon EC2). <http://aws.amazon.com/ec2>.
- [9] M. A. Rahaman, A. Schaad, and M. Rits. Towards secure SOAP message exchange in a SOA. In *SWS '06: Proceedings of the 3rd ACM workshop on Secure Web Services*, pages 77-84, New York, NY, USA, 2006. ACM Press.
- [10] Meiko Jenson, Jorg Schwenk, Nils Gruschka, Luigi Lo Iacono. On Technical Security Issues in Cloud Computing. *IEEE International Conference on Cloud Computing 2009*.
- [11] D. Kormann and A. Rubin, "Risks of the passport single signon protocol," *Computer Networks*, vol. 33, no. 1-6, pp. 51-58, 2000.
- [12] M. Slemko, "Microsoft passport to trouble," 2001, <http://alive.znep.com/~marcs/passport/>.
- [13] Andreas Haeberlen. A Case for Accountable Cloud. *Max Planck Institute of Software System (MPI-SWS)*.
- [14] S. Gajek, J. Schwenk, M. Steiner, and C. Xuan, "Risks of the cardspace protocol," in *ISC'09: Proceedings of the 12th Information Security Conference, LNCS*. Springer, 2009.
- [15] X. Chen, S. Gajek, and J. Schwenk, "On the Insecurity of Microsoft's Identity Metasystem CardSpace," *Horst G'ortz Institute for IT-Security, Tech. Rep. 3*, 2008.

- [16] Nils Gruschka and Luigi Lo Iacono. Vulnerable Cloud: SOAP Message Security Validation Revisited. *NEC Laboratories Europe Rathausallee 10 D-53757 Sankt Augustin (Germany), 2009 IEEE.*
- [17] Mladen A. Vouk. Cloud Computing- Issues, Research and Implementations. *Proceedings of the ITI 2008 30<sup>th</sup> Int. Conf. on Information Technology Interfaces, June 23-26,2008, Cavtat, Croatia.*
- [18] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, Daviv Patterson, Ariel Rabkin, Ion Stoica and Matei Zaharia. A View of Cloud Computing. *Communications of the ACM, April 2010.*
- [19] Nuno Santos, Krishna P. Gummadi, and Rodrigo Rodrigues. Towards trusted cloud computing. In *Proc. HotCloud*, June 2009.
- [20] Dave Levin, John R. Douceur, Jacob R. Lorch and Thomas Moscibroda. TrInc: Small trusted hardware for large distributed system. In *Proc. NSDI*, April 2007.
- [21] Jinpeng Wei, Xiaolan Zhang, Glenn Ammons, vasanth Bala, Peng Ning. Managing Security of Virtual Machine Images in a Cloud Environment. *CCSW'09*, November 13,2009, Chicago,Illinois, USA.
- [22] SOAP, <http://www.w3.org/TR/soap/>

# Security-Oriented Formal Techniques

Marcantoni F. , Paoloni F. and Polzonetti A.

School of Science and Technology – University of Camerino ITALY

**Abstract** - *Please consider these Instructions as guidelines for preparation of Final Camera-ready Papers. The Camera-Ready Papers would be acceptable as long as it is formatted reasonably close to the format being suggested here. Note that these instructions are reasonably comparable to the standard IEEE typesetting format. Type the abstract (100 words minimum and 150 words maximum) using Italic font with point size 10. The abstract is an essential part of the paper. Use short, direct, and complete sentences. It should be brief and as concise as possible.*

**Keywords:** Security, Formal Methods

## 1 Introduction

Security of software systems is a critical issue in a world where Information Technology is becoming more and more pervasive. The number of services for everyday life that are provided via electronic networks is rapidly increasing, as witnessed by the longer and longer list of words with the prefix "e", such as e-banking, e-commerce, e-government, where the "e" substantiates their electronic nature. These kinds of services usually require the exchange of sensible data and the sharing of computational resources, thus needing strong security requirements because of the relevance of the exchanged information and the very distributed and untrusted environment, the Internet, in which they operate. It is important, for example, to ensure the authenticity and the secrecy of the exchanged messages, to establish the identity of the involved entities, and to have guarantees that the different system components correctly interact, without violating the required global properties.

Unfortunately, many authoritative security-related organizations as, e.g., the CERT at Carnegie Mellon University, report a growing number of computer system vulnerabilities which are often the result of exploits against defects in the design or code of software. The approach most commonly employed to address such defects is to attempt to a posteriori "repair the flaw" by making it more difficult for those defects to be exploited. This solution, however, does not certainly get to the root cause of the problem and threat. A complementary approach is, instead, to model and verify security requirements from the very first specification of software systems, so to reduce as much as possible the presence of vulnerabilities on the final product. The use of formal techniques can thus play an important role to reveal possible security flaws from the very first phases of software development, to understand in depth the causes, and to remove them before it is too late and it becomes necessary to invent, if possible, some retroactive remedy. The interest in

formal methods for security is confirmed by a very active international community, and by the increasing number of new international workshops and conferences on the topic.

The aim of this project is to put together a consortium of 3 Universities which are already active in the fields of formal methods for security and of software and protocol verification, and which will focus their effort on common research targets. We intend to broadly work on many different aspects of security, mainly focusing on "language-based" techniques, which have the advantage of verifying security of programs directly on their formal specification, without the need of analysing their execution. We believe that this approach is particularly appealing both because it can often be automated through efficient verification algorithms and because it gives the programmer a clear comprehension of security requirements and mechanisms. We will consider both high (i.e., application) level properties as, e.g., information flow and "Service-Oriented" security, and low (i.e., communication) level properties as, e.g., authentication, secrecy and non-repudiation on standard and ad-hoc networks. We will also study how results achieved on "standard" symbolic models scale to computational and causal models, the former providing a more concrete representation of cryptography and the latter expressing security properties in terms of explicit cause-and-effect relations.

As illustrated in more detail in the following sections), our job is of a foundational nature, since it focuses on the definition and development of formal methodologies for the analysis of various aspects of information security.

## 2 National and International background

This job will focus on diverse research topics related to the application of formal methods to security, that we shortly describe below.

### 2.1 Communication and Network Security

Cryptographic protocols are one of the fundamental mechanisms for achieving security on computer networks. Wide-area networks are, in fact, not controllable and there is a need to protect sent/received data through cryptographic techniques. Even if these protocols are often just a few lines of codes, many attacks subverting the protocol logic and invalidating the expected security properties have been found. These attacks are not necessarily based on cryptographic flaws and can be reproduced even when cryptography is considered as a fully reliable black box. In the literature we find a huge amount of contributions on the analysis and verification of security protocols, but only a few of them

follow a language-based approach, i.e., are based on static-analysis. We mention here some relevant papers on secrecy [A99,AB05] and authentication [BBDNN05, BFM07, GJ03, GJ04]. We intend to go on on this line of research by focussing on abstract interpretation and control flow analysis of cryptographic protocols and abstract communication primitives to make programming independent of cryptographic implementation. Moreover, we also aim to extend the symbolic protocol analysis approach [AVISPA,Bla01,RSGLR00] in order to allow for the specification and verification of a larger class of protocols and properties than currently possible, as well as of different attackers models, extending preliminary work such as [HDMV05,HDMVB06].

We will finally consider security on Ad-hoc networks. A Mobile Ad-Hoc Network is an autonomous system composed of devices communicating with each other via radio transceivers. Mobile devices are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Trust establishment in the context of ad-hoc networks is still an open and challenging field [Gli04,PM06], because of lack of a fixed networking infrastructure, high mobility of the devices, limited range of the transmission, shared wireless medium, and physical vulnerability. We would like to develop formal models of trust that fit the constraints of ad-hoc networking, integrating these models in a process calculus for ad-hoc networks [NH06,Mer07,God07], thus developing an appropriate theory to formally prove security properties.

## 2.2 Application Security

Controlling information flow in programs and systems is a fundamental security issue whose theoretical foundations have been extensively studied. The aim is to control secret information so that it cannot flow towards unprivileged users who do not have the clearance to access it. Non-Interference is one of the reference properties for achieving this kind of control, and it was introduced by Goguen and Meseguer in [GM82]. The main idea is to require that any possible modifications of high level data have no observable effects at lower levels or, in other words, do not interfere with lower views of the system. In literature, we find many variants and extensions of Non-Interference on process calculi and simple imperative languages; see, e.g., [BCF02, BCFLP04, FG95, FRS05, GM04, MSZ06, RWW96, RS01, SS00, SM03, SV98]. Our research will specifically focus on Information Flow Security of distributed programs with cryptography, secure refinement of programs, security of (a multi-threaded fragment of) Java and extending the abstract non-interference framework [GM04] in order to deal with more powerful attackers.

Security also plays a crucial role in Service Oriented Computing. In this scenario, applications are built by assembling stand-alone components distributed over a

network, called services. Services are open, i.e., built with little or no knowledge about their operating environment, their clients, and further services. Therefore, their secure composition and coordination may require peculiar mechanisms. Web Services [S02] built upon XML technologies are probably the most illustrative and well developed example of this paradigm. We intend to extend the results of [BDF06a, BDF06b], where we propose an approach based on semantic descriptions and a methodology which automates the process of discovering services and planning their composition in a secure way. Moreover, we plan to scale up the techniques developed for protocol analysis to security services. There are a number of preliminary approaches in this direction [BMPV06,SAMOA], but none of them has yet reached the required maturity.

## 2.3 Quantitative Aspects of Security

There are recent papers studying how formal analysis scales to computational security, a model of security requiring resistance over all the possible probabilistic polynomial-time attacks. This model, differently from Dolev-Yao, does not consider cryptography as a secure black box (see, e.g., [AR00, BCK05, BPW03, L05]). A formal symbolic analysis, à la Dolev-Yao, is typically simpler and easier to automate with respect to computational models. It is thus appealing to understand how symbolic formal results scale to these models and under which cryptographic assumptions this may happen. Even in this setting, the language-based approach has not been extensively studied. An interesting paper in this direction is [L05], which proposes a type system for message secrecy. It exploits a semantics based on the "simulatable cryptographic library" [BPW03] to scale the results to computational models. We intend to develop a static analysis based on [BFM07] for the verification of authentication protocols using the "simulatable cryptographic library".

We also intend to explore hybrid models that combine the two approaches: symbolic and computational. There is already a related literature [PW01, CCKLLPS06, MRST06,CP07] that in particular highlights a fundamental role of nondeterminism, for which an arbitrary resolution may lead to undesired conclusions. Thus, the main open problems are a correct management of nondeterminism and the study of hierarchical techniques that take care of computational aspects as well. The recent case study [ST07] analyses a simple and well known authentication protocol using Probabilistic Automata and a new notion of computationally bounded approximated simulation that allows an abstract system to emulate computational steps of a concrete system up to some negligible error. This case study constitutes a significant starting point for developing hierarchical and compositional proof methods for security..

## 2.4 Causal models for security

In the literature on cryptographic protocols analysis, we find some recent approaches in which the causal dependencies among events play a very important role [BCM07,CW01,FHG98,P99]. Strand spaces [FHG98] are a well known method in which causal dependencies are made explicit. In the inductive method of [P99], dependencies are instead a consequence of inductive rules. Proved Transition systems [DP92,DP99] represent an extension of transition systems towards causality. Proved Transition Systems can be considered as a sort of compact representation of computations, containing all the possible encodable and relevant information. Transitions are enriched with labels encoding their proofs, i.e. the steps involved in the deduction process of the action just executed. By inspecting the transition labels, it is possible to infer the causal dependencies, represented through a set of references to previous transitions. Starting from the enhanced semantics of [BetAl05], we intend to investigate the possible application of causal semantics based on Proved Transition Systems and on true-concurrent models, to the analysis of cryptographic protocols.

The Distributed State Temporal Logic (DSTL) [MSS04] permits to causally relate properties, which might hold in distinguished components of a system, in an asynchronous setting. The logic includes a primitive operator to specify events, thus allowing us to mix conditions and events in the specification formulae. The ability to deal with events explicitly enhances the expressiveness and simplicity of logical specifications, and seems especially adequate in the case of security properties specification. Starting from [MS04], we intend to further investigate the use of DSTL for the specification and verification of applications in which components presents various security requirements.

## 3 Results and Suggestions

Information security is becoming more and more relevant given the increasing usage of computers and networks for critical applications as, e.g. e-commerce, home-banking, purchase of digital goods and, in general, on-line services. It becomes thus very relevant to understand in depth the security requirements of distributed applications and to investigate methods for the automated verification of such requirements. The primary aim of the job is the study of foundations of information security and the development of formal methods for the specification and verification of security properties of programs, systems and computer networks.

We intended to cover many different aspects of security working both on high (i.e., application) level properties as, e.g., information flow and "Service-Oriented" security, and on low (i.e., communication) level properties as, e.g., authentication, secrecy and non-repudiation on standard and

ad-hoc networks. Regarding formal methods, we mainly intended to investigate "language-based" techniques, which have the advantage of verifying security of programs directly on the code, without the need of analysing their execution. We believe that this approach was particularly appealing both because it could often be automated through efficient verification algorithms and because it gave the programmer a clear comprehension of security requirements and mechanisms.

We divided the work in four that reflect the logical and temporal scheduling of activities, corresponding to a "standard agenda" of the development of formal methods for security:

Step 1 - Security oriented languages and models. We studied security oriented languages, i.e., languages specifically developed for the specification and verification of security properties.

Step 2 - Security properties. We studied and formalized security properties on the languages defined in the previous step

Step 3 - Analysis techniques. We studied analysis techniques for the properties and languages described above. We implemented and extended verification tools based on the above mentioned techniques

### 3.1 Communication and Network Security.

We are interested in the analysis of cryptographic protocols through abstract interpretation, type systems, control flow analysis and causal semantics . We planned to extend the study of cryptographic protocols to distributed applications based on cryptography, by integrating this study with the program analysis techniques. We proposed new security-oriented languages and process calculi for distributed systems. We developed a logic for expressing local and global properties of distributed systems. Finally, we studied security models for ad-hoc networks.

### 3.2 Application Security.

We studied different aspects of program security through abstract interpretation: in particular, we are interested in models and methods for verifying non-interference in presence of active attackers and in probabilistic computations; we have dealt with confidentiality and, in particular, both with "termination covert channels" in which the attacker gets information by observing the program termination, and with "timing covert channels". We studied properties for the secure refinement of programs in order to achieve a step-by-step development of secure applications, starting from abstract specifications. Finally, we studied primitives for the secure composition of clients and services in the setting of "Service-Oriented Computing".

### 3.3 Quantitative Aspects of Security.

We intended to study how properties described above, scale on finer-grained models, in which time and probabilities are explicitly modeled. We studied techniques to detect and remove timing attacks, by transforming a program so that its timing behavior is corrected while the input/output behavior is preserved. We also intended to develop new analysis techniques for computational security of cryptographic protocols. On the one hand, we developed type-based techniques for the correctness of protocols expressed on the cryptographic library proposed by Backes-Pfitzmann-Waidner; on the other hand, we studied soundness results of the symbolic model with respect to the computational model, through the work on approximated simulation relations of Segala and Turrini.

### 3.4 Causal models for security.

In the formalization of security properties it might be beneficial to reason in terms of causality among events. For example, in entity authentication we have that authentication should always be caused by the actual execution of the protocol by the claimant. We intended to give a new causal semantics to cryptographic protocols which enables us to directly observe the causality between the protocol conclusion, i.e., the authentication, and the corresponding execution by the authenticated entity. In doing this, we investigated both true-concurrent models like, e.g., event structures, and models of causality based on proved transition systems.

## 4 Conclusions

For each part of the job we give a list of the main results. These results are intentionally very specific, so to be verifiable.

For the Communication and Network Security:

- new formal models of trust for ad-hoc networks and integration of these models into suitable process calculi.
- a new security-oriented temporal logic for communicating processes.
- definition of an abstract interpretation of challenge-response authentication protocols;
- definition of new Control Flow Analyses for security protocols;
- extension of the verification tool proposed in [BBDNN05] to the new Control Flow Analyses;

- use of symbolic techniques and refinement for the verification of security properties;
- extension of AVISPA to the logic described;
- investigation of possible extensions of AVISPA to other techniques developed.

For the Application Security :

- definition of security-oriented imperative languages with cryptographic communication primitives;
- new abstract communication primitives that make programming independent of cryptographic implementation.
- new general framework for secure stepwise refinement of programs;
- new dynamic type systems for the security of distributed applications with cryptography;
- extension of the call-by-property invocation mechanism of Service-Oriented Computing to other security properties and non-functional aspects;
- extension of existing orchestration techniques to scenarios in which services may be published on-the-fly and may become temporarily unavailable;
- extension of abstract Non-Interference in order to deal with active attackers able to exploit probabilistic techniques;
- application of abstract Non-Interference to data bases and data mining.

For the Quantitative Aspects of Security :

- a new calculus for cryptographic protocols, with both a symbolic and a computational semantics based on the simulatable cryptographic library by Backes, Pfitzmann and Waidner [BPW03];
- a new hybrid model for security protocols combining symbolic and computational aspects;
- an extension of the process calculus LySa which is able to deal with type misinterpretation attacks.
- type systems for cryptographic protocols with both a symbolic and a computational semantics.

For the Causal models for security :

- new causal semantics for existing calculi of cryptographic protocols.
- new causality-based formalizations of security properties;
- new formalizations of security properties using the logic DSTL.
- specializations of already studied techniques to the new semantics, with special attention to authentication protocols

## 5 References

- [A99] M. Abadi. Secrecy by Typing in Security Protocols. *Journal of the ACM*, 46(5):749–786, 1999.
- [AB05] M. Abadi and B. Blanchet. Analyzing Security Protocols with Secrecy Types and Logic Programs. *Journal of the ACM*, 52(1):102–146, 2005
- [AR00] M. Abadi and P. Rogaway. Reconciling Two Views of Cryptography (The Computational Soundness of Formal Encryption). In *proc. of IFIP TCS 2000 (LNCS 1872)* pp. 3–22.
- [AVISPA] The AVISPA Project. [www.avispa-project.org](http://www.avispa-project.org)
- [BBDNN05] C.Bodei, M.Buchholtz, P.Degano, F.Nielson, H.R.Nielson. Static Validation of Security Protocols. *JCS* 13(3), 2005.
- [BCF02] C.Braghin, A. Cortesi, and R. Focardi. Security Boundaries in Mobile Ambients. *Computer Languages*, 28(1):101-127, 2002
- [BCFLP04] C. Braghin, A. Cortesi, R. Focardi, F.L. Luccio, and C. Piazza. Nesting Analysis of Mobile Ambients. *Computer Languages, Systems & Structures* 30(3-4):207-230, 2004
- [BCK05] M. Baudet, V. Cortier and S. Kremer. Computationally Sound Implementations of Equational Theories against Passive Adversaries, In *Proc. of ICALP'05*. LNCS 3580. pp. 652-663.
- [BCM07] M. Backes, A. Cortesi, M. Maffei. Abstracting Multiplicity in Cryptographic Protocols. In *Proc. of IEEE CSF'07*, pp. 355-369
- [BDF06a] M.Bartoletti, P.Degano, G.L.Ferrari. Plans for service composition. *Proc. of WITS*, 2006.
- [BDF06b] M.Bartoletti, P.Degano, G.L.Ferrari. Types and effects for secure orchestration. *Proc. of IEEE CSFW*, 2006.
- [BetA15] C.Bodei, M.Buchholtz, P.Degano, M.Curti, C.Priami, F.Nielson, H.R.Nielson. On Evaluating the Performance of Security Protocols specified in Lysa. *Proc. of PACT05, LNCS 3606*. Rees Source Person. “Title of Research Paper”; name of journal (name of publisher of the journal), Vol. No., Issue No., Page numbers (eg.728—736), Month, and Year of publication (eg. Oct 2006).
- [BFM07] M. Bugliesi, R. Focardi and M. Maffei. Dynamic Types for Authentication, *Journal of Computer Security*, IOS Press, 15(6):563-617, 2007
- [Bla01] B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. *IEEE CSFW'01*
- [BMPV06] M. Backes, S. Moedersheim, B. Pfitzmann, L. Vigano`. Symbolic and Cryptographic Analysis of the Secure WS-Reliable Messaging Scenario. In *Proc. of FOSSACS'06*. LNCS 3921.
- [BPW03] M. Backes, B. Pfitzmann, and M. Waidner. A Universally Composable Cryptographic Library. In *proc. of ACM CCS 2003*, pp. 220-230.
- [CCKLLPS06] R. Canetti, L. Cheung, D. Kirli Kaynar, M. Liskov, N. A. Lynch, O. Pereira, R. Segala: Time-Bounded Task-PIOAs: A Framework for Analyzing Security Protocols. In *Proc. of DISC'06*. LNCS 4167, pp 238-253.
- [CP07] K. Chatzikokolakis, C. Palamidessi. Making Random Choices Invisible to the Scheduler. In *Proc. of CONCUR'07*. LNCS 4703, pp. 42-58.
- [CW01] F.Crazzolaro, G.Winskel. Events in Security Protocols. In *ACM CCS*, 2001.
- [DP92] P.Degano, C.Priami. Proved Trees. In *Proc. of ICALP'92*.
- [DP99] P.Degano, C.Priami. Non Interleaving Semantics for Mobile Processes. *TCS* 216, 1999.
- [FG95] R. Focardi and R. Gorrieri, A Classification of Security Properties for Process Algebras, *Journal of Computer Security*, 3(1):5-33, 1995
- [FHG98] F.J.T. Fábrega, J.C.Herzog, J.D.Guttman. Strand spaces: Why is a security protocol correct? *JCS* 7(2-3), 1999.
- [FRS05] R. Focardi, S. Rossi, A. Sabelfeld: Bridging Language-Based and Process Calculi Security. In *Proc. of FoSSaCS 2005* pp. 299-315. LNCS 3441
- [GJ04] A. D. Gordon and A. Jeffrey. Types and effects for asymmetric cryptographic protocols. *Journal of Computer Security*, 12(3-4):435–483, 2004

- [Gli04] V.D.Gligor. Security of Emergent Properties in Ad-Hoc Networks. Security Protocols Workshop 2004
- [GM04] R. Giacobazzi and I. Mastroeni. Abstract Non-Interference. POPL'04
- [God07] J.C. Godskesen. A Calculus for Mobile Ad Hoc Networks. COORDINATION'07
- [HDMV05] P. Hankes Drielsma, S. Moedersheim, L. Vigano`. A Formalization of Off-Line Guessing for Security Protocol Analysis. LPAR04
- [HDMVB06] P. Hankes Drielsma, S. Moedersheim, L. Vigano`, D. Basin. Formalizing and Analyzing Sender Invariance. FAST'06
- [L05] P. Laud. Secrecy types for a simulatable cryptographic library. In Proc. of the 12th ACM CCS '05. New York, NY, 26-35.
- [MRST06] J. C. Mitchell, A. Ramanathan, A. Scedrov, V. Teague. A probabilistic polynomial-time process calculus for the analysis of cryptographic protocols. TCS 353, 2006
- [MSZ06] A. C. Myers, A. Sabelfeld, and S. Zdancewic. Enforcing Robust Declassification. Journal of Computer Security, 14(2):157-196, 2006.
- [NH06] S.Nanz, C.Hankin. A Framework for Security Analysis of Mobile Wireless Networks. TCS 367, 2006
- [Mer07] M.Merro. On the Observational Theory of Mobile Ad-Hoc Networks. Information and Computation, to appear.
- [MSS04] C.Montangero, L. Semini and S. Semprini. Logic Based Coordination for Event-Driven Self-Healing Distributed Systems. Proc. of COORDINATION'04, LNCS 2949.
- [MS04] C.Montangero, L. Semini. Formalizing an Adaptive Security Infrastructure in Mobadtl. Proc. of FCS'04.
- [P99] L.C.Paulson. Proving security protocol correct. Proc. of Lics, 1999.
- [PM06] A.A. Pirzada, C.McDonald. Trust Establishment in Pure Ad-Hoc Networks. Wireless Personal Communication 379, 2006
- [PW01] B. Pfitzmann, M. Waidner. A Model for Asynchronous Reactive Systems and its Application to Secure Message Transmission. IEEE Symposium on S&P 2001
- [RS01] P. Ryan and S. Schneider, Process algebra and Non-Interference, Journal of Computer Security 9(1/2):75-103, 2001.
- [RSGLR00] P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, and B. Roscoe. Modelling and Analysis of Security Protocols. 2000
- [RWW96] A.W. Roscoe, J.C.P. Woodcock and L. Wulf, Non-Interference through determinism, Journal of Computer Security 4(1):27-54, 1996.
- [S02] M.Stal. Web services: Beyond component-based computing. Comms. Of the ACM 55(10), 2002.
- [SAMOA] Samoa: Formal Tools for Securing Web Services. <http://research.microsoft.com/projects/samoa/>.
- [SS00] A. Sabelfeld and D. Sands, Probabilistic Noninterference for multi-threaded programs, in: Proc. of IEEE CSFW 2000, pp.200-215.
- [SM03] A. Sabelfeld and A.C. Myers, Language-based information-flow security, IEEE Journal on Selected Areas in Communication 21(1):5-19, 2003.
- [ST07] R. Segala, A. Turrini. Approximated Computationally Bounded Simulation Relations for Probabilistic Automata. IEEE CSF07
- [SV98] G. Smith and D.M. Volpano, Secure information flow in a multi-threaded imperative language, in: Proc. of 25th ACM POPL, pp.355-364, 1998

# Internal Vs. External Penetrations: A Computer Security Dilemma

Pedro A. Diaz-Gomez, Gilberto ValleCarcamo, Douglas Jones

Computing & Technology Department, Cameron University, Lawton, OK, USA

**Abstract**—*In computer security it has been said that internal penetrations are the highest threat for data and information. This paper took the challenge to investigate if such a common belief is true. Various statistics are analyzed with the goal to give some light to the research community about internal and external penetrations. This paper highlights a weakness in computer security called “the unknown”, which corresponds to intrusions to computers and network resources from which organizations do not know the cause.*

**Keywords:** computer security, data breach, external penetration, internal penetration.

## 1. Introduction

It is a common belief that most of all penetrations to computer resources come from within the organization [1], [4], [12], [17], [22], [24], [20] and, it is normal to think that computer users, who have rights and access to particular resources in the system, constitute the principal threat. *Anderson* [2] indicates that the internal penetrator has no barriers to surpass in order to have access to the computer, and that their intrusion activity could be difficult to track. Three categories of users are identified: the masquerader, the legitimate, and the clandestine user. The masquerader is a user that steals credentials to have access to computers, pretending to be a trusted party. The legitimate user is the user that has been granted access to computer resources by an organization, and uses his or her own credentials to use them. The clandestine user has or can get superuser privileges. All these intrusions constitute a security threat to computer resources [2].

The barrier that classifies insider from outsider is difficult to draw. *Anderson* [2], for example, defines an outsider as the one that has no permission to use computer resources. In this sense, an outsider could as well be an employee who has no rights to use the computer, as well as a hacker, that can seize security mechanisms in order to have access to it. *Pfleeger*, on the other hand, gives various definitions of the term insider: as an employee or other member of an organization who has permission to use the system, as customers who perform transactions with an organization as part of services or businesses, as anyone identified and authenticated by the system—could be a masquerader, as someone that executes actions on the system on behalf of an outsider, and as a former employee that uses privileges not revoked or that uses privileges secretly created while at work [20]. The CERT cybersecurity survey defines an insider, as a current

or former employee, service provider or contractor; and outsider, as someone that has never been granted computer and network access privileges of an organization [9]. In this sense, a former employee with revoked privileges that is able to bypass security mechanisms will be considered an insider.

Some reports go against the common trend which says that insiders are the highest threat for computer resources. The *Data Breach Investigations Report* gives statistics which show the trend that outsiders are responsible for a higher number of intrusions and a higher number of records breached [25], [26].

A statement about what actors of computer penetrations are responsible for the majority and more devastating attacks could be difficult to demonstrate. Not only such statement could be biased by the observer—in particular if it is a vendor—the type of organizations that report or not report, but also for the data sample. Coming from inside, coming from outside or working in conjunction, computer penetrators are developing new techniques, like network sniffers and RAM scrapers, that allow them to perform sophisticated penetrations and avoid discovery [26]. Computer security countermeasures have been addressed to mitigate such threats, like intrusion detection systems, firewalls and anti-viruses; however, errors, misconfigurations and noncompliance with security policies have allowed some successful penetrations that could be avoided if those countermeasures are in place [26].

The order of this paper is as follows: Section 2 presents some basic definitions, Section 3 describes the penetration problem, Section 4 presents statistics about external vs. internal penetrations, Section 5 relates to the analysis of statistics presented, and Section 6 presents the conclusions and future work.

## 2. Basic Definitions

The following definitions are used in this paper:

- **Threat:** any potential danger to computers and network resources, like unauthorized access to confidential information, virus infection and system malfunction [3], [16]. There are external threats that originate from outside the organization, and internal threats that originate from within the organization [25].
- **Threat agent:** the actual penetrator or intruder that performs the threat, like outsiders, insiders, viruses and trojan horses [16].

- **Outsider:** an external threat agent, in other words, an agent from outside the organization [25], or an agent not authorized to use the system [3].
- **Insider:** internal threat agent, in other words, an agent that belong to the organization [25]. For example, this includes an authorized user that surpasses his or her legitimate access rights [3].
- **Penetration or intrusion:** all incidents involving the successful breach to computer software, computer systems or computer networks [2], [19]. There are internal penetrations and external penetrations depending if the penetrations were performed by an identified and authorized user, or by someone not identified, or not authorized to use the system [1].
- **Incident:** an event or set of events that affects an organization negatively. An incident can be observed, verified and documented [16], such as with a data breach.
- **Don't Know:** If organizations do not know whether there was any unauthorized use of their computer systems and networks [6].
- **Unknown:** All incidents involving an unknown cause [19].
- **None:** The organization reports no penetration.

### 3. The Penetration Problem

Anderson [2] studied the penetration problem from the prospective of whether an attacker is authorized to use the computer and whether an attacker is authorized to use data and programs. These two events give the following combinations: external penetration, internal penetration and misfeasance.

External penetration is considered from the prospective of access to the computer and its data/programs. Not just the case of an outsider, who is not part of an organization or its affiliates is considered, but the case of an employee or contractor who has no access to computer resources and data.

Internal penetration is considered for attackers that have access to a computer, but who are not authorized to use certain computer's data and programs. Anderson highlights that in some organizations, internal penetration is more frequent than external penetration, because internal penetrators already have authorization to use computers. An internal penetrator can be a masquerader that could be an outsider who has already gained access to the system, an employee without full access, or an employee that is using others' credentials. An internal penetrator could be a legitimate user of a computer who misuses his or her access permissions to use the system. The clandestine, is considered an internal penetrator, and is the attacker that is able to change in operating systems' parameters in order to hide tracks of the penetration.

Table 1: Percentage of incidents from inside reported by CSI/FBI.

year	1-5	6-10	>10	Don't Know
1997	47%	14%	3%	35%
1998	70%	20%	11%	—
1999	37%	16%	12%	35%
2000	38%	16%	9%	37%
2001	40%	12%	7%	41%
2002	42%	13%	9%	35%
2003	45%	11%	12%	33%
2004	52%	6%	8%	34%
2005	46%	7%	3%	44%

Table 2: Percentage of incidents from outside reported by CSI/FBI.

year	1-5	6-10	>10	Don't Know
1997	43%	10%	1%	45%
1998	74%	18%	9%	—
1999	43%	8%	9%	39%
2000	39%	11%	8%	42%
2001	41%	14%	7%	39%
2002	49%	14%	9%	27%
2003	46%	10%	13%	31%
2004	52%	9%	9%	30%
2005	47%	10%	8%	35%

Anderson's study has been addressed elsewhere [3], [13], [24] and it certainly reached the goal of classification of penetrations, but it is important to have in mind that this seminal work was at a time where interconnecting networks were not a high threat. However, Anderson's work posted the problem and gave the solution, of his time, to the difficulty of defining internal and external penetrations.

## 4. Statistics

Some difficulties were encountered in the goal of presenting the most complete and updated statistics available in free repositories of the internet. This research found a few places with reliable statistics. The presentation of statistics were different—some reported percentages, others raw data—the format changed within the same report making statistical inferences a challenging task. This paper tried to perform some statistical inferences on statistics available, and tried to motivate other researches in pursuing a more rigorous statistical analysis.

### 4.1 Statistics CSI/FBI

Tables 1–3 show totals corresponding to organizations—represented primarily by United States corporations, government agencies, financial institutions, educational institutions, medical institutions and other organizations [7]—that have between 1–5, 6–10 or more than 10 incidents per year. Each row is approximately 100% because numbers are rounded to the nearest integer.

Table 4 was calculated taking the corresponding proportion of *inside incidents* as in Table 1, *outside incidents* as in

Table 3: Total percentage of incidents reported by CSI/FBI.

year	1-5	6-10	>10	Don't Know
1996	46%	21%	12%	21%
1997	48%	23%	3%	27%
1998	61%	31%	9%	—
1999	34%	22%	14%	29%
2000	33%	23%	13%	31%
2001	33%	24%	11%	31%
2002	42%	20%	15%	23%
2003	38%	20%	16%	26%
2004	47%	20%	12%	22%
2005	43%	19%	9%	28%
2006	48%	15%	9%	28%
2007	41%	11%	26%	23%
2008	47%	14%	13%	26%

Table 4: Proportional percentage of incidents from inside, outside and don't know, calculated from CSI/FBI Reports.

year	Respondents	Inside	Outside	Don't Know
1997	48%	40%	33%	27%
1998	45%	50%	50%	—
1999	63%	38%	32%	29%
2000	61%	37%	32%	31%
2001	65%	33%	35%	31%
2002	64%	36%	41%	23%
2003	67%	37%	37%	26%
2004	57%	37%	42%	22%
2005	65%	32%	40%	28%

Table 2, as well as the *don't know* proportion from Tables 1 and 2, with respect to the total presented in Table 3. No data was found in public repositories of the internet, from years 1996, and 2006 – 2009 that give the classification of insiders, outsiders and don't know.

Average of the percentages for inside incidents (37.7%), outside incidents (37.9%), and don't know (24.1%) were calculated from Table 4. These averages were corroborated with the bootstrap technique [14]. One thousand samples of size nine taken randomly with repetition from Table 4 gives 37.77 for the mean of averages of percentages of inside incidents, 37.99 for the mean of averages of percentages of outside incidents, and 24.16 for the mean of averages of percentages of don't know, with an estimated error of 1.606, 1.829 and 3.044 respectively.

As all reports from the CSI/FBI reviewed present statistics in percentages [5], [6], [7], this paper tried to infer the number of incidents. For doing that, over 100 random samples averages of 1 – 5, 6 – 10, 11 – 30, and 31 – 60 incidents were drawn using Tables 1, 2, and Table 3. Table 5 as well as Figure 1 give the corresponding results.

The Pearson Coefficient [18] calculated for the Number of inside incidents and outside incidents, as in Table 5, gives a value of 0.624, which does not show a strong linear correlation between these two variables. The Fisher's coefficient  $\rho$  [15] corroborates such statement with the range  $-0.069 < \rho < 0.910$  that includes the value 0. There is not a linear correlation between the number of inside incidents

Table 5: Number of incidents from inside and outside, inferred from 1997 – 2005 CSI/FBI reports.

year	Inside Incidents			Outside Incidents		
	Respondents	Ave.	Std.	Respondents	Ave.	Std.
1997	218	841.0	53.47	212	562.9	36.09
1998	184	1244.9	33.77	142	948.1	23.30
1999	308	1809.7	43.90	280	1439.0	30.28
2000	359	2092.3	36.83	341	2014.7	62.66
2001	348	1200.2	26.36	316	1614.2	30.23
2002	289	1473.3	36.09	301	1815.5	28.83
2003	328	1537.2	46.96	336	1635.0	39.93
2004	280	1022.2	33.33	280	1152.0	33.23
2005	453	1164.8	30.57	453	1740.9	40.17

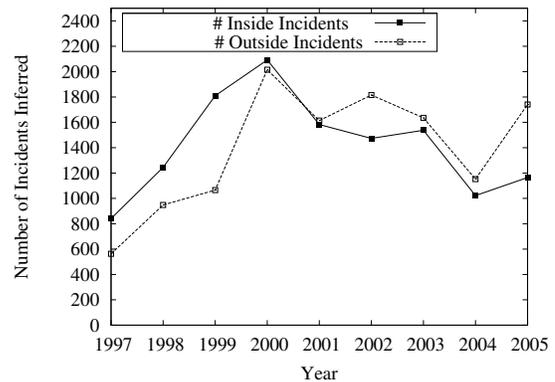


Fig. 1: Mean number of inside and outside incidents. Inferred from 1997 – 2005 CSI/FBI reports.

and the number of outside incidents.

## 4.2 Statistics DataLossDB

Table 6 reports the findings in the database *DataLossDB.org* [11] which records security breaches from a variety of institutions like government, financial, education and medical institutions. Three categories of inside are outlined: *inside incident* that corresponds to someone from inside the company, such as a disgruntled employee, *inside malicious* that is someone who eavesdrops, steals, or damages information, uses information in a fraudulent manner, or denies access to other authorized users, and *inside accidental* that is the result of carelessness or lack of knowledge from an employee [23]. The averages of inside incident, inside malicious and inside accidental are 6.0, 19.5 and 56.3, with percentages of 7.33%, 23.83% and 68.82%.

Table 7, left side, sum the three categories of *inside incident* from table 6, rewrite the number of *outside incidents* and the number of *unknown*. The right side calculates percentages of inside, outside and unknown from 2000 to 2009. Figure 2 shows the number of inside and the number of outside as the left part of table 7. A Pearson coefficient of 0.922 shows a strong correlation between these two data sets. To corroborate previous statement, the Fisher's coefficient

Table 6: Incidents per year found on DataLossDB.org.

Year	Inside			# Outside	# Unk.	Total
	# Inc.	# Mal.	# Acc.			
2000	0	0	2	6	1	9
2001	0	2	6	10	0	18
2002	0	2	2	2	0	6
2003	1	2	0	11	0	14
2004	1	1	3	18	0	23
2005	1	9	22	104	5	141
2006	8	32	134	338	24	536
2007	13	24	76	382	9	504
2008	29	70	141	499	49	780
2009	7	53	177	306	48	591

Table 7: Total number &amp; percentage of incidents per year found on DataLossDB.org.

Year	Number			Percentage		
	Inside	Outside	Unknown	Inside	Outside	Unknown
2000	2	6	1	22%	67%	11%
2001	8	10	0	44%	56%	0%
2002	4	2	0	67%	33%	0%
2003	3	11	0	21%	79%	0%
2004	5	18	0	22%	78%	0%
2005	32	104	5	23%	74%	4%
2006	174	338	24	32%	63%	4%
2007	113	382	9	22%	76%	2%
2008	240	491	49	31%	63%	6%
2009	237	306	48	40%	52%	8%

was calculated giving the range  $0.698 < \rho < 0.981$  that does not include the zero (0) value.

Small values in Table 7 suggest the possibility of outliers. The Quartiles corresponding to the number of inside and outside incidents were calculated. For the data set *Inside*, second column in Table 7,  $Q1 = 3.75$ ,  $Median = 20$  and  $Q3 = 190$ , no outliers were found; and for the data set *Outside*, third column in Table 7,  $Q1 = 9$ ,  $Median = 61$  and  $Q3 = 349$ , no outliers were found. The  $p$ -value of 0.313 shows that the two data sets are not significant different at the 95% confidence level.

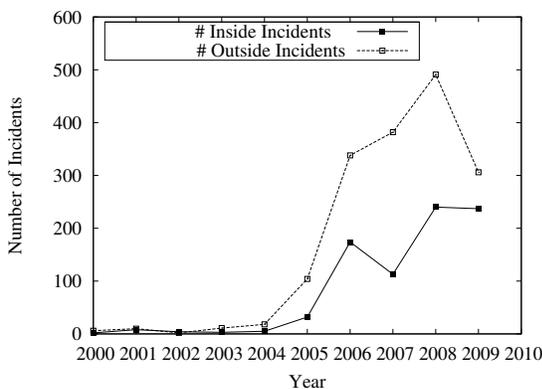


Fig. 2: Number of inside and outside incidents. DataLossDB.org reports at September 2010.

The average of the number of inside, outside and unknown is 81.8, 166.8, and 13.6, which gives the percentages of 31.19% for insiders, 63.61% for outsiders, and 5.18% for unknown. The raw averages were tested with the bootstrap technique as described in Section 4.1, and the corresponding values obtained were 81.77, 163.96, and 13.53 for inside, outside and don't know, with errors of 31.23, 55.09, and 5.88 correspondingly.

### 4.3 Statistics CSO/CERT

Table 8 includes statistics reported by business and government executives, as well as professionals and consultants [9]. This table needs some explanation. The years 2004 and 2005 sums approximately 200%, and this is because the report counts *Outsiders* as 100%, as well as *Insiders* [8]. For example, *don't know* has a value of 30% in the side of *Outsiders*, and 30% in the side of *Insiders*, in other words, it counts as 60%. Year 2006, as well as 2007, sums 300%, because this time the presentation of statistics changed; a new sections reports independently *unknown* adding an additional 100% to the statistics [10]. The report corresponding for 2010 is a little bit more difficult to handle, because it is now giving the mean and median for outsiders, insiders, and unknown that counts for 100% [9]. In *Section Two, numeral 1*, the *CERT* report describes the question about organizations that have experienced a cybersecurity event during the last 12 months—August 2008-July 2009—40% answered *none*, and 60% answered *any*. This is the 40% that appears in Table 8, year 2010, column *None*.

Table 8 shows a new column *None* that is not present in previous statistics—*FBI, DataLossDB.org*. *None* is different from *unknown* that indicates that the organization reports an intrusion but it does not know where it comes from—from inside, outside or unknown.

Finding some statistical inferences this time is more difficult. Table 9 shows the actual percentages used in order to find an average from the years at hand. Now *unknown* is 30% for year 2004, because it was considered as counted twice, one time with the outsiders report and another time for the insiders report—See Table 8. Same case is considered for year 2005, but for the rest of the years, *unknown* is reported independently, not in conjunction with insiders and outsiders. With these assumptions, in average for the years reported by *CERT* as in Table 9, 31.0% for insiders, 48.7% for outsiders, and 20.4% for unknown. The averages were tested with the bootstrap technique as describes in Section 4.1, and the corresponding values obtained were 30.96, 48.62, and 20.33 for insiders, outsiders and don't know, with errors of 0.59, 1.93, and 1.45 correspondingly.

### 4.4 Statistics Verizon

*Verizon* reports *confirmed breaches* that are representative of all breaches in all organizations [25], represented

Table 8: Percentage of incidents reported by CERT.

year	# Resp.	Ins.	Out.	None	Don't Know	Total
2004	342	41%	64%	37%	60%	202%
2005	554	39%	77%	47%	38%	201%
2006	328	55%	80%	128%	37%	300%
2007	443	49%	76%	142%	33%	300%
2010	523	34%	46%	40%	24%	144%

Table 9: Percentage of incidents inferred from CERT reports.

year	Percentage Used			Proportion			Total
	Inside	Outside	Unknown	Inside	Outside	Unknown	
2004	41%	64%	30%	30.4%	47.4%	22.2%	1
2005	39%	77%	19%	28.9%	57.0%	14.1%	1
2006	55%	80%	37%	32.0%	46.5%	21.5%	1
2007	49%	76%	33%	31.0%	48.1%	20.9%	1
2010	34%	46%	24%	32.6%	44.2%	23.2%	1

primarily by retail, financial services, food and beverages, manufacturing, business services and hospitality.

Table 10 estimates are from Figure 6 of the 2009 report [25]. Totals are greater than 100% because of the participation of external and internal with partner associations.

The proportion of internal (22.14), external (71.57) and partner (30.57) were calculated as an average from Table 10, having in mind that outsiders, insiders, and partners included not only themselves but possibly intersections between themselves<sup>1</sup>. As in previous reports, the bootstrap technique was applied to corroborate the averages calculated. The values of 22.08, 71.75, and 30.28 were obtained for insiders, outsiders and partners, with errors of 4.03, 3.81, and 5.28 respectively.

### 5. Analysis

As the focus of this paper is to address the inside vs. the outside threat, it is clear from the *FBI* reports recorded from 1997 to 2005—See Table 4 Section 4.1—that the averages of percentages are approximately of equal proportion for inside incidents (37.7%) and outside incidents (37.9%) with similar estimated errors of 1.606 and 1.829. The number of incidents derived shows a dominant number of inside incidents over outside incidents from the period 1997–2000, and a dominant number of outside incidents over inside incidents from 2001 – 2005—See Figure 1. There was not a linear correlation between the number of inside incidents

<sup>1</sup>Values shown do not sum 100% as per intersections to be addressed on Section 5

Table 10: Percentage of incidents reported by Verizon.

year	Inside	Outside	Partner	Total
2004	12%	92%	8%	112%
2005	28%	60%	41%	129%
2006	15%	75%	40%	130%
2007	16%	65%	44%	125%
2008	18%	73%	39%	130%
2009	20%	74%	32%	126%
2010	46%	62%	10%	118%

and the number of outside incidents inferred from the period 1997 – 2005.

The database *DataLossDB.org* shows the categorization of insiders as inside incident (7.33%), inside malicious (23.83%) and inside accidental that is the bigger threat in this category with 68.82%. These percentages are taken from the averages of data from 2000 – 2009—See Table 6. Including these three categories makes inside incidents 31.2% under outside incidents 63.6%. The two data sets show a positive trend from 2000 – 2009 and a *p* – value = 0.313 shows that the two data sets are not significantly different at the 95% confidence level. There is a strong linear correlation between the number of inside incidents and outside incidents (*pearson – coefficient* = 0.922), but as the data is left skewed, averages of incidents give high errors. The average of inside incidents is 81.77 with an estimated error of 31.23. The average of number of outside incidents is 163.96 with an estimated error of 55.09, and the average of number of unknown is 13.53 with an estimated error of 5.88. However, there is no doubt that the number in outside incidents outperformed the number of inside incidents in all years except 2002. The right part of Table 7 was built in order to show how percentages give some general ideas, but they do not present the real picture. For example, two inside incidents, five inside incidents and 113 inside incidents correspond to the same percentage of 22%.

Making statistical inferences with the data reported from *CERT* is difficult, not only because reports change the way of presenting statistics every two years or so, but because with the data available, it is difficult to derive statistics more useful as the number of penetrations. Given the percentages inferred as in the right part of table 9, outside incidents with 48.7% outperformed inside incidents with 31.0% from 2004 – 2007 and 2010 reports<sup>2</sup>. Taking the percentages as presented by *CERT*—See Table 8, every year the percentage of outsiders outperformed the percentage of insiders.

*Verizon* reports confirmed breaches, so the source of the threat is known. Averages obtained give a proportion of 71.57 of outsiders over 22.14 of insiders, and as Table 10 shows, every year the percentage of outside incidents outperformed the percentage of inside incidents. This time, data was not normalized to 100%, which means that the proportions inferred have some intersections. This is an interesting fact that is shown in Figure 3 inferred from the 2009 report that indicates 43% only by external, 11% only by internal, 7% only by partner and 39% multiple sources [25].

*FBI*, *DatalossDB.org* and *CERT* reported *unknown* or *don't know* incidents. *FBI* reported an average of 24.2% of *don't know* from 1996 – 2008, *DatalossDB.org* reported on average 5.18% of *unknown* from 2000 – 2009—with four years reporting 0, and *CERT* reported an average 20.4% of

<sup>2</sup>2010 reports from August 2008 to July 2009 [9].

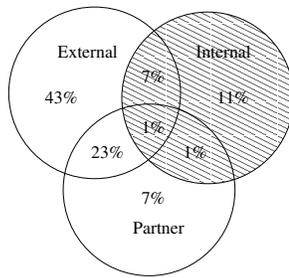


Fig. 3: Percentages of external, internal and partner data breaches as in Table 10. Year 2009. Intersections are inferred.

don't know/unknown from 2004 – 2007 and 2010.

*DatalossDB.org* and *Verizon* reported significant errors as a cause of data breaches; of the inside incidents reported by *DatalossDB.org* on average 68.8% corresponds to inside accidental—See Section 4.2, and *Verizon's* 2009 report gives 67% of cause of breaches due to significant errors [25].

## 6. Conclusions and Future Work

This paper presented and analyzed some statistics about the number/percentage of penetrations coming from inside, or outside organizations or from unknown sources. With this data, the reader could, at least partially, conclude if the general statement says that the highest threat for computers and its resources come from within organizations is true. However, computer security is complex [21] and making such a statement has the likelihood of not being true in some situations. Other variables not considered in this study, like the number of records breached, and the amount of money companies are losing, will improve this research and are part of future work.

Coming from inside the organization, or coming from outside, or coming as a partnership, could help security managers by setting appropriate security mechanisms in place, but coming from unknown sources, or reporting no penetration, should make organizations realize that current security mechanisms are not valid any more, and/or security policies, procedures and standards are not applied as they should be. *Verizon* reported that more than 60% of data breaches were discovered by third parties, and that more than 86% of breaches were avoidable through simple or intermediate controls [25], [26].

The perimeter to secure has been expanded, or maybe there is no perimeter at all [7].

## References

- [1] J. P. Anderson, "Computer security technology planning study," Deputy For Command and Management Systems. HQ Electronic Systems Division (AFSC), Fort Washington, PA, Tech. Rep., 1972.
- [2] —, "Computer security threat monitoring and surveillance," James P. Anderson Co., Fort Washington, PA, Tech. Rep., 1980.

- [3] R. G. Bace, *Intrusion Detection*. USA: MacMillan Technical Publishing, 2000.
- [4] T. Bengtson, "Shazam secure to help bankers with it security," 2005, accessed October 2010. [Online]. Available: <http://www.allbusiness.com/financeinsurance/933194.html>
- [5] CSI Computer Crime and Security Survey Report, "CSI/FBI Computer Crime and Security Survey," 2000, accessed November 2010. [Online]. Available: <http://www.citadel-information.com/library/4/2003-fbi-csi-survey.pdf>
- [6] —, "8th CSI/FBI Computer Crime and Security Survey," 2003, accessed November 2010. [Online]. Available: <http://www.citadel-information.com/library/4/2003-fbi-csi-survey.pdf>
- [7] —, "14th CSI Computer Crime and Security Survey," 2009, accessed November 2010. [Online]. Available: <http://www.personal.utulsa.edu/james-childress/cs5493/CSISurvey/CSISurvey2009.pdf>
- [8] CSO magazine, U.S. Secret Service, CERT Coordination Center, "2004 Cybersecurity Watch Survey — Survey Results," 2004, accessed November 2010. [Online]. Available: [http://www.cert.org/insider\\_threat/](http://www.cert.org/insider_threat/)
- [9] CSO magazine, U.S. Secret Service, CERT Program, Deloitte, "2010 Cybersecurity Watch Survey — Survey Results," 2010, accessed November 2010. [Online]. Available: [http://www.cert.org/insider\\_threat/](http://www.cert.org/insider_threat/)
- [10] CSO magazine, U.S. Secret Service, CERT Program, Microsoft Corp., "2007 Cybersecurity Watch Survey — Survey Results," 2007, accessed November 2010. [Online]. Available: [http://www.cert.org/insider\\_threat/](http://www.cert.org/insider_threat/)
- [11] DATALOSSDB, 2009, accessed July 19/2010. [Online]. Available: [www.datalossdb.org](http://www.datalossdb.org)
- [12] D. Denning, "Cyber security as an emergent infrastructure," 2003, accessed October 2010. [Online]. Available: <http://faculty.nps.edu/dedenning>
- [13] P. A. Diaz-Gomez and D. F. Hougen, "Improved off-line intrusion detection using a genetic algorithm," in *Proceedings of the 7th International Conference on Enterprise Information Systems*, 2005, pp. 66–73.
- [14] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. USA: Chapman & Hall/CRC, 1998.
- [15] GISS, "GISS Goddard Institute for Space Studies," 2009, accessed December 2009. [Online]. Available: <http://icp.giss.nasa.gov/education/statistics/page-3.html>
- [16] S. Harris, *All in One CISSP*. USA: MacGraw Hill, 2008.
- [17] P. Hupston, "How to enhance computer network security," 2009, accessed July 19/2010. [Online]. Available: [http://computeraccessories.suite101.com/article.cfm/how\\_to\\_enhance\\_computer\\_network\\_security#ixzz0t3e7q2Yq](http://computeraccessories.suite101.com/article.cfm/how_to_enhance_computer_network_security#ixzz0t3e7q2Yq)
- [18] D. D. Jensen and P. R. Cohen, "Multiple comparisons in induction algorithms," *Machine Learning*, vol. 38, no. 3, pp. 309–338, 2000.
- [19] M. E. Kabay, "Educational security incidents (esi) year in review," 2009, accessed July 19/2010.
- [20] Salvatore J. Stolfo et. al, *Insider Attack and Cyber Security Beyond the Hacker*. Springer, 2008.
- [21] B. Schneier, *Secrets & Lies*. USA: Wiley Computer Publishing, 2000.
- [22] C. Schou and D. Shoemaker, *Information Assurance for the Enterprise. A Roadmap to Information Security*. USA: McGraw Hill, 2007.
- [23] J. Shah, "The threat within: Protecting information assets from well-meaning employees," 2009, accessed September 2010. [Online]. Available: [www.net-security.org/article.php?id=1289](http://www.net-security.org/article.php?id=1289)
- [24] W. Stallings, *Network Security Essentials. Fourth Edition*. USA: Pearson Prentice Hall, 2011.
- [25] Verizon Business Risk Team, "2009 Data Breach Investigations Report," 2009, accessed November 2010. [Online]. Available: [http://www.verizonbusiness.com/resources/security/reports/2009\\_databreach\\_rp.pdf](http://www.verizonbusiness.com/resources/security/reports/2009_databreach_rp.pdf)
- [26] —, "2010 Data Breach Investigations Report," 2010, accessed November 2010. [Online]. Available: [http://www.verizonbusiness.com/resources/reports/rp\\_2010-data-breach-report\\_en\\_xg.pdf](http://www.verizonbusiness.com/resources/reports/rp_2010-data-breach-report_en_xg.pdf)

# Practical Network Security Teaching in an Online Virtual Laboratory

Christian Willems and Christoph Meinel

Hasso-Plattner-Institute, University of Potsdam, Potsdam, Germany

**Abstract**—*The rapid burst of Internet usage and the corresponding growth of security risks and online attacks for the everyday user or enterprise employee have emerged the terms Awareness Creation and Information Security Culture. Nevertheless, security education has remained an academic issue mainly. Teaching system security or network security on the basis of practical experience inherits a great challenge for the teaching environment, which is traditionally solved using a computer laboratory at a university campus. The Tele-Lab project offers a system for hands-on IT security training in a remote virtual lab environment – on the web, accessible by everyone.*

*An important part of security training focuses on network security: which attacks exist on the different network layers? What is the impact of those attacks? And, how can we secure a network through proper configuration or protective measures like firewalls?*

*The paper at hand briefly presents usage, management and operation of Tele-Lab as well as its architecture. Furthermore, this work introduces the integration of the Virtual Distributed Ethernet technology (VDE) into the Tele-Lab Server and the realization of learning units on network security with complex exercise scenarios such as eavesdropping on local network traffic or Man-in-the-Middle attacks by means of ARP spoofing.*

**Keywords:** Web-based Training, Security Education, Virtual Laboratory, Virtual Machines

## 1. Introduction

Increasing propagation of complex IT systems and rapid growth of the Internet draws attention to the importance of IT security issues. Technical security solutions cannot completely overcome the lacking awareness of computer users, caused by laziness, inattentiveness, and missing education. In the context of awareness creation, IT security training has become a topic of strong interest – as well as for educational institutions as for companies or even individual Internet users.

Traditional techniques of teaching (i.e. lectures or literature) have turned out to be not suitable for security training, because the trainee cannot apply the principles from the academic approach to a realistic environment within the class. In security training, gaining practical experience through exercises is indispensable for consolidating the

knowledge. Precisely the allocation of an environment for these practical exercises poses a challenge for research and development. That is, since students need privileged access rights (root/administrator-account) on the training system to perform most of the imaginable security exercises. With these privileges, students might easily destroy a training system or even use it for unintended, illegal attacks on other hosts within the campus network or the Internet world.

The classical approach is to provide a dedicated computer lab for security training. Such labs are exposed to a number of drawbacks: they are immobile, expensive to purchase and maintain, and must be isolated from all other networks on the site. Of course, students are not allowed to have Internet access on the lab computers. Hands-on exercises on network security topics even demand to provide more than one machine to each student, which have to be interconnected (i.e. a Man-in-the-Middle attack needs three computers: one for the attacker and two other machines as victims).

Tele-teaching for security education consists of multimedia courseware or demonstration software mostly, which does not offer practical exercises. In simulation systems users have a kind of hands-on experience, but a simulator doesn't behave like a realistic environment and the simulation of complex systems is very difficult – especially when it comes to interacting hosts on a network. The Tele-Lab project builds on a different approach for a Web-based tele-teaching system (explained in detail in section 2).

The enhanced Tele-Lab architecture proposed in this paper makes this teleteaching platform even more equivalent to a physical dedicated computer security lab: integration of a virtual networking solution described in section 3 allows to provide training environments for complex exercise scenarios in a dynamic and flexible manner.

Section 4 introduces two learning units on network security – an eavesdropping scenario and the practical application of a Man-in-the-Middle attack – that show the feasibility of this solution. Section 5 summarizes and gives an outlook on future enhancements to the Tele-Lab platform as well as on additional use cases.

## 2. Tele-Lab: A Remote Virtual Security Laboratory

The Tele-Lab platform (accessible at <http://www.tele-lab.org/>, see Figure 1) was initially proposed as a

standalone system [1], later enhanced to a live DVD system introducing virtual machines for the hands-on training [4], and then emerged to the Tele-Lab server [3], [6]. The Tele-Lab server provides a novel e-learning system for practical security training in the WWW and inherits all positive characteristics from offline security labs. It basically consists of a web-based tutoring system and a training environment built of virtual machines. The tutoring system presents learning units that do not only offer information in form of text or multimedia, but also practical exercises. Students perform those exercises on virtual machines (VM) on the server, which they operate via remote desktop access. A virtual machine is a software system that provides a runtime environment for operating systems. Such software-emulated computer systems allow easy deployment and recovery in case of failure. Tele-Lab uses this feature to revert the virtual machines to the original state after each usage.

With the release of the current iteration of Tele-Lab, the platform introduced the dynamic assignment of several virtual machines to a single user at the same time. Those machines are connected within a virtual network (known as *team*, see also in [2]) providing the possibility to perform basic network attacks such as interaction with a virtual victim (i.e. port scanning). A victim is the combination of a suitably configured virtual machine running all needed services and applications and a collection of scripts that simulate user behavior or react to the attacker's actions (see also exemplary description of a learning unit below). A short overview of the architecture of the Tele-Lab platform is given later in this section.

## 2.1 Learning Units in Tele-Lab – an exemplary walkthrough

Learning units follow a straight-forward didactic path beginning with general information on a security issue, getting more concrete with the description of useful security tools (also for attacking and exploiting) and culminating in a hands-on exercise, where the student has to apply the learned concepts in practice. Every section concludes with hints on how to prevent the just conducted attacks.

An exemplary Tele-Lab learning unit on *malware* (described in more detail in [5]) starts off with academic knowledge such as definition, classification, and history of malware (worms, viruses, and Trojan horses). Methods to avoid becoming a victim and relevant software solutions against malware (scanners, firewalls) are presented as well. Afterwards, various existing malware kits and ways of distribution are described in order to prepare the hands-on exercise. Following an offensive teaching approach (see [7] for different teaching approaches), the user is asked to take the attacker's perspective – and hence is able to lively experience possible threats to his personal security objectives. The closing exercise for this learning unit on malware is to *plant a Trojan horse on a scripted victim's*

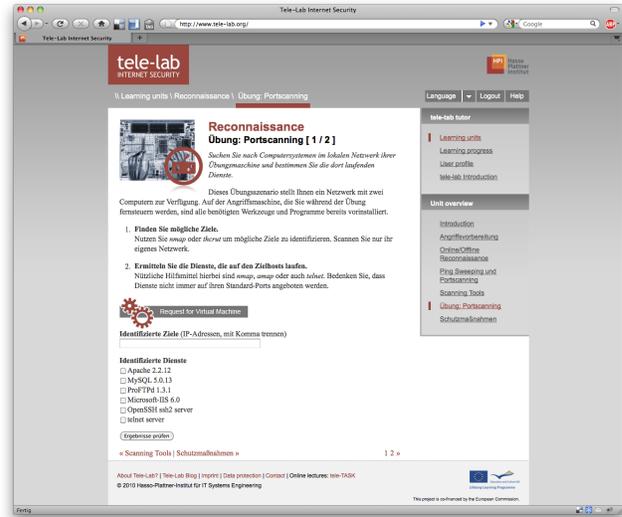


Fig. 1: Screenshot of the Tele-Lab Tutoring Interface

*computer system* – in particular it is the outdated Back Orifice Trojan horse.

Back Orifice (BO) is a Remote Access Trojan Horse developed by the hacker group “Cult of the Dead Cow” (see [9]). In order to distribute the Trojan horse to the attacker, the student has to prepare a carrier for the BO server component and send it to the victim via e-mail. A carrier is usually a “gimmick” application that has actually no useful functionality but installs the Trojan horse server in the background. The script on the victim's virtual machine will answer the mail and indicate that the Trojan horse server has been installed (mail attachment has been opened).

The next step is the application of knowledge gained in a prior learning unit on *Reconnaissance*: in order to find the now vulnerable virtual machine, the network must be scanned for hosts that offer a service on the port used for the Back Orifice server. This can be done using a port scanner like the well-known *nmap* tool. The student can now use the BO client to take control of the victim's system and spy out some private information. The knowledge of that information is the user's proof to the Tele-Lab tutoring environment, that the exercise has been solved successfully.

Such an exercise implies the need for the Tele-Lab user to be provided with a team of interconnected virtual machines: one for attacking (all necessary tools installed), a mail server for e-mail exchange with the victim and a vulnerable victim system (unpatched Windows 95/98 in this case). Remote Desktop Access is only possible to the attacker's VM.

Other learning units are also available on, e.g., authentication, wireless networks, secure e-mail, reconnaissance, firewalls, etc. The system can easily be enhanced with new content.

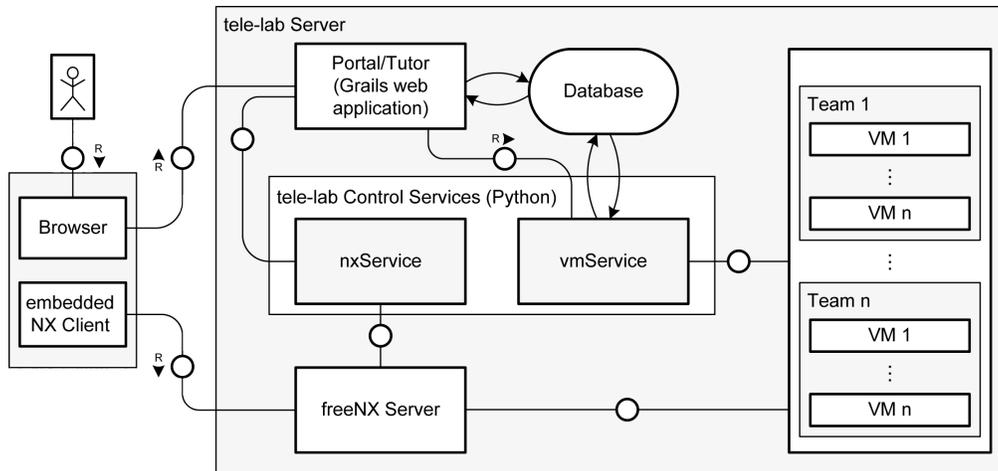


Fig. 2: Overview – Architecture of the Tele-Lab Platform

## 2.2 Architecture of the Tele-Lab Platform

The current architecture of the Tele-Lab server is a refactored enhancement to the infrastructure presented in [6]. Basically it consists of the following components (illustrated in Figure 2).

**Portal and Tutoring Environment:** The Web-based training system of Tele-Lab is a custom *Grails application* running in a *Tomcat application server*. This web application handles user authentication, allows navigation through learning units, delivers their content and keeps track of the students' progress. It also provides controls to request a team of virtual machines for performing an exercise.

**Virtual Machine Pool:** The server is charged with a set of different virtual machines which are needed for the exercise scenarios – the pool. The resources of the physical server limit the maximum total number of VMs in the pool. In practice, a few (3-5) machines of every kind are started up. If all teams for a certain exercise scenario are in use, new instances can be launched dynamically (again depending on the current load of the physical host). Those machines are dynamically connected to teams and bound to a user on request. The current hypervisor solution used for providing the virtual machines is *KVM/Qemu* [10], [11]. The *libvirt* package [16] is used as a wrapper for the virtual machine control. *LVM (Linux Logical Volume Management)* provides virtual hard discs that are capable of copy-on-write-like differential storage. Differential storage is important to save space on the physical hard disc, because the Tele-Lab server holds so called *VM templates* as master images and clones multiple instances of each template for use within the exercise environment. VM templates also contain configuration files defining hardware parameters like memory, number of CPUs, and network interfaces.

**Database:** The Tele-Lab database holds all user information, the content for web-based training and learning unit

structure as well as the information on virtual machine and team templates. *Team templates* are models for connected VMs that allow performing specific exercise scenarios. The database also persists current virtual machine states.

**Remote Desktop Access Proxy:** The Tele-Lab server must handle concurrent remote desktop connections for different users performing exercises. Those connections are proxied using a free implementation of the NX server (*freeNX*, see [12]). The NX server forwards incoming connections to the respective assigned virtual machine accessing the Qemu framebuffer device via *VNC (Virtual Network Computing)*. The NX Client software launched from the student's browser connects to the NX Server using SSH-based authentication: client and server mutually certify each others identity using public-key authentication. Subsequently, the NX Client connects to a specific session with extra user credentials. For mandatory encryption of the remote sessions, NX offers *transport layer security (TLS)*.

**Administration Interface:** The Tele-Lab server comes with a sophisticated web-based administration interface that is also implemented as Grails application (not depicted in Figure 2). The main functionality of this interface is content management for the web-based training environment and user management for the whole platform. Additionally, the admin interface can be used for manual virtual machine control, monitoring and for registering new virtual machines or team templates.

**Tele-Lab Control Services:** Purpose of the central Tele-Lab control services is bringing all the above components together. To realize an abstraction layer for encapsulation of the virtual machine monitor (or hypervisor) and the remote desktop proxy, the system implements a number of *lightweight XML-RPC web services*: the *vmService* and the *remoteDesktopService*. The *vmService* is to control virtual machines – start, stop or recover them, grouping teams or as-

signing machines or teams to a user. The remoteDesktopService is used to initialize, start, monitor, and terminate remote desktop connections to machines, which are assigned to students when they perform exercises. The above-mentioned Grails applications (portal, tutoring environment, and web admin) let the user and administrators control the whole system using the web services.

On the *client side*, the user needs a web browser supporting SSL/TLS and the appropriate Java-plugin for the browser only. For the remote desktop connections, the *NX WebCompanion* is included in the tutoring web application. The WebCompanion is a launcher application for the NX Client implemented as Java applet.

### 3. Virtual Networking for Tele-Lab

As already stated, many scenarios for exercises in security training demand for a networked environment. Exercises on single host training systems are limited to very few tasks that could possibly also be performed on a physical local system without any harm. More interesting and complex exercises (like the malware learning unit described in section 2) and especially exercises on network security as introduced later in section 4 cannot be performed without a training environment providing machines that are connected within a local network.

Earlier implementations of Tele-Lab could connect virtual machines combined to a team using multicast groups: each team is provided with an individual multicast socket that is connected to each team member's virtual network. Routing, firewall, and virtual network devices on the physical host are dynamically configured to separate the network segments from each other. Each multicast group (VM team) can communicate internally only.

To understand this idea, we have to explain the networking concept of Qemu in detail: the virtualization suite sets up a *VLAN (virtual LAN)* for each Qemu process. Those VLANs can be understood as virtual hubs, where you can attach virtual network interfaces – such as the one of the virtual machine running in that process. All attached interfaces to a VLAN intercept all packages sent via that virtual hub. To connect the VLANs of a team of virtual machines, Tele-Lab connects a multicast socket to the virtual LAN of each machine belonging to the respective team, when it starts up. This technique for setting up a virtual network in a Tele-Lab team limits the resulting virtual Ethernet-based networks to:

- LAN segments with a hub (no switched networks)
- simple network structures: no routing, no internet-working (interconnection of networks)
- static IP addresses for the VM templates: this limits the reusability of VM templates, i.e. if one wants to have more than one instance of the same virtual machine in one exercise scenario (respectively team template)

Since the paradigm for Tele-Lab is to provide a training environment being as realistic as possible, the integration

of software-emulated networking devices to overcome the above limitations is a highly desirable enhancement.

#### 3.1 Virtual Distributed Ethernet (VDE)

A suitable solution for more sophisticated networking within the VM teams in Tele-Lab exists with the *Virtual Distributed Ethernet (VDE)* project [8]. VDE is a system which consistently emulates all aspects of Ethernet networking on the data-link layer in a completely realistic manner. VDE maps hardware devices from the physical world – like switches, plugs and cables – on software running in user-mode. The main components of a VDE installation are:

*VDE switch* – a highly customizable software emulation of an Ethernet switch. It supports VLANs, different operation modes (switch/hub), cascading several VDE switches (including Spanning Tree Protocol), and extensive command line management. You can attach different kinds of network interfaces, such as TUN/TAP interfaces, QEMU/KVM-based virtual machines, and VDE plugs. *TUN* and *TAP* are virtual network devices provided by the Linux kernel. While *TAP* (as in network tap) simulates an Ethernet device and operates on ISO/OSI layer 2, *TUN* (as in network TUNnel) simulates a network layer device and operates with layer 3 packets (i.e. IP packets).

*VDE plug* – the virtual counterpart of an Ethernet plug can be connected to a VDE switch. It sends all data from the standard input to the VDE switch which is connected to and writes all data from the virtual switch to standard output. A tool named *dpipe* (a bi-directional pipe) can connect two VDE plugs to a virtual cable by diverting the standard output of one VDE plug to the standard input of the other one (and vice-versa). *wirefilter* is an enhanced version of *dpipe*, which also allows for simulating problems and limitations from the physical world like packet loss, duplicated packets, limited bandwidth or different MTUs.

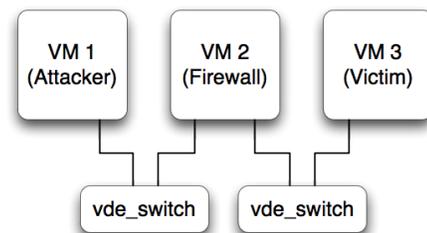


Fig. 3: Exemplary Deployment for Virtual Distributed Ethernet in Tele-Lab

A possible VDE setup with virtual machines for a complex Tele-Lab learning unit may look like illustrated in Figure 3: let the task be a remote exploitation of VM 3, the attacker uses VM 1. While this would be an easy task if attacker and victim would be connected to the same local network, it gets much more challenging as soon as there is a firewall

between the attacker and target host. The use of two VDE switches, both connected to different network interfaces of the firewall host (VM 2) allows to compile such an exercise scenario.

VDE switches and VDE plugs can also be connected if they run on different physical hosts, which is also a useful feature for a further enhanced Tele-Lab architecture (see Outlook in section 5).

### 3.2 Integrating VDE into the Tele-Lab Architecture

When Tele-Lab creates a new team of virtual machines, the *vmService* (see Figure 2) is responsible for starting Qemu processes for each VM and for setting up the virtual network that connects the team members. It consumes a team configuration provided as XML file and transforms its elements to parameters for command line calls. Such an XML file for the example configuration from Figure 3 would look like depicted in Figure 4 (without attributes not relevant for virtual networking):

```
<tl:team name="Example Team" >
  <!-- virtual machine instances -->
  <tl:machine name="VM 1 (Attacker)">
    <tl:networkInterface
      mac="00:11:22:33:44:55"
      networkName="net0" />
  </tl:machine>
  <tl:machine name="VM 2 (Firewall)">
    <tl:networkInterface
      mac="11:22:33:44:55:66"
      networkName="net0" />
    <tl:networkInterface
      mac="22:33:44:55:66:77"
      networkName="net1" />
  </tl:machine>

  <tl:machine name="VM 3 (Victim)">
    <tl:networkInterface
      mac="33:44:55:66:77:88"
      networkName="net1" />
  </tl:machine>

  <!-- virtual network -->
  <tl:network name="net0" id="1"
    mode="switch" />
  <tl:network name="net1" id="2"
    mode="switch" />
</tl:team>
```

Fig. 4: Exemplary XML Team Configuration

After parsing the XML data, the *vmService* starts up virtual machines from the VM templates identified by the `<tl:machine>` element and initiates the respective network interfaces specified with the enclosed `<tl:networkInterface>` items, i.e. two interfaces for the firewall machine (VM 2).

It also starts an instance of *VDE Switch* for each virtual network specified with `<tl:network>`, either as hub or as switch depending on the mode value. The network interfaces of the virtual machines are bound to the matching switch instances.

The assignment of IP addresses inside the virtual machines posed a challenge during implementation, since they had to be allocated dynamically. An obvious solution was to attach a *DHCP server* to each VDE switch after startup using TAP interfaces. This DHCP server assigns an IP address to each of the virtual machines connected to the virtual switch based on its MAC address. IP addresses for the virtual machines can also be defined in the team configuration file. If an administrator decides to do so, the DHCP server for the respective team is dynamically configured to issue those defined IP addresses to the network interface with the corresponding MAC address.

Due to security constraints, users of virtual machines in Tele-Lab should not be able to access any services running on the physical host. For this reason, the DHCP server and the TAP interface are shut down, after the DHCP leases have been issued.

The generation of the above described XML representations of virtual networks will be realized as a web based tool: Tele-Lab administrators can use a convenient interface to combine virtual machine templates to a team and define the network connections for the team members.

## 4. Network Security Exercise Scenarios

There are a lot of conceivable exercise scenarios in the area of network security, which require the provision of a networked training environment. Two such exercises have already been introduced earlier in this paper: the *malware* learning unit from section 2 needs three hosts on a network (attacker and victim machines, mail server). The exemplary scenario on *remote exploitation* outlined in section 3 requires three hosts on two different networks. In the following, two more learning units on network security are presented briefly.

### 4.1 Exercise Scenario: Eavesdropping of Network Traffic

Eavesdropping is basically about secret listening to some private communication of two (or more) communication partners without their consent. In the domain of computer networks, the common technique for eavesdropping is *packet sniffing*. There are a number of tools for packet sniffing – *packet analyzers* – freely available on the Internet, such as the well-known *tcpdump* or *Wireshark* [13] (used in this learning unit).

A learning unit on packet sniffing in a local network starts off with an introduction to communication on the data-link layer (Ethernet) and explains the difference between a network with hub and a switched environment. This is important for eavesdropping, because this kind of attack is

way easier when connected to a hub. The hub will forward every packet coming in to all its ports and hence to all connected computers. These hosts decide, if they accept and further compute the incoming data based on the MAC address put in the destination field of the Ethernet frame header: if the destination MAC is the own MAC address, the Ethernet frame is accepted, or dropped otherwise. If there is a packet analyzer running, also frames not intended for the respective host can be captured, stored and analyzed. This situation is different in a switched network: the switch does not broadcast incoming data to all ports but interprets the MAC destination to “switch” a dedicated line between source and destination ports. In consequence, the Ethernet frame is only delivered to the actual receiver.

After providing general information on Ethernet-based networking, the learning unit introduces the idea of packet sniffing and describes capabilities and usage of the packet analyzer *Wireshark*, especially how to capture data from the Ethernet device and how to filter and read the captured data.

The practical exercise presents the following task to the learner: “*Sniff and analyze network traffic on the local network. Identify login credentials and use them to obtain a private document.*” The student is challenged to enter the content of this private document to proof, that she has solved the task.

When requesting access to a training environment, the user is assigned to a team of three virtual machines: the attacker machine equipped with the *Wireshark* tool, and two machines of (scripted) communication partners: Alice and Bob. In this scenario, Bob’s machine hosts an FTP server and a Web server, while Alice’s VM runs a script that generates traffic by initiating arbitrary connections to the services on Bob’s host. Among those client/server connections are successful logins to Bob’s FTP server. As this learning unit focuses on sniffing and the interpretation of the captured traffic, the machines are connected with a hub. There is no need for the attacker to get into a Man-in-the-Middle position in order to capture the traffic between Alice and Bob.

Since FTP does not encrypt credentials, the student can obtain username and password to log in to that service using the stolen credentials. On the server, the student finds a file called *private.txt* that contains the response to the challenge mentioned above.

The lesson concludes with hints on preventing eavesdropping attacks, such as the usage of services with secure authentication methods (i.e. SFTP or ftps instead of plain FTP) and data encryption.

## 4.2 Exercise Scenario: Man-in-the-Middle Attack with ARP Spoofing

The general idea of a Man-in-the-Middle attack (MITM) is to intercept communication between two communication partners (Alice and Bob) by initiating connections between

the attacker and both victims and spoofing the identity of the respective communication partner (Fig. 5). More specific, the attacker pretends to be Bob and opens a connection to Alice (and vice versa). All traffic between Alice and Bob is being relayed via the attackers computer. While relaying, the messages can be captured and/or manipulated.

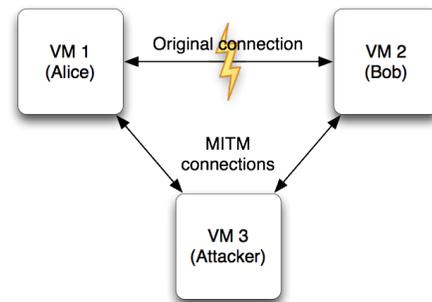


Fig. 5: General Idea of Man-in-the-Middle Attacks

MITM attacks can be implemented on different layers of the TCP/IP network stack, i.e. *DNS cache poisoning* on the application layer, *ICMP redirecting* on the Internet layer or *ARP spoofing* in the data-link layer. This learning unit focuses on the last-mentioned attack, which is also called *ARP cache poisoning*.

The Address Resolution Protocol (ARP) is responsible for resolving IP addresses to MAC addresses in a local network. When Alice’s computer opens an IP-based connection to Bob’s one in the local network, it has to determine Bob’s MAC address at first, since all messages in the LAN are transmitted via the Ethernet protocol (which only knows about the MAC addresses). If the Alice only knows the IP address of Bob’s host, (i.e. 192.168.0.10) she performs an *ARP request*: Alice sends a broadcast message to the local network and asks, “*Who has the IP address 192.168.0.10?*” Bob’s computer answers with an *ARP reply* that contains its IP address and the corresponding MAC address. Alice stores that address mapping in her *ARP cache* for further communication.

*ARP spoofing* [14] is basically about sending forged ARP replies: referring to above example, the attacker repeatedly sends ARP replies to Alice with Bob’s IP address and the own MAC address – the attacker pretends to be Bob. When Alice starts to communicate with Bob, she sends the ARP request and instantly receives one of the forged ARP replies from the attacker. She then thinks, the attackers MAC address belongs to Bob and stores the faked mapping in her ARP cache. Since the attacker performs the same operation for Alice’s MAC address, he can also manage to imply Bob, that his MAC address is the one of Alice. In consequence, Alice sends all messages to Bob to the MAC address of the attacker (same for Bob’s messages to Alice). The attacker just has to store the original MAC addresses of Alice and

Bob to be able to relay to the original receiver.

A learning unit on ARP spoofing begins with general information on communication in a local network. It explains the Internet Protocol (IP), ARP and Ethernet including the relationship between the two addressing schemes (IP and MAC addresses).

Subsequently, the above attack is described in detail and a tool, that implements ARP spoofing and a number of additional MITM attacks is presented: *Ettercap* [15]. At this point, the learning unit also explains what the attacker can do, if he becomes Man-in-the-Middle successfully, such as specifying *Ettercap filters* to manipulate the message stream.

The hands-on exercise of this chapter asks the student to perform two different tasks. The first one is the same as described in the exercise on packet sniffing above: “*monitor the network traffic, gain FTP credentials and steal a private file from Bob's FTP server*”. The training environment is also set up similar to the prior scenario. The difference is that the team of three virtual machines is connected through a virtual switch this time (instead of a hub), so that capturing the traffic with Wireshark would not reveal the messages between Alice and Bob. Again, the student has to proof the successful attack by putting in the content of the secret file in the tutoring interface.

The second (optional) task is to apply a filter on the traffic and replace all images in transmitted HTML content by an image from the attackers host (which would be displayed in Alice's browser). This attack is still working and dangerous in many currently deployed local network installations. The only way to protect oneself against ARP spoofing would be the usage of SSL with a careful verification of the hosts certificate, which is explained in conclusion of the learning unit.

A future enhancement of the practical exercise on ARP spoofing would be the interception of an SSL secured channel: Ettercap also allows a more sophisticated MITM attack including the on-the-fly generation of faked SSL certificates, which are presented to the victims instead of the original ones. The Man-in-the-Middle can then decrypt and re-encrypt the SSL traffic when relaying the messages

## 5. Conclusions and Outlook

The paper at hand presents a comprehensive infrastructure for a remote virtual computing lab for security education. The described enhancements with the Virtual Distributed Ethernet software suite allows the implementation of training environments for complex network security exercises, such as the learning units on packet sniffing and ARP spoofing.

Future work on the system includes the creation of more learning units in the network security domain as well as the implementation of technical enhancements. Additional learning units may cover topics like other Man-in-the-Middle attacks (i.e. the above mentioned DNS cache poisoning),

firewall configuration, intrusion detection and prevention, etc.

Technical enhancements planned for the next iterations of the Tele-Lab server are

- integrating a convenient administration interface for the creation of team templates, precisely a graphical editor for virtual networks, where you can drag and drop virtual machine templates, switches and network cables,
- switching the Remote Desktop Access from NX to an HTML5/AJAX based VNC client (i.e. *noVNC*, see <http://kanaka.github.com/noVNC/>),
- the implementation of tools for remote collaborative learning and tutoring (e.g. Remote Desktop Assistance),
- and clustering on application level to provide larger virtual machine pools.

The clustering enhancement will allow users of interconnected Tele-Lab servers to use virtual machines running on other physical hosts than the one known to the user. The integration of VDE even allows having the virtual machines of one team running on different physical machines.

## References

- [1] J. Hu, M. Schmitt, C. Willems, and C. Meinel. “A tutoring system for IT-Security”, in *Proceedings of the 3rd World Conference in Information Security Education*, p. 51–60, Monterey, USA, 2003.
- [2] C. Border. “The development and deployment of a multi-user, remote access virtualization system for networking, security, and system administration classes”, *SIGCSE Bulletin*, 39(1): p. 576–580, 2007.
- [3] J. Hu, D. Cordel, and C. Meinel. “A Virtual Machine Architecture for Creating IT-Security Laboratories”, Technical report, Hasso-Plattner-Institut, 2006.
- [4] J. Hu and C. Meinel. “Tele-Lab IT-Security on CD: Portable, reliable and safe IT security training”, *Computers & Security*, 23:282–289, 2004.
- [5] C. Willems and C. Meinel. “Awareness Creation mit Tele-Lab IT-Security: Praktisches Sicherheitstraining im virtuellen Labor am Beispiel Trojanischer Pferde”, in *Proceedings of Sicherheit 2008*, p. 513–532, Saarbruecken, Germany, 2008.
- [6] C. Willems and C. Meinel. “Tele-Lab IT-Security: an Architecture for an online virtual IT Security Lab”, *International Journal of Online Engineering (iJOE)*, X, 2008.
- [7] W. Yurcik and D. Doss. “Different approaches in the teaching of information systems security”, in *Security, Proceedings of the Information Systems Education Conference*, p. 32–33, 2001.
- [8] R.Davoli. (2011) Virtual Distributed Ethernet homepage. [Online]. Available: <http://vde.sourceforge.net/>
- [9] Cult of the Dead Cow. (2011) Back Orifice – Windows Remote Administration Tool homepage. [Online]. Available: <http://www.cultdeadcow.com/tools/bo.php>
- [10] Red Hat, Inc. (2011) Kernel-based Virtual Machine (KVM) homepage. [Online]. Available: <http://www.linux-kvm.org/>
- [11] F. Bellard. (2011) QEMU – Open Source Processor Emulator homepage. [Online]. Available: <http://www.qemu.org/>
- [12] F. Franz. (2011) FreeNX – the free NX project homepage. [Online]. Available: <http://freex.berlios.de/>
- [13] Wireshark Foundation. (2011) Wireshark homepage. [Online]. Available: <http://www.wireshark.org/>
- [14] S. Whalen. (2011) An Introduction to ARP Spoofing. [Online]. Available: [http://www.rootsecure.net/content/downloads/pdf/arp\\_spoofing\\_intro.pdf](http://www.rootsecure.net/content/downloads/pdf/arp_spoofing_intro.pdf)
- [15] A. Ornaghi and M. Valleri. (2011) EttercapNG homepage. [Online]. Available: <http://ettercap.sourceforge.net/>
- [16] The Libvirt Developers. (2011) libvirt – The virtualization API homepage. [Online]. Available: <http://libvirt.org/>

# Design and Implementation of a Critical Infrastructure Security and Assessment Laboratory

Guillermo A Francia III, Nouredine Bekhouche, and Terry Marbut

Jacksonville State University  
Jacksonville, Alabama

**Abstract** - *The globally-connected information superhighway, known as cyberspace, ushered our dependence on information technology to support our critical infrastructure. In a recent study [1] conducted by the United States Government Accountability Office (GAO) on critical infrastructure protection, the lessons learned from the first Cyber Storm exercise have yet to be fully addressed. In October, 1997, the report of the President's Commission on Critical Infrastructure Protection acknowledged that there is a widespread capability to exploit critical infrastructure vulnerabilities [2]. In October 2001, the Bush administration created the President's Critical Infrastructure Protection Board through Executive Order 13231. The Board and the Department of Energy have developed the non-prescriptive twenty-one (21) steps to improve cybersecurity of SCADA networks [3]. The research associated with this paper will serve as an instrument to facilitate the realization of each of those recommended 21 steps. And more specifically, this paper presents the design and implementation of an experimental Critical Infrastructure Security and Assessment Laboratory (CISAL) and activities associated with it. The laboratory is envisioned to be a training facility for future computer security professionals.*

**Keywords:** Cybersecurity, critical infrastructure, SCADA, industrial controls, vulnerability assessment, information assurance.

## 1 Introduction

Supervisory Control and Data Acquisition (SCADA) systems are made up of instruments, computers, and applications that provide controls and data acquisitions for essential commodities and services for our daily sustenance and activities. It is obvious that they play a major role in our critical infrastructure and require a high level of protection from threats propagated through physical or electronic media. These systems together with Programmable Logic Controllers (PLC) are widely used in the industrial sectors and critical infrastructures (private or federal). Typical industrial

applications include electric grid, water and wastewater, oil and natural gas, chemical, transportation, pharmaceutical, pulp and paper, food and beverage, and discrete manufacturing such as aerospace, automotive, and durable goods. The interdependence of critical infrastructures has been investigated extensively and an article in the December 2001 IEEE Control Systems Magazine identifies SCADA communications as a common link between these systems. SCADA systems as part of the central station, shown in Figure 1, collect data and perform centralized monitoring and control (remote control in many cases). The data is collected from distributed controllers, sensors, and communication devices that may be scattered over a wide geographical area. PLCs are industrial computers used to control equipment and processes and represent the distributed (local) controllers depicted in Figure 1. The PLC is equipped with a variety of communication capabilities that allow the PLC to communicate with the SCADA system and with the field devices that the PLC controls. These field devices include sensors, valves, and motors. These capabilities are referred to as the Network and the Device Networks in the block diagram depicted in Figure 1.

Leading manufacturers of the type equipment used in these systems include Rockwell Automation, GEFanuc, Modicon (Schneider Electric), and Siemens. These companies provide a variety of communications options that utilize the latest technologies to increase the flexibility, efficiency, and safety of the control systems' operations. Traditionally, a PLC (Programmable logic controller) would communicate with a slave machine using one of several possible open or proprietary protocols, such as Modbus, Profibus, CANopen, or DeviceNet. Nowadays, Ethernet is increasingly used as the link-layer protocol, with one of the above protocols as the application-layer. These communications options exist in both wired and wireless configurations (802.15, Wireless HART, Zigbee, and ISA100.11a as examples) and may pose potential cyber security risks. Many critical infrastructure sites utilize Remote Terminal Units (RTU) as part of their control strategies since different functional areas may be separated by significant geographical distances. The RTU provides needed

control at remote areas. Often, telecontrol strategies are used to enable remote control, remote signaling, and remote maintenance of these areas. Typically these systems can include all forms of transmission media (dedicated line, wireless, dial-up network, and mobile wireless) and represent a significant concern regarding system security. Further, the shift to next generation standards based protocols such as the IEC 61850 is underway. These next generation protocols are based on the common information models (CIM) as described in [5]. The main concern about these protocols is the lack of cyber security benchmarks or standards to which they can be measured, controlled, and improved.

The popular technologies used in the wireless network of SCADA systems include the following:

- (1) Wi-Fi: the IEEE standard 802.11 specifies the technology for the Wireless Local Area Network (WLAN), commonly known as Wi-Fi. Wi-Fi works in the 2.4GHz ISM band;
- (2) WiMax: WiMax is a wireless option for wired broadband communication. It is defined in the IEEE 802.16 standard and

has a data rate capability of 100Mbps. The technology supports transmissions over extended range which makes it ideal for controlling devices over long distances;

- (3) Industrial Radios: many vendors offer 900 MHz and 2.4 GHz industrial radios operating in the ISM band. They provide low data rates ranging between 9600bps up to 1.5 Mbps using Frequency Hopping Spread Spectrum (FHSS);
- (4) Unlicensed band: Wireless ISP use of unlicensed spread spectrum using between 900 MHz and 2.4 GHz. Many small towns offer wireless ISP services. This technology is capable of transporting SCADA data between substations and the co-op office. Proper security is highly recommended;
- (5) Cellular 3G: Commercial 3G digital cellular is a very viable technology for feeder or SCADA, but it has its own set of challenges, just like other technologies;
- (6) General Packet Radio Service (GPRS): GPRS is a packet-based radio service that enables an “always on” connection.

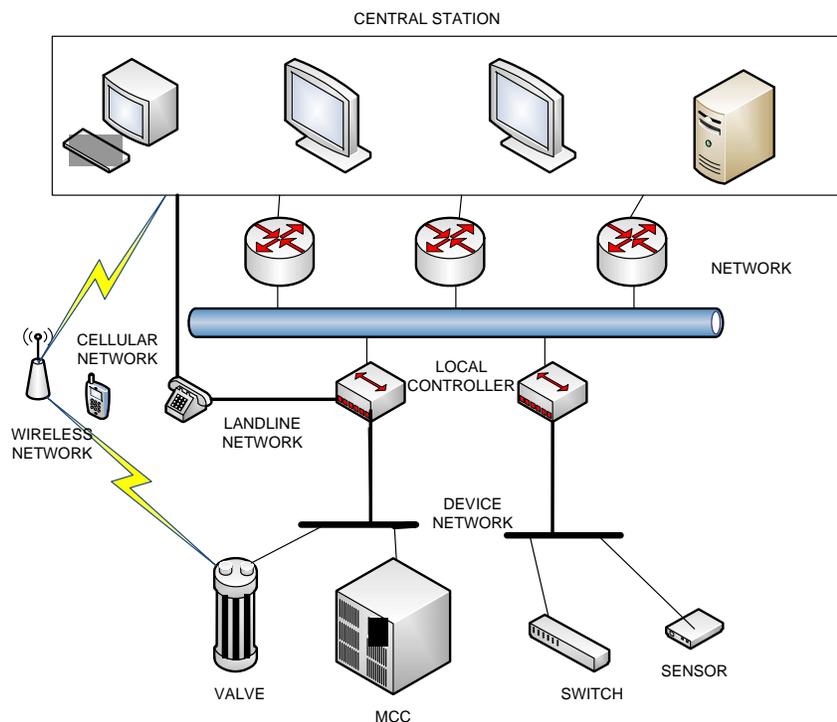


Figure 1. A Typical SCADA System Configuration

The lack of authentication or confidentiality mechanism in almost all SCADA protocols makes them vulnerable to attacks. Exacerbating this deficiency is the fact that RTUs are not physically secured. Most often in SCADA systems, passwords are sent in the clear and there is no way to authenticate the master server. By sending a false control message from a computer connected to the Internet, anyone can manipulate traffic signals, power switching stations, process-control systems, or sewage-water valves, creating major concerns to public safety and health.

Additionally, many SCADA servers and RTUs make use of the Microsoft Windows operating system. The reason for the prevalence of the Windows operating systems on SCADA systems is its familiarity among engineers and technicians. Windows systems handle access control policies by attaching multiple privileges to different types of objects which, possibly turns into vulnerabilities. SCADA systems running Windows will most certainly suffer from these same vulnerabilities.

## 2 Related Research Works

The interest on critical infrastructure security is steadily growing. Published literature on SCADA security protection, vulnerabilities, and standards are currently and steadily being generated. SCADA system security testbeds are being implemented and utilized for checking and mitigating vulnerabilities. Davis, et.al. [4] described the development of a testbed designed to assess the vulnerabilities introduced in a power system infrastructure when connected to public networks. The testbed is comprised primarily of a software-based simulation system that models a networked SCADA system. The testbed is used to study the effects of Distributed Denial of Service (DDoS) attacks on a simulated SCADA system for a power generation plant.

Giani, at. al. [6] described a SCADA security testbed that uses simulation, emulation, and implementation-based techniques to realize a reference SCADA architecture. The testbed is used to implement three types of experimental attacks: denial of service attacks on sensors, integrity attacks on sensor outputs, and phishing attacks on a web server to gain access to protected information. The development of a novel SCADA system for laboratory testing in a test facility at Murdoch University in Western Australia is described by Patel, et. al. [7]. The dynamic SCADA testing system is designed using a graphical programming language, LabView, and its Data logging and Supervisory Control (DSC) module for testing various system configurations. Although the SCADA test facility is not designed for

security research, it provides some important insights on how a SCADA security testbed should be developed.

The National SCADA Test Bed (NSTB) at the Idaho National Laboratory (INL) is a national facility for securing SCADA communications and controls within the energy sector [8]. It provides the necessary expertise and resources in identifying and mitigating critical security flaws in control systems and equipment. Current activities of the national test bed are listed in [8] as:

- Vulnerability assessments of vendor control systems;
- Development of integrated intrusion detection, prevention, and event correlation capability for control system applications;
- Development of integrated cyber risk analysis capability for the energy sector;
- Development of cost-effective methods for secure communication between control centers and remote devices; and
- Development of next generation architecture designs and cyber security solutions.

The intent of the CISAL facility is not to duplicate NSTB's activities and projects but to augment its focus on an area of critical need. Further, we want to be a catalyst for stimulating research and education in the STEM disciplines by providing a facility that is openly accessible to the academic community—a feature which may not be easily attained in a national laboratory setting.

## 3 Objectives

The objectives of the CISAL project are as follow:

- 1) To design and implement a testing and assessment laboratory for critical infrastructure security research.
- 2) To investigate existing vulnerabilities of critical infrastructure systems, execute penetration tests, and design corresponding remediation measures.
- 3) To develop benchmarks to define critical infrastructure security standards and to create compliance audit tools for those standards.
- 4) To develop security mechanisms to augment standard communication protocols (ModBus, DF1, DNP3, OPC) on SCADA systems and to investigate the improvement of SCADA control process through an authentication mechanism using Public Key Infrastructure (PKI) digital signatures.

- 5) To facilitate industry participation in extending the laboratory capability in critical infrastructure security research in various sectors.
- 6) To investigate the impact of wireless technologies on critical infrastructure protection.
- 7) To develop a secure operating system that can be used for critical infrastructure controls.

- 8) To investigate risk assessment methodologies and formulate a generic methodology that will be specific to critical infrastructure protection.

### 4 Design Issues and Schematics

Figure 2 illustrates the overall functional layout of the laboratory.

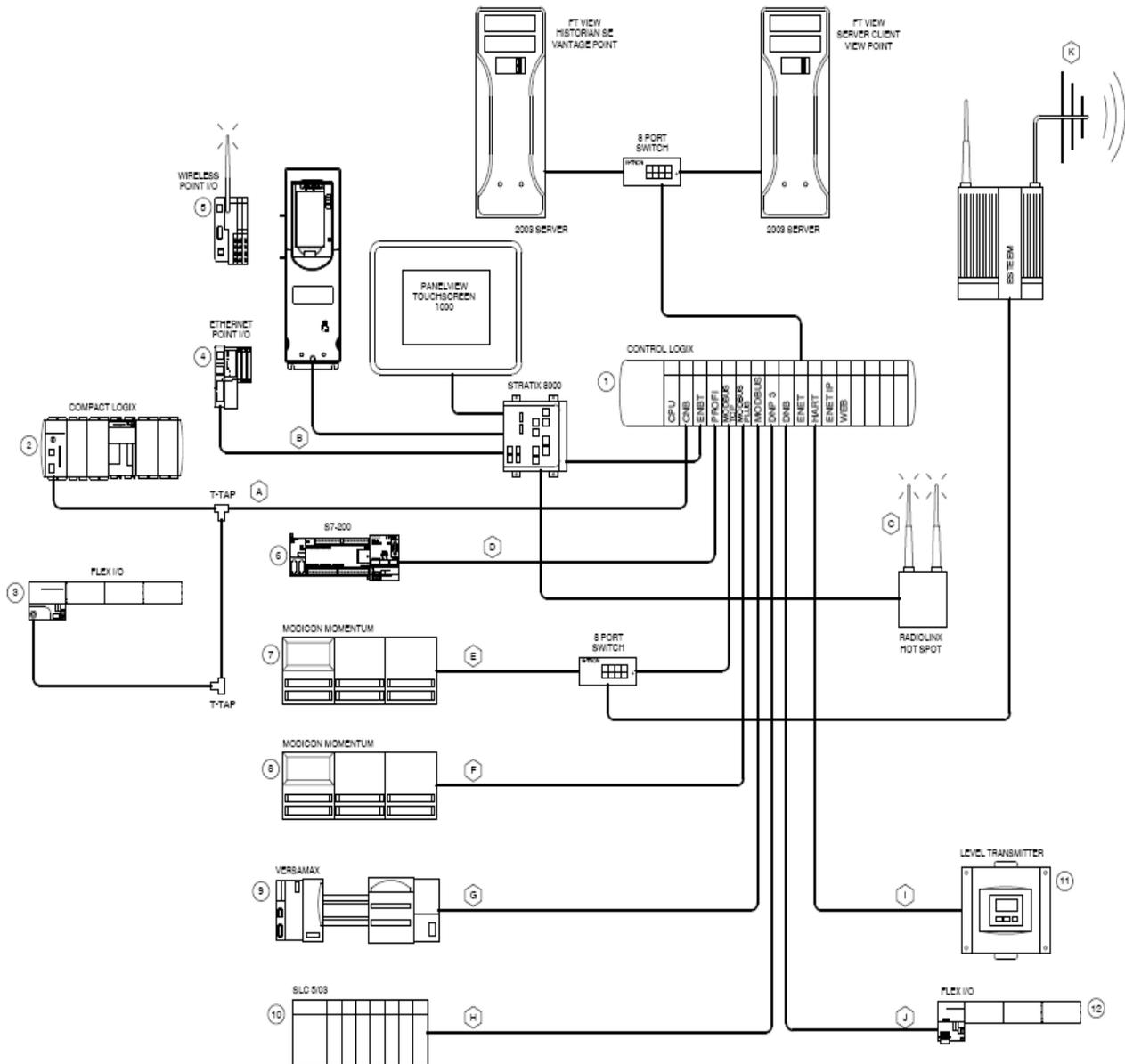


Figure 2. The Main Laboratory Panel

The laboratory equipment is actually housed in five separate enclosures. The primary enclosure is designed around a master controller utilizing Allen Bradley

Control Logix PLC equipment with a PanelView HMI interface communicating through a Stratix 8000 Ethernet Switch. The Control Logix equipment was selected

based on its popularity among critical infrastructure systems surveyed and its ability to communicate effectively with the communications protocols utilized by other control equipment manufacturers. The master controller provides reporting, and direct and/or supervisory control to various servers, field devices and distributed controllers that represent some aspect of a typical critical infrastructure control strategy. The primary enclosure also houses other Allen Bradley control equipment including Flex I/O, Control Logix and SLC5/03 controllers.

A second enclosure is designed to incorporate the various communication protocols utilized by other control equipment manufacturers. This enclosure includes Modicon (Modicon Momentum), GE/FANUC (VersaMax), and Siemens PLC equipment (S7-200), in order to provide researchers with the ability to investigate potential vulnerabilities in any of the protocols associated with these controllers. Three of the distributed controllers (Modicon Momentum, Micrologix 1400, and Micrologix 1100) were designed as portable units and housed in separate enclosures. These units all communicate with the master controller through different wireless protocols and will facilitate demonstrations and training at off-campus locations and vulnerabilities associated with wireless communications.

Three different types of RTUs are used in the system. Each RTU has a local processor (PLC) made by a different company from the other RTUs. Each local controller exchanges data with the central controller through a wireless communications system. Each RTU has an operator display (human machine interface) to show information related to the process such as pump status, liquid level, etc. The operator displays are made by the same company (Automation Direct). Each RTU uses a different communication method and is briefly described below:

- RTU 1 consists of Modicon Momentum PLC, an operator display, and an ESTEEM 900 MHz Ethernet serial radio for communication purpose.
- RTU 2 consists of Micrologix 1400 AB processor, an operator display, and an RADIOLINX Hot Spot 802.11b/g WiFi for communication purpose.
- RTU 3 consists of Micrologix 1100 AB processor, an operator display, and a DISI Cellular modem to provide a cellular communication capability using any cell phone provider.

## 5 Conclusions and Future Plans

This paper outlined the design and implementation of an ongoing experimental critical infrastructure security and assessment laboratory. The laboratory is designed to emulate, as well as simulate, the control systems that are prevalent in real-world critical infrastructures. Thus, the list of installed equipment includes a variety of hardware (both legacy and state-of-the-art) from various manufacturers. The activities and projects in this laboratory will be designed and structured to provide practical experiences while illustrating theory in the pertinent research areas.

The challenge for the authors will be in the continual development of these activities and the introduction of novel practices that will leverage the availability of state-of-the-art equipment and system tools. Future work will include:

- Vulnerability assessment of the various control devices and the systems in which they are integrated;
- Development of security best practices for critical infrastructures;
- Development of a secure operating system that will control these devices;
- Collection and analysis of forensic data from critical infrastructure networks; and
- Development of advanced vulnerability assessment tools for critical infrastructures.

## 6 Acknowledgements

This paper is based upon a project partly supported by the National Science Foundation under grant award OCI-0959687. Opinions expressed are those of the authors and not necessarily of the Foundation.

## 7 References

- [1] United States Government Accountability Office (GAO), "Critical Infrastructure Protection DHS Needs to Fully Address Lessons Learned from Its First Cyber Storm Exercise." Report GAO-08-825, September 2008.
- [2] President's Commission on Critical Infrastructure Protection. "Critical Foundations— Protecting America's Infrastructures"  
Website: <http://www.fas.org/sgp/library/pccip.pdf>.  
Access date: January 25, 2011.

[3] President's Critical Infrastructure Protection Board and the Department of Energy "21 Steps to Improve Cyber Security of SCADA Networks."

Website: [http://www.oe.netl.doe.gov/docs/prepare/21step\\_sbooklet.pdf](http://www.oe.netl.doe.gov/docs/prepare/21step_sbooklet.pdf). Access date: December 18, 2010.

[4] Davis, C.M., Tate, J.E., Okhravi, H., Grier, C., Overbye, T. J. and Nicol, D., "SCADA Cyber Security Testbed Development," *Proceedings of the 38th North American Power Symposium (NAPS 2006)*, Carbondale, IL, September 2006, pp. 483-488.

[5] Kim, G.S. and Lee, H.H. "A Study on IEC 61850 Base Communication for Intelligent Electronic Devices," *Proceedings of the IEEE 9<sup>th</sup> Russian-Korean International Symposium on Science and Technology*, Vol 1, Novosibirsk, Russia, 2005, pp. 765-770.

[6] Giani, A., Karsai, G., Roosta, T., Shah, A., Sinopoli, B., and Wiley, J., "A Testbed for Secure and Robust SCADA Systems," *ACM SIGBED Review*, Vol. 5, Issue 2 (July, 2008). Special Issue on the 14<sup>th</sup> IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '06) WIP Session.

[7] Patel, M., Cole, G. R., Pryor, T.L., and Wilmot, N.A., "Development of a Novel SCADA System for Laboratory Testing," *ISA Transactions* 43 (2004). Pp. 477-490.

[8] National SCADA Test Bed Fact Sheet. Idaho National Laboratory. Website: <http://www.inl.gov/scada/factsheets/d/nstb.pdf>. Accessed date: March 08, 2011.

# Data Center Physical Security Ontology for Automated Evaluation

Nanta Janpitak and Chanboon Sathitwiriawong

Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang,  
Bangkok 10520, Thailand

**Abstract** - Nowadays, most business operations are supported by IT systems. Therefore, their availability is critical to keep business running smoothly and continuously. In order to provide high quality IT services, a well-managed data center is required to house computer servers, storage systems, network devices, and their associated components. Downtime of the data center can be costly resulting in production and business losses so that the high availability requirement of the data center is needed. Apart from availability, the data center also requires a dependable and secure computing including such attributes as confidentiality, reliability, safety, integrity, maintainability, etc. This paper introduces an ontology-based framework for data center physical security by gathering and mapping the requirement from well-known information security standards such as COBIT, ISO/IEC 27002, and ITIL. In order to fulfill the safety requirement of the data center, this ontology-based framework is also designed to be applicable with National Fire Protection Association (NFPA) code and standard for protecting all data center occupants and for limiting data center property loss from fire. The completion of this ontology will be used for the knowledge sharing and also as an input for data center physical security evaluation tool.

**Keywords:** Ontology; Data Center; Dependable Computing; Information Security; Automated Evaluation

## 1 Introduction

Data Center is a facility used for housing a large amount of computer and communications equipment maintained by an organization for the purpose of handling the data necessary for its operations [1]. Since data center contains many sensitive organization's data, the access to these data by authorized person is one of the mandatory feature of data center. The access control of data can be done physically and logically. This paper focuses on physical access control by referring to section DS12-Manage the Physical Environment from the mapping of CoBiT 4.1, ITIL V3 and ISO/IEC 27002 [2]. This publication is the new ITGI/OGC guide intended to help companies achieve maximum governance and value in a down economy.

Data center always requires a non-stopped service or 24x7 availability. The availability of data center mainly requires the

protection of computer equipment and personnel. There are some potential hazards which may occur and impact the availability of data center. Fire is the most potential hazard which can create severe effects on data center. Fire can occur in a data center by mechanical failure, intentional arson, or natural causes. This paper focuses on how to deal with fire for protection of computer equipment and personnel by designing the data center to comply with National Fire Protection Association (NFPA) code and standard [3].

Most large organizations have defined their information security policy to protect their information asset by interpreting multiple requirements such as laws, regulations, well-known standards, and some other requirements. IT personnel including IT managers are well aware that information security policy is important to follow but they do not take much effort to understand and remember what the rules or policies said. They always leave this responsibility to Information Security Expert, which is a rare personnel in each organization. In order to evaluate the policy compliance, the Information Security Expert has to do the evaluation manually using their information security expertise. The manual evaluation is always time consuming, complex and requires expert knowledge. Once the Information Security Expert leaves the company, other personnel cannot evaluate their policy compliance because of the lack of information security knowledge.

Another major problem in managing data center is that most facilities in the data center are normally installed and supported by other departments rather than IT department. For instance, Production Engineering department is responsible for the planning and design of overall facilities, Maintenance department is responsible for preventive maintenance, and Safety department is responsible for supporting and monitoring the fire protection system. This makes more complexity to data center facilities management. IT department must provides a clear communication to those departments to make sure that the data center related policies are met, so that IT personnel should have a good understanding and knowledge in data center policy requirement.

As mentioned above, various approaches to organize the information security knowledge and automate the evaluation process of policy compliance have been proposed. Ontology is one of the tools which many researchers have recently

proposed to support the knowledge sharing and enhance the automated evaluation process. The overview of ontology will be provided in section 3.

The overall proposal of this paper is the ontology-based framework for data center physical security. This ontology is designed by gathering many sources of requirement in order to develop a single source of knowledge and prepare for a future automated evaluation process.

The rest of this paper is organized as follows. Section 2 explains the overview of data center and discusses some related works. Section 3 explains the overview of ontology and discusses some information security related works. Section 4 describes the proposed ontology-based framework for data center physical security. Finally in section 5 summarization and conclusions of this paper are provided.

## 2 Overview of data center and related works

A data center is a facility used to house computer servers, storage systems, network devices, and their associated components. Downtime of data center can be costly resulting in production and business losses. In order to reduce business interruptions, an effective management with a comprehensive design of data center is required.

As data center becomes more and more central in the present age of internet communication, both research and operations communities have begun to explore how to better design and manage them. There are some materials providing guideline for data center design such as Sun Microsystems provides "Enterprise Data Center Design and Methodology" [4], Cisco Systems provides "Data Center Fundamentals" [5]. Those materials provide a comprehensive design guideline to cover the different areas of data center requirements such as cable management, network infrastructure, environmental controls, power management, physical security, etc. The data center ontology-based framework designed in this paper is based on the guideline in "Enterprise Data Center Design and Methodology" from Sun Microsystems.

Thomas [6] discussed an idea that computer security should be improved through environmental controls. An environment which is constantly varying will produce unreliable equipment operation. Humidity has to be controlled as well as temperature. For instance, the temperature in data center should not be adjusted by human comfort but rather machine, the relative humidity should not exceed 80 percent, the maximum number of people allowed in a data center at any one time should be determined, etc. Working space is also an important element when designing the data center. Since the computing equipment has a very heavy load, so floor selection process should be carefully done. As Thomas's main point, the environmental controls have been put an attention as one main component in the proposed ontology which will be discussed in section 4.

Robert [7] presented an overview of some technical and managerial of protecting the data center resources including personnel from any accidental damage. This paper focuses on the engineering management aspects of 6 major areas of concern regarding operational data security: personnel, facilities, computer hardware, computer software, communications, and procedures. In addition to the many important design considerations such as temperature, humidity, cooling water and fire fighting system, the interference of electro-magnetic to computer hardware is also examined in this paper.

In order to protect the information asset, the information security policy is defined by interpreting multiple requirements such as laws and regulations. To ensure that the information security policy is followed, the evaluation process is required. The policies or any related requirements regarding to the data center physical security are considered as a non-technical requirement which is hard to be interpreted and transformed to a machine-readable form. This makes an evaluation or validation process hard to be performed. In order to enable the automated process, an ontology technology has been selected to transform the non-technical requirement into machine-readable form. The overview of ontology will be described in the next section.

## 3 Overview of ontology and information security related works

Ontology is an explicit specification of a conceptualization [8]. A conceptualization is the objects, concepts, and other entities that are assumed to exist in some areas of interest and the relationships that hold among them. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purposes. Ontology represents knowledge in a formal and structured form as well as provides a better communication, reusability and organization of knowledge and a better computational inference.

From the previous section, information security needs a better knowledge sharing method for better communication. It also requires automating any related processes such as evaluation and monitoring. Ontology has been studied and proposed by many researchers to cover those requirements. Since the area of interest and the relationships that hold among them is considered as a domain, so that from now, the information security area will be called as information security domain in this paper.

A study of Carlos et al. [9] is useful to shorten the review of previous works in information security domain. They used OntoMetric [10] to compare various security ontologies proposed by many researchers. Finally, they have concluded that the existing ontologies in this domain are not prepared for being reused and extended. They suggested that the community should put efforts to join and improve the developed ontologies. Besides combining different terms from

different creators looks impossible despite being a domain expert.

Stefan et al. [11, 12] proposed a security ontology based on the security relationship model described in the National Institute of Standards and Technology Special Publication 800-12 [15]. Their security ontology was also developed based on the basic concepts and taxonomy of dependable and secure computing [16] which will be used and discussed later in the proposed ontology section. The core concepts of their security ontology were grouped in three sub-ontologies: security sub-ontology, enterprise sub-ontology, and location sub-ontology. Security sub-ontology consists of attribute, control, threat, vulnerability, and rating, derived from well established information security standard. Enterprise sub-ontology consists of asset, person, and organization. Location sub-ontology only stores a list of locations. Finally they have given the relations between these concepts such as each *threat* has been connected to *asset* concepts by the *threatens* relation, organization concept has been connected to the *assets* by the *ownedBy* relation. In [11] they presented a tool called "SecOntManager" by using their security ontology to simulate threats. From the simulation example, it shows the impact of fire on the infrastructure, what countermeasures exist and how the outage costs for each simulation of countermeasure.

Stefan et al. also proposed other ontologies related to information security domain such as ontology-based framework to improve the preparation of ISO/IEC 27001 audits [13] by mapping with the security ontology which they had earlier proposed. A Common Criteria (CC) ontology [14] comprising the entire CC domain with special focus on security assurance requirements is relevant for the evaluation.

Ju and Minzhe proposed an ontology for vulnerability management called OVM [17]. The top level concepts in this ontology are Vulnerability, IT\_Product, Attacker, Attack, Consequence, and Countermeasure. At the earlier steps, the OVM retrieves the common vulnerabilities from the National Vulnerability Database (NVD) which is the US government repository of standard-based vulnerability data. The OVM then links the *Vulnerability* concept to *IT\_Product* concept by *hasAffectedProduct* and *hasVulnerability* relations. By retrieving vulnerability data from OVM, the information can then be served as the knowledge base for vulnerability management.

From many information security ontologies that we had reviewed, we found that the ontologies developed for a full information security domain are not easy to be used in the future. We support an idea that the information security domain should be split into subdomains and each subdomain should have a connection point to link each other. Then a full information security domain can be created by linking all subdomains. This makes it easier to be implemented in the next steps. The ontology-based framework for data center physical security is one of the information security subdomains that will be explained in the next section.

## 4 Ontology-based framework for data center physical security

To ensure that the protection of data center is effective, the IT department should have an effective compliance management. To support the compliance management, the ontology-based framework for data center physical security has been developed in details as follows subsections.

### 4.1 Overview of data center physical security framework

In order to support the compliance management, the ontology-based framework for data center physical security has been developed the same as the data center policy by consolidating the requirements from various well-known standards for computer security, fire protection and environmental control as depicted in figure 1.

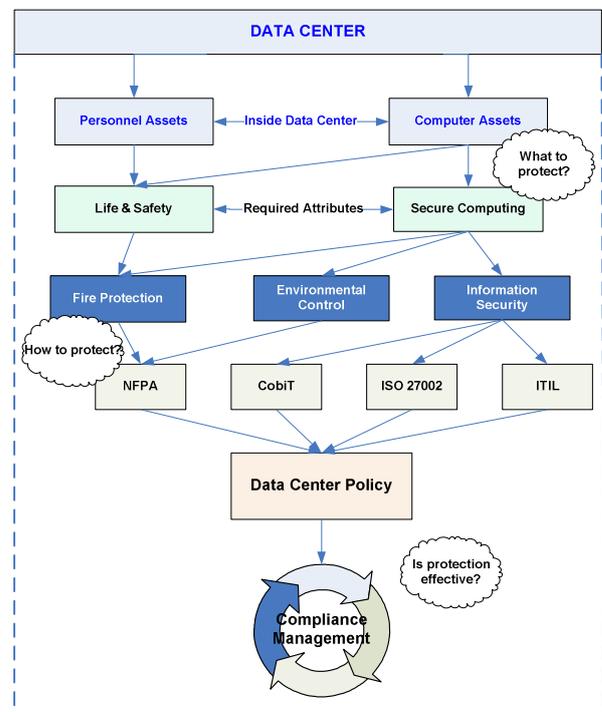


Figure 1. Data Center Physical Security Framework

### 4.2 Overview of ontology development methodology

To increase the ontology efficiency and ensure the consistency of ontology structure, during development we followed the guidelines from many sources. First we studied how to build the ontology by using a guideline from [18]. The guide was built using Protégé ontology editor [19] which is the same tool that we have used for our ontology development. To develop ontology by using Protégé we followed the guideline from [20]. There is no single correct ontology-design methodology [18] so that we studied a few ontology

development methodologies and finally we decided to follow a recently defined methodology from [21] incorporating with the guide in [18]. This ontology development methodology covers the steps from the initiation stage to the maintenance stage of ontology. The methodology consists of three main subprocesses: specification, concretization and implementation. Each subprocess consists of a set of activities that will be described along the ontology development in the next subsection.

### 4.3 Ontology for data center physical security

The ontology for data center physical security has been developed by following the methodology in [21] as the following processes.

#### 1. Specification subprocess (Equivalent to step 1 in [18])

**1.1 Activity 1: Describing the domain.** The ontology for data center physical security is a subdomain of the information security domain. This ontology is the consolidation of requirements from various well-known standards for computer security, fire protection and environmental control to protect information asset and ensure that data center can provide a continuous service with a minimum downtime. The completion of this ontology will be used for the knowledge sharing and also as an input for data center physical security evaluation tool.

**1.2 Activity 2: Elaborating motivating scenarios and competency questions.** Due to page constraint, the complete scenario and competency questions will not be presented in this paper. The example of competency question is "How to prevent fire in data center?" which will be used as example in the validation activity (subprocess 3.3).

**1.3 Activity 3: Determining the ontology goal and scope.** This ontology goal is to represent the set of data center physical security requirements as reusable knowledge for preparing either manual or automated evaluation process. This ontology is limit to the physical security which does not include the logical security. This ontology covers only fire hazard which is the most potential occurrence at all parts of the world. The other natural disasters such as flood or storm can be included in the future.

#### 2. Concretization subprocess

**2.1 Activity 1: Defining classes and class hierarchy (Equivalent to step 3-4 in [18]).** In this task, a list of terms that represent the most important entities in data center physical security domain has been enumerated as classes. The list of important classes and subclasses is shown in table 1. Definitions of important classes are provided after table.

TABLE 1. Key Item List as Class and Subclass

Class	Subclass
	DataCenter
	Safeguard
DataCenterDesignGuideline	Physical_LogicalSecurity
	AvoidingHazards
	HVACandEnvironmentalControls
Reference	ISO2700x
	COBIT
	ITIL
	NFPA
Facility	PhysicalAccessControl
	FireDetectionSystem
	FireSuppressionSystem
	PowerRedundancy
	HVAC
Threat	Hazard:Fire
	PowerLoss
	UnstableEnvironment
	UnauthorizedAccess

- The class "DataCenter" is defined as the highest level class in this domain.
  - The class "Safeguard" is defined to contain the countermeasure that use to deal with each threats which impact to the availability of data center.
  - The class "DataCenterDesignGuideline" is defined to contain the contents of guideline retrieved from [4] that will be referred by each safeguard.
  - The class "Reference" is defined to contain the contents of requirements from various well-known standards. The reference is divided into 4 subclasses as "ISO2700x", "COBIT", "ITIL", and "NFPA".
  - The class "Facility" is defined to contain the facilities that have to be used in accordance with the guidelines or references. The facility is divided into a few main subclasses such as "PhysicalAccessControl", "PowerRedundancy", and "FireSuppressionSystem".
- 2.2 Activity 2: Identifying class relations, attributes and properties (Equivalent to step 5 in [18]).** Only the classes will not provide enough information. In this task, the main relations, attributes and properties were created. The example of class relations is shown in table 2.
- 2.3 Activity 3: Representing rules and restrictions (Equivalent to step 6 in [18]).** This task is to analyze the restrictions represented in the class relations, attributes and properties. On the other hand, in general usage a restriction is a specific type of rule that sets a finite boundary defined for a type of process or function. The example of class attributes, properties, rules and restriction is shown in table 3. Then, the classes, class hierarchy with relations have been captured in a graphical diagram to represent the

linkage and relation between each component as shown in figure 2.

2.4 *Activity 4: Representing individuals (Equivalent to step 7 in [18]).* This task is to define individual instances of each class. The instances of data center physical security were created as shown in table 4.

### 3. The implementation subprocess

3.1 *Activity 1: Creating a computational ontology.* The goal of this activity is to convert the ontology which was designed in the prior subprocesses into a formalized representation interpretable by a machine, using an appropriate language with formal semantics. There are different languages to be used for this task. The most relevant ones are RDF (Resource Description Framework) and OWL (Web Ontology Language). In order to carry out this activity, the Protégé ontology editor [19] which is the most popular ontology development tool has been used. The ontology was built in OWL by using the Protégé as shown in figure 3.

3.2 *Activity 2: Verifying the ontology.* The goal of verification process is to avoid future propagation of errors. To compare with ontology which was designed in the prior subprocesses, the graphical view of class hierarchies were generated by using OWLViz plug-ins. The consistency checking was done by using a Reasoner plug-ins.

*Activity 3: Validating the ontology.* In order to validate the ontology, it is necessary to verify whether the ontology can answer the competency questions. We have to do some semantic queries by using the semantic web query language SPARQL [22] and Jena API [23]. Hereunder is an example result of SPARQL query to answer the competency question "How to prevent fire in data center?"

```

SELECT ?x WHERE { ?x rdf:type
<#Guideline_FirePrevention> }
Result:
...
Guideline_FP_NoSmoking
Guideline_FP_NoCombustibleMaterials
...
SELECT ?x WHERE
{ <#Guideline_FP_NoSmoking>
rdfs:comment ?x }
Result:
Smoking should never be allowed in the data center.
Signs should be posted at entryways and inside.

```

The example query result shows how this ontology answers the competency questions. In the future steps, the query result will be used in semantic web technology for user friendly mode.

TABLE 2. An excerpt of the relation table of the data center physical security

Class Name	Relation	Class Name	Inverse Relation
DataCenter	required	SecurityAttribute	requiredBy
SecurityAttribute	impactedBy	Threat	impactTo
Fire	detectedBy	FireDetection System	toDetect
	Suppressed By	FireSuppression System	toSuppress
PowerLoss	preventedBy	Power Redundancy	toPrevent
Unstable Environment	controlledBy	HVAC	toControl
Unauthorized Access	preventedBy	PhysicalAccess Control	toPrevent
FireDetection System	followedTo	DataCenter Guideline	followedBy
DataCenter Guideline	referredTo	Reference	referredBy

TABLE 3. An excerpt of the class attributes and properties of the data center physical security

Class	Property	Type	Restrictions
DataCenter	has Component	Instant	class{RoomComponent}
	required	Instant	class{SecurityAttribute}
	threatenBy	Instant	class{Hazard:Fire}
Hazard:Fire	detectedBy	Instant	class{Guideline_FireDetection}
	preventedBy	Instant	class{Guideline_FirePrevention}
	Suppressed By	Instant	class{Guideline_FireSuppression}
Unstable Environment	controlledBy	Instant	classes={HVACandEnvironmentalControls}
Unauthorized Access	preventedBy	Instant	classes={PhysicalAccessControl}

TABLE 4. An excerpt of the class attributes and properties of the data center physical security

Class	Instance Name	Property Name	Property Value
DataCenter	PirmaryDC	required	Availability
		threatenBy	Threat_Fire
		hasComponent	DCRoof
		hasComponent	EgressDoor
		hasComponent	DoorSideWall
Room Component :Door	EgressDoor	madeBy	Steal
		hasFireRating	1
		isSwingOut	true
Room Component :Wall	DoorSideWall	equippedTo	Facility_Fire Extinguisher_Co2
		hasFireRating	1
Facility:FireSuppressionSystem	Facility_FireExtinguisher_Water	isMandatory	true
	Facility_FireExtinguisher_CO2	isMandatory	true

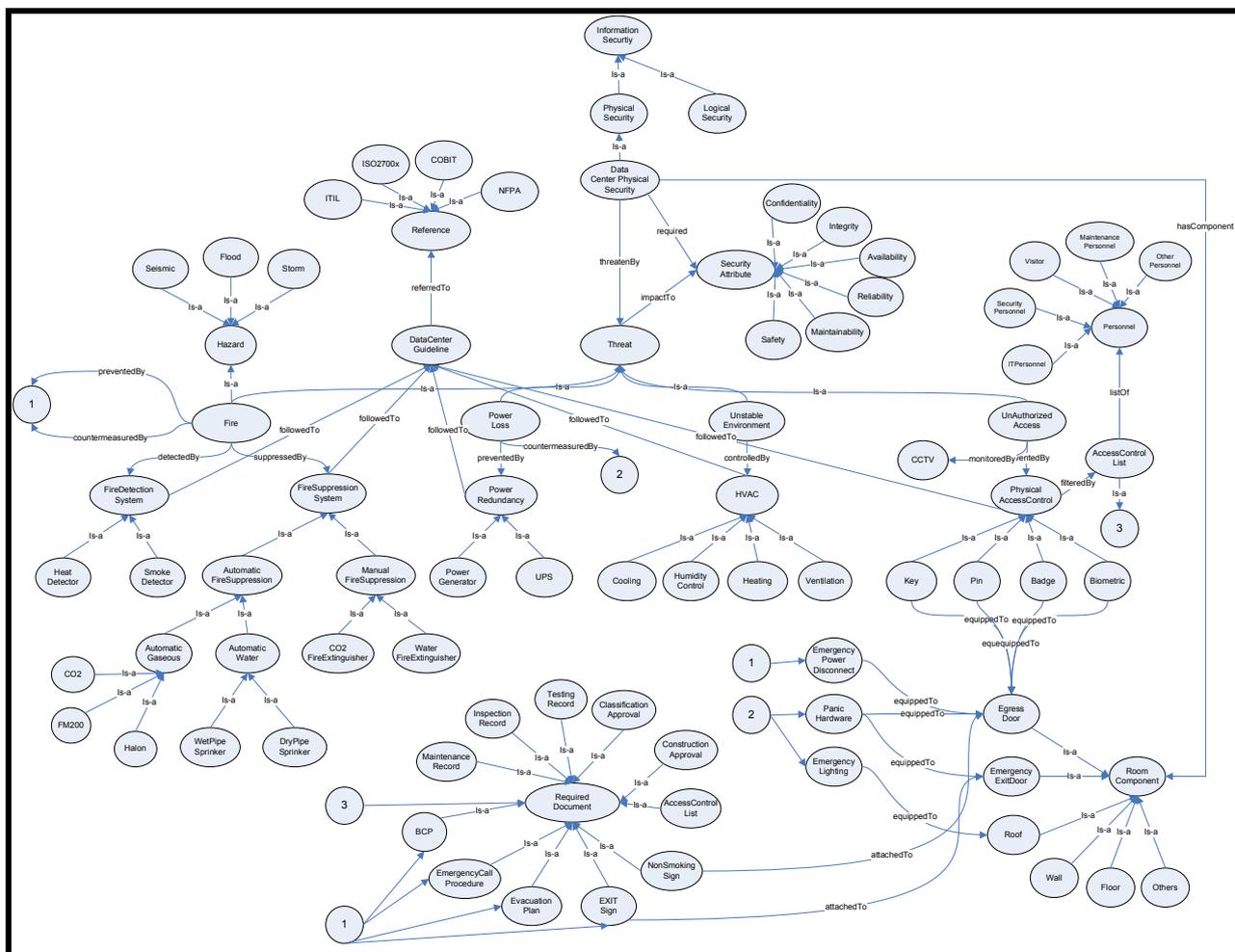


Figure 2. Components Linkage Diagram

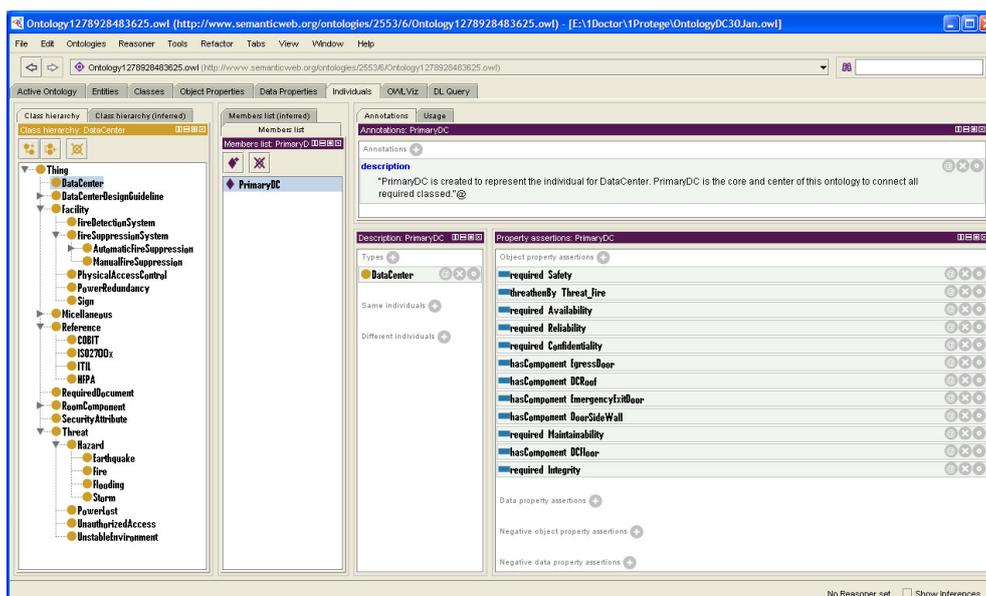


Figure 3. Protégé logical view window

## 5 Conclusions

This paper presents an ontology developed for information security knowledge sharing by focusing on data center physical security. The data center physical security ontology can be used for an automated evaluation in the future to enhance the automated compliance process. This ontology also provides the connection point to the information security domain. Since the information systems can be accessed by either physical or logical so that the information security should be split into physical security and logical security. Then the data center physical security is linked to the main information security domain under physical security section. Apart from data center, there are other physical entities such as a client, a telecommunication asset, etc. The controls of those entities can be developed as ontology and linked into information security domain under physical security section. The ontology proposed and developed that is presented in this paper would be continuing improved and used for our work in progress (Data Center Physical Security Evaluation Tool).

## 6 References

- [1] Glossary of MMC Terminology. Available: [http://msdn.microsoft.com/en-us/library/bb246417\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb246417(VS.85).aspx).
- [2] ITGI/OGC. "Aligning COBIT 4.1, ITIL V3 and ISO/IEC 27002 for business benefit". A management briefing from ITGI and OGC. Available: <http://www.isaca.org>.
- [3] National Electric Code, National Fire Protection Association International, Boston, Mass., U.S.A.
- [4] R. Snevely. "Enterprise data center design and methodology". Palo Alto, California: Sun Microsystems Press, A Prentice Hall Title, 2002.
- [5] M.Portolani. "Data center fundamentals". Indianapolis, Indiana: Cisco Press, 2004.
- [6] T. C. Richards. "Improving computer security through environmental controls"; Security Audit and Control Review, Vol. 1, No. 3, Fall 1982, pp. 18-24.
- [7] R. J. Wilk. "Engineering management considerations in data center security"; Proceedings of the 4th annual symposium on SIGCOSIM: management and evaluation of computer technology, 1973, pp. 11-22.
- [8] T. R. Gruber. "Towards principles for the design of ontologies used for knowledge sharing"; International Journal of Human-Computer Studies, Vol.43, 1995, pp. 907-928.
- [9] C.Blanco et al. "A systematic review and comparison of security ontologies"; International Conference on Availability, Reliability and Security (ARES). Barcelona, 2008, pp. 813-820.
- [10] A. Lozano-Tello, A. Gómez-Pérez. "ONTOMETRIC: A method to choose the appropriate ontology"; Journal of Database Management. Special Issue on Ontological analysis, Evaluation, and Engineering of Business Systems Analysis Methods, Vol.15, 2004, pp.1-18.
- [11] A. Ekelhart, S. Fenz, M. Klemen, and E. Weippl. "Security ontology: Simulating threats to corporate assets"; In A. Bagchi and V. Atluri, editors, Information Systems Security, volume 4332 of Lecture Notes in Computer Science, Springer, 2006, pp. 249–259.
- [12] S. Fenz, A. Ekelhart. "Formalizing information security knowledge"; In 4th International Symposium on Information, Computer, and Communications Security (ASIACCS '09), 2009, pp. 183-194.
- [13] S. Fenz, G.Goluch, A. Ekelhart, B. Riedl, and E. Weippl. "Information security fortification by ontological mapping of the ISO/IEC 27001 standard"; In 13th IEEE International Symposium on Pacific Rim, 2007, pp. 381-388.
- [14] A. Ekelhart, S. Fenz, G. Goluch, and E.Weippl. "Ontological mapping of common criteria's security assurance requirements"; In 22nd IFIP TC-11 International Information Security Conference (IFIPSEC'07), 2007, pp. 85-95.
- [15] NIST. An Introduction to Computer Security – The NIST Handbook. Technical report, NIST (National Institute of Standards and Technology), October 1995. Special Publication 800-12.
- [16] A. Avizienis, J.-C. Laprie, B. Randell and C. Landwehr. "Basic concepts and taxonomy of dependable and secure computing"; IEEE Trans. Dependable and Secure Computing, vol. 1, no. 1, 2004, pp. 11-33.
- [17] J. A. Wang and M. Guo. "OVM: an ontology for vulnerability management"; Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, 2009, pp. 1-4.
- [18] N. F. Noy and D. L. McGuinness. "Ontology development 101: A guide to creating your first ontology"; Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001.
- [19] Stanford Center for Biomedical Informatics Research. "The Protégé ontology editor and knowledge acquisition system". Available: <http://protege.stanford.edu>.
- [20] M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroe. "A practical guide to building OWL ontologies using the Protege-OWL plugin and CO-ODE tools edition 1.2"; University of Manchester, 2009.
- [21] G. Brusa, M. L. Caliusco, and O. Chiotti. "Towards ontological engineering: a process for building a domain ontology from scratch in public administration"; Expert Systems: The Journal of Knowledge Engineering, Vol. 25, Issue 5, 2008, pp. 484–503.
- [22] SPARQL Query Language for RDF. Available: <http://www.w3.org/TR/rdf-sparql-query/>.
- [23] Jena – A Semantic Web Framework for Java. Available: <http://jena.sourceforge.net/index.html>

# A Strategy for Information Security: TOGAF

L. Ertaul<sup>1</sup>, A. Movasseghi<sup>2</sup>, and S. Kumar<sup>2</sup>

<sup>1</sup>Math & Computer Science, CSU East Bay, Hayward, CA, USA

<sup>2</sup>Math & Computer Science, CSU East Bay, Hayward, CA, USA

**Abstract** - *In the old culture of security concept, information security was based on securing the ownership of information. This kind of protection accomplished through monitoring and securing the physical network devices and application software. In the new paradigm, which is entirely based on distributed network architecture and relationship within and across different enterprises that each uses a combination of non-proprietary and also proprietary information, the information and infrastructure access requires far beyond only the physical perimeter. In this new paradigm the technology team, policy makers and the legal advisors require a dynamic inter-action. Based on the current available TOGAF (The Open Group Architecture Framework) security information, this paper proposes a framework to provide information security at the enterprise-level which reflects recent realities of information and access sharing in enterprise networks...*

**Keywords:** Enterprise Security Planning, Information Security, TOGAF

## 1 Introduction

Today's globally distributed network systems require a management team that manages the viewpoints of all the stakeholders in the business, to collect objectives from each department and provide a solution that covers all their security requirements. [1]

The type of stakeholders that enterprise security architecture and also information technology team need to work as a single team include risk-management, corporate legal console, security auditors and different business managers.[1], [2]

This paper proposes a process-based, dynamic, information centric security framework which through different functional boundaries will help in resolving different viewpoints, and also provide a methodology for security policy both within and across the enterprise networks perimeter using TOGAF (The Open Group Architecture Framework).

## 2 The Security Problem

One reason to analyze the current security effectiveness of information security policies is due to fundamental changes in the basic assumptions which those policies are based upon such as, by securing the physical perimeters of the information now the security of that property is achieved.

The history of securing the physical devices (hardware), applications (software) and the storage media is dating back to 1983 (Department of Defense Trusted Computer system Evaluation Criteria).[3] Enterprises for the past twenty years invested part of their budget for securing computing equipments, operating systems, communication channels, and storage properties. These properties are no longer requires security and in fact the computing platform is so much available to public that no longer is consider as a property.[1], [2]

What once were consider highly secure resources only thirty years ago such as storage, CPUs, network connections, and memory, are available today virtually anywhere, anytime. Yet the way business managers, decision makers, technical members, and others have learned and thought to handle the security of their data is by protecting their computing platforms and resources. [1], [2]

In today's globally distributed information networks constantly requires that all the sectors, private/public and other consumers to assume a new type of risks. At the same token, the people are responsible to manage these new types of risks seems not to understand and familiar with these issues adequately. [1],[2]

Industry groups, policy makers, interest groups, and regulators are working together on developing a new regulations and standards that can help the enterprises to control and manage the security of their information systems.

## 3 How Information Security Achieved?

The term "security" is usually used in two scenarios. The first can be thought as set of functions and features to protect integrity and confidentiality and also the availability. This case of security is well known and developed by the users and vendors alike, which resulted in broad selection of tools, application and standards that available for consumers. The TOGAF security group currently does not make a major contribution that differs from what other standards are already providing. [1],[2],[4],[5]

In second scenario, the security is thought as a property of the information systems such as usability, manageability and quality. In this term, the security is assumed as a non functional property which makes this property more difficult to measure and discusses clearly.[1],[2]

This paper explores this non functional part of security, to give some definition and also provide some strategy which can be used in conjunction with TOGAF security standard.

#### 4 Enterprise Information Security Architecture

The architecture of enterprise information system consists of managing the viewpoints between consumer, business, and public sectors interests and provides a common resolution as shown in Figure 1. These interests include:

- ❖ Consumer expressed their interest to control the use their information's by both public and business sectors. This can be achieved by government help through regulations and legislation to prevent the misuse of those information
- ❖ Public, such as public safety, infrastructure protection from cyber attacks, national security, financial risk management and consumer protection.
- ❖ Business interests to provide environment to achieve the highest financial outcome for their shareholders and balance the risk and rewards. To provide higher return by minimizing the risk. The information security in business sector is viewed as risk management issue which can be resolved in favor of business goals.

The information security architecture objective is to mitigate the tensions between three sectors by developing non functional security components. [1][2][4][5]

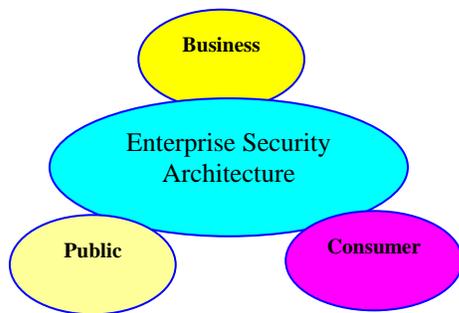


Figure 1 Enterprise Information Security Architecture

#### 5 Closed perimeter vs. Centralize Information

In past the information security used to control the resources such as storage, computing and communication channels security by defining a closed perimeter by controlling:[1][2]

- ❖ Information access

- ❖ In/out traffic
- ❖ Time of access
- ❖ Entry port and type of service
- ❖ Resources to access computing platforms and speed
- ❖ Network connectivity access and bandwidth
- ❖ Storage media

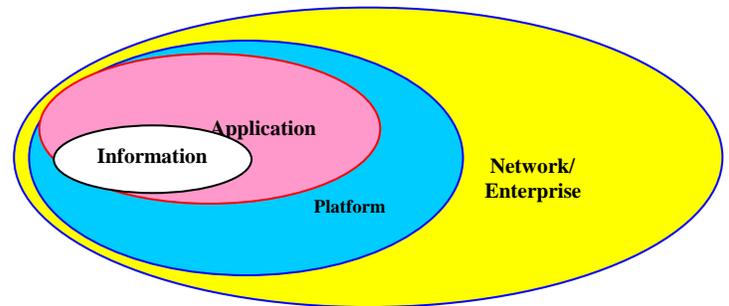


Figure 2 Layered Perimeter based Security Model

In today's networks, due to change of paradigm with vast availability of resources, the new paradigm requires support of supply chains, outsourcing, and other type of business needs. Current network perimeter security, Fig.2, concept have been shrinking and even in some cases disappearing that they not have a great control over major portion of communication and traffic passes through each boundary such as web, email, VOIP, and some type of encrypted data (SMTP, VPN). In new model, perimeter still exists, but changing to type of perimeters with no specific shape. The information can traverse between traditional boundaries with a shape that is not very clear to inexperienced security architecture.



Figure 3 Centralized Information Security Architecture

The question comes to mind that "How a security architect can establish a security quality for something as intangible as information by using non functional security model?"

At this point we have entered a new model and paradigm called "Centralized information" security as shown in Fig 3.

This new model which is shapeless with not clear perimeter shows that the traditional boundary such as platform,

application and enterprise perimeters does not exist any longer. Instead, this shapeless boundary surrounds the information from one entity to another.

## 6 Information Control

In our current model of shapeless boundaries, control of computing assets such as information which is intangible can be functionally equivalent to having “ownership” of physical assets in physical world. The new security definition becomes a question of being able to maintain an equivalency of “ownership” in traditional model through control over the computing assets such as information wherever in enterprise/network they reside. Based on that, following are the major principles of control that emerge:[1][2][4][5]

- 1) The assets can only be controlled within specific boundary, once the asset/information traverse outside the controlled environment, the control of the information by the owner has lost.
- 2) Remote control is hard; it's hard enough to control information within a managed application such as firewall perimeter. Controlling the information beyond enterprise boundaries is very hard to manage and guarantee. One way to resolve this issue to enable the global enterprise to share its sensitive information with suppliers, business partners, customers and outsource providers by an acceptable level of risk. This can be establish through:

- ❖ NDA
- ❖ Information sharing legal agreements
- ❖ Control expectations through standards
- ❖ Control practices for technical, physical entities that can be verified

## 7 Information Control within Virtual Boundary

TOGAF [4][5] presented a framework for security control for securing the information within enterprise virtual perimeter. This framework consists of:

- Define the requested action and how to response
- Monitor the incoming actions
- Force the action to be taken

as shown in Fig 4.

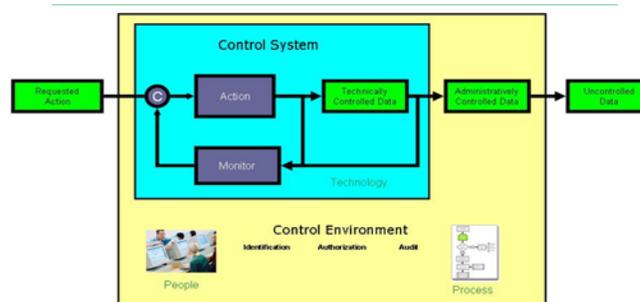


Figure 4 Virtual Frameworks for Security Control

## 8 Security Policy Compliance

To be able to control the information beyond the traditional boundaries, it's appropriate to discuss the role of compliance with security standards and policy such as TOGAF recommendations.

The security compliance model works in away that the enterprise security policy complies with standard and external policies such as TOGAF security architect framework as shown in Fig 5.

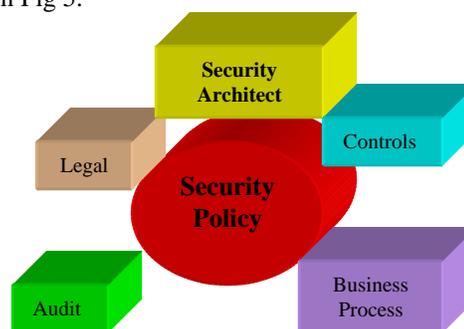


Figure 5 Security Policy Framework

Following are the major actions required for deploying a successful security police in enterprise environment:

- Define Compliance Goal – By answering “What standard security policy needs to be compliance?”. TOGAF security framework can be used as the starting point plus other external requirements such as SLA (Service Level Agreements), customer expectations and corporate policies.
- Evaluate and Assess above Objectives – For the above objectives, the legal team must assess these requirements and check to see which can be apply to the enterprise and the business.
- Create Enterprise Policy – Not legal objectives in nature must be reviewed with business management and process people to create a policy that is consistent with the above objectives.
- External requirements compliance evaluation – Evaluate the security policy through legal team. This

team provides a document and the monitor that followed across enterprise organization.

- The Security Architect must implement a framework such as TOGAF that compliant to above security policy.
- Security Policy Audit Function – To assess whether the enterprise security process and technology comply with the enterprise generated security policy and also assurance that the management has been carried out these policies.

Information control beyond the enterprise control environment in general requires sort of formal agreements such as Service Level Agreement (SLA). To extend the control beyond the original perimeters, this control consists of:

- Generate formal agreement to control flow of information across involved enterprises perimeters
- Provide an SLA and a way to audit compliance

SLA is a legal and business management process. Audit and verification of compliance to that SLA can be achieved through standard audit framework known as SAS-70 (statement on Auditing Standards-70) report.[6]

SAS 70 audit is a highly specialized audit conducted in accordance with Statement on Auditing Standards (SAS) No. 70, Service Organizations. The product of a SAS 70 audit is presented by the auditor to the service organization in the form of a Service Auditors Report. The SAS 70 Service Auditors Report can be either a Type I or Type II. Advanced provides both types of services, as well as pre-SAS 70 assessments.

This report can help the enterprise to avoid conducting regular audit of critical systems; the enterprise information owner can request SAS-70 report from the vendor. From this report the enterprise security team can assess the degree of vendor compliance, which helps the enterprise security architect to evaluate if the vendor meets the information's owner expectations for control and manage its information.

## 9 Conclusions

Whether called “Centralized information security”, “virtual-boundary” or “control of information devices”, the information security team must consider impacting factors such as policy, technology, and economic for information security. The team also required to represent all the stakeholders “views” within the process. Security as a combination effort of people, technology, and processes in enterprise architecture framework is controlling the information security across the organization perimeters. Based on these requirements, in today's information security strategies, corporate policy/legal, and the audit are the major

stakeholders, which force the architecture community to articulate these stakeholders within the community.

The security architect can mitigate and facilitate the different view points of the stakeholders through providing a dialog between them.

TOGAF open group security division as a leading organization contributes toward providing solutions for security information. In this role the open group team facilities and encourage development of open standards, tool, and method to improve current enterprise security information essential practices and methodology. Some of the highly relevant components to support the security strategy are listed as:

- ❖ Even though auditing and monitoring are the key components of the security, but not many standards in these areas. What and what not you should monitor? This can be a joint project with legal team to clarify what should be monitor.
- ❖ Integrate with TOGAF monitoring and required corrective actions as a development of additional views on control.
- ❖ For compliance, develop additional views.
- ❖ Form a information security perspective with the help of legal, technical and audit members.

## 10 References

- [1] B. G. Raggad, “Information Security Management: Concepts and Practice”, CRC Press, 2010.
- [2] H. H. Carr, C. A. Snyder, B. N. Bailey, “The management of Network Security: Technology, Design, and Management Control”, Prentice Hall, 2010.
- [3] NIST Document DoD 85, “Department of Defense Trusted Computer system Evaluation Criteria” <http://csrc.nist.gov/publications/history/dod85.pdf>
- [4] The Open Group web site: TOGAF <http://www.opengroup.org/togaf/>
- [5]: The Open Group Security Forum: <http://www.opengroup.org/security/>
- [6]: The Systems on Auditing Standards (SAS-70) report [http://sas70.com/sas70\\_overview.html](http://sas70.com/sas70_overview.html)

# Enterprise Security Planning with TOGAF-9

L. Ertaul<sup>1</sup>, A. Movasseghi<sup>2</sup>, and S. Kumar<sup>2</sup>

<sup>1</sup>Math & Computer Science, CSU East Bay, Hayward, CA, USA

<sup>2</sup>Math & Computer Science, CSU East Bay, Hayward, CA, USA

**Abstract.** Enterprise security architecture is a unifying framework and reusable services that implement policy, standard and risk management decision. The purpose of the security architecture is to bring focus to the key areas of concern for the enterprise, highlighting decision criteria and context for each domain. TOGAF-9 architecture framework provides guidance on how to use TOGAF-9 to develop Security Architectures and SOA's. This paper addresses the enterprise architect of what the security architect will need to carry out their security architecture work. It is also intended as a guide to help the enterprise architect avoid missing a critical security concern.

**Keywords:** Enterprise Security Planning, Enterprise Architectures, TOGAF

## 1. Introduction

The Open Group Architecture Framework (TOGAF) is a framework - a detailed method and a set of supporting tools for developing enterprise architecture [1]. TOGAF 9 is much different from other architecture frameworks such as Zachman, as it is lot more process driven and gives you a way to essentially codify architectural patterns [2]. Key enhancement in TOGAF 9 is the introduction of a seven-part structure and reorganization of the framework into modules with well-defined objectives. This will allow future modules to evolve at different speeds and with limited impact across the entire blueprint -- something that's needed if you're looking to create architecture within compartments and have those compartments operating independently [1],[3],[5]. TOGAF 9, first of all, is more business focused. Before that it was definitely in the IT realm, and IT was essentially defined as hardware and software. The definition of IT in TOGAF 9 is the lifecycle management of information and related technology within an organization. It puts much more emphasis on the actual information, its access, presentation, and quality, so that it can provide not only transaction processing support, but analytical processing support for critical business decisions [4].

## 2. TOGAF Structure

As shown in Fig 1, TOGAF structure consists of;

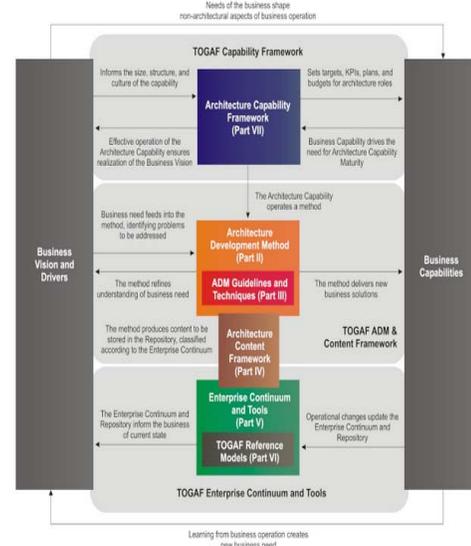


Figure 1. TOGAF Structure

**PART I (Introduction)** -This part provides a high-level introduction to the key concepts of enterprise architecture and in particular the TOGAF approach. It contains the definitions of terms used throughout TOGAF and release notes detailing the changes between this version and the previous version of TOGAF.

**PART II (Architecture Development Method)** - This part is the core of TOGAF. It describes the TOGAF Architecture Development Method (ADM) - a step-by-step approach to developing enterprise architecture.

**PART III (ADM Guidelines and Techniques)** This part contains a collection of guidelines and techniques available for use in applying TOGAF and the TOGAF ADM.

**PART IV (Architecture Content Framework)** This part describes the TOGAF content framework, including a structured metamodel for architectural artifacts, the use of re-usable architecture building blocks, and an overview of typical architecture deliverables.

**PART V (Enterprise Continuum & Tools)** This part discusses appropriate taxonomies and tools to categorize and store the outputs of architecture activity within an enterprise.

**PART VI (TOGAF Reference Models)** This part provides a selection of architectural reference models, which includes the TOGAF Foundation Architecture, and the Integrated Information Infrastructure Reference Model (III-RM).

**PART VII (Architecture Capability Framework)** This part discusses the organization, processes, skills, roles, and responsibilities required to establish and operate an architecture function within an enterprise.

The intention of dividing the TOGAF specification into these independent parts is to allow for different areas of specialization to be considered in detail and potentially addressed in isolation. Although all parts work together as a whole, it is also feasible to select particular parts for adoption whilst excluding others. For example, an organization may wish to adopt the ADM process, but elect not to use any of the materials relating to architecture capability [1].

### 3. TOGAF-9 Security Architecture

Security Architecture is a cohesive security design which addresses the requirements and in particular the risks of a particular environment/scenario and specifics what security controls are to be applied where. The design process should be reproducible. This definition is intended to specify only that, architecture is a design, which has a structure and addresses the relationship between the components [6][7].

#### 3.1 Security for Architecture Domains

All groups of stakeholders in the enterprise will have security concerns. These concerns might not be obvious as security-related concerns unless there is special awareness on the part of the IT architect. It is desirable to bring a security architect into the project as early as possible. In TOGAF 9, throughout the phases of the ADM, guidance will be offered on security-specific information which should be gathered, steps which should be taken, and artifacts which should be created. Architecture decisions related to security, like all others, should be traceable to business and policy decisions, which should derive from a risk analysis.

#### 3.2 Areas of Concerns for Security Architecture

**Authentication:** The authenticity of the identity of a person or entity related to the system in some way [8],[7].

**Authorization:** The definition and enforcement of permitted capabilities for a person or entity whose identity has been established.

**Audit:** The ability to provide forensic data attesting that the system was used in accordance with stated security policies.

**Assurance:** The ability to test and prove that the system has the security attributes required to uphold the stated security policies.

**Availability:** The ability of the system to function without service interruption or depletion despite abnormal or malicious events.

**Asset Protection:** The protection of information assets from loss or unintended disclosure, and resources from unauthorized and unintended use.

**Administration:** The ability to add and change security policies, add or change how policies are implemented in the system, and add or change the persons or entities related to the system.

**Risk Management:** The organization's attitude and tolerance for risk. (This risk management is different from the special definition found in financial markets and insurance institutions that have formal risk management departments.)

#### 3.3 Security Architecture Artifacts

Typical security architecture artifacts should include. 1.) Business rules regarding handling of data/information assets. 2.) Written and published security policy. 3.) Codified data/information asset ownership and custody. 4.) Risk analysis documentation. 5.) Data classification policy documentation.

### 3.4 ADM Security Architecture Requirement Management

Security Policies and security standards are one of the most important part of enterprise requirement management process. Security policies are established at executive level and have the characteristics like durability, resistant to impulsive change, and not technology specific. Once established act as a requirement for all architecture projects. Security standards are highly dynamic and state technological preferences used to support security policies. Security standards will manifest themselves as security-related building blocks in the Enterprise Continuum. Security patterns for deploying these security-related building blocks are referred to in the Security Guidance to Phase E.

New security requirements arise from many sources:

1. A new statutory or regulatory mandate
2. A new threat realized or experienced
3. A new IT architecture initiative discovers new stakeholders and/or new requirements.

In the case where 1. and 2. above occur, these new requirements would be drivers for input to the change management system discussed in Phase H. A new architecture initiative might be launched to examine the existing infrastructure and applications to determine the extent of changes required to meet the new demands. In the case of 3. above, a new security requirement will enter the requirements management system.

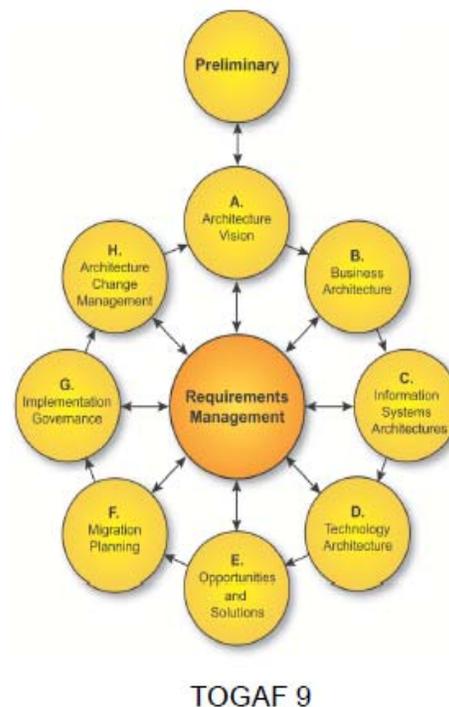
## 4. Security Architecture and ADM

Security architecture and ADM have eight different phases as explained below

### 4.1 Preliminary Phase

As shown in Fig 2, this phase is responsible for the defining and documenting applicable rules and security policies requirements. In TOGAF 9, ISO/IEC 17799:2005 is used for the formation of security policies. In order to implement these policies there is need to identify a security architect or security architecture team. Security considerations can conflict with functional considerations and a security advocate is required to ensure that all issues are addressed and conflicts of interest do not prevent explicit consideration of difficult issues. If the business model of organization

does encompass group of other organizations, then a common ground should need to be established between architects of different organization so they can develop interfaces and protocols for exchange of security information related to federated identity, authentication and authorization. So, the inputs to this phase would be written security policy, relevant statutes, list of applicable jurisdictions and outputs comes out in form of list of applicable regulations, list of applicable security policy, security team roster, list of security conditions and boundary conditions. [7][9].



**Figure 2. TOGAF Security Architecture and ADM**

### 4.2 Phase A (Architecture Vision)

In Phase A, the main intention of security architect is to obtain management support for security measures. All the security related architecture decision should be documented and the concern management peoples and executives should need to identified and frequently updated about the security related aspects of project. Tension between delivery of new business functions and security policies do exist. So the processes solving such disputes must be established at the early stage of the project. Other architects and management need to

identify that the role of security architect is safeguard the assets of enterprise. Any existing disaster recovery and business continuity plan must be understood and their relationship with the planned system must be defined and documented. All the architecture decisions must be made between the context of environment within which system will be placed and operate. So the physical, business and regulatory environment must be defined [7][9].

#### 4.3 Phase B Business Architecture

Phase B help to locate the legitimate actors who will interact with the product/service/process. Many subsequent decisions regarding authorization will rely upon a strong understanding of the intended users, administrators, and operators of the system, in addition to their expected capabilities and characteristics. It must be borne in mind that users may not be humans; software applications may be legitimate users. Those tending to administrative needs, such as backup operators, must also be identified, as must users outside boundaries of trust, such as Internet-based customers. The business process regarding how actors are vetted as proper users of the system should be documented. Consideration should also be made for actors from outside the organization who are proper users of the system. The outside entities will be determined from the high-level scenarios developed as part of Phase A. Security measures, while important, can impose burden on users and administrative personnel. Some will respond to that burden by finding ways to circumvent the measures. Examples include administrators finding ways to create "back doors" or customers choosing a competitor to avoid the perceived burden of the infrastructure. The trade-offs can require balancing security advantages against business advantages and demand informed judicious choice. Identify and document interconnecting systems beyond project control. Assets are not always tangible and are not always easy to quantify. Examples include: loss of life, loss of customer good will, loss of a AAA bond rating, loss of market share. Determine and document appropriate security forensic processes in order to proper implementation of security policies which in turn helps to catch the security breaches. Determine and document how much security (cost) is justified by the threats and the value of the assets at risk [7][9].

#### 4.4 Phase C Information Systems Architectures

A full inventory of architecture elements that implement security services must be compiled in preparation for a gap analysis. Every state change in any system is precipitated by some trigger. Commonly, an enumerated set of expected values of that trigger initiates a change in state. However, there are likely other potential trigger inputs that must be accommodated in non-normative cases. Additionally, system failure may take place at any point in time. Safe default actions and failure modes must be defined for the system informed by the current state, business environment, applicable policies, and regulatory obligations. Safe default modes for an automobile at zero velocity may no longer be applicable at speed. Safe failure states for medical devices will differ markedly from safe failure states for consumer electronics. Standards are justly credited for reducing cost, enhancing interoperability, and leveraging innovation. From a security standpoint, standard protocols, standard object libraries, and standard implementations that have been scrutinized by experts in their fields help to ensure that errors do not find their way into implementations. From a security standpoint, errors are security vulnerabilities. Presumably, in the event of system failure or loss of functionality, some value is lost to stakeholders. The cost of this opportunity loss should be quantified, if possible, and documented. Existing business disaster/continuity plans may accommodate the system under consideration. If not, some analysis is called for to determine the gap and the cost if that gap goes unfilled [7][9].

#### 4.5 Phase D (Technology Architecture)

Security architect should assess and baseline current security-specific technologies (enhancement of existing objective), revisit assumptions regarding interconnecting systems beyond project control, identify and evaluate applicable recognized guidelines and standards. Every system will rely upon resources that may be depleted in cases that may or may not be anticipated at the point of system design. Examples include network bandwidth, battery power, disk space, available memory, and so on. As resources are utilized approaching depletion, functionality may be impaired or may fail altogether. Design steps that identify non-renewable resources, methods that can recognize resource depletion, and measures that can respond through limiting the causative factors, or through limiting the effects of resource depletion to non-critical functionality, can enhance the overall reliability and availability of the system [7] [9].

#### 4.6 Phase E (Opportunities and Solution)

Identify existing security services available for re-use. From the Baseline Security Architecture and the Enterprise Continuum, there will be existing security infrastructure and security building blocks that can be applied to the requirements derived from this architecture development engagement. For example, if the requirement exists for application access control external to an application being developed, and such a system already exists, it can be used again. Statutory or regulatory requirements may call for physical separation of domains which may eliminate the ability to re-use existing infrastructure. Known products, tools, building blocks, and patterns can be used, though newly implemented. Also, Engineer mitigation measures addressing identified risks. Having determined the risks amenable to mitigation and evaluated the appropriate investment in that mitigation as it relates to the assets at risk, those mitigation measures must be designed, implemented, deployed, and/or operated. Since design, code, and configuration errors are the roots of many security vulnerabilities, taking advantage of any problem solutions already engineered, reviewed, tested, and field-proven will reduce security exposure and enhance reliability [7] [9].

#### 4.7 Phase F (Migration Planning)

In a phased implementation the new security components are usually part of the infrastructure in which the new system is implemented. The security infrastructure needs to be in a first or early phase to properly support the project. Secondly, during the operational phases, mechanisms are utilized to monitor the performance of many aspects of the system. Its security and availability are no exception. Security of any system depends not on design and implementation alone, but also upon installation and operational state. These conditions must be defined and monitored not just at deployment, but also throughout operation [7][9].

#### 4.8 Phase G (Implementation Governance)

Establish architecture artifact, design, and code reviews and define acceptance criteria for the successful implementation of the findings. Implement methods and procedures to review evidence produced by the system that reflects operational stability and adherence to security policies. To achieve all those things it is necessary to trained people to ensure correct deployment, configuration, and operations of security-

relevant subsystems and components; ensure awareness training of all users and non-privileged operators of the system and/or its components[7][9].

#### 4.9 Phase H (Architecture Management)

Incorporate security-relevant changes to the environment into the requirements for future enhancement (enhancement of existing objective) [7] [9].

### 5. Conclusions

Unless the security architecture can address a wide range of operational requirements and provide real business support and enablement, rather than just focusing upon short-term point solutions, then it will likely fail to deliver what the business expects. This type of failure is a common phenomenon throughout the information systems industry, not just in the realm of security architecture. Yet it is not sufficient to compile a set of business requirements, document them and then put them on the shelf, and proceed to design a security architecture driven by technical thinking alone. Being a successful security architect means thinking in business terms at all times, and setting up quantifiable success metrics that are developed in business terms around business performance parameters, not technical ones.

### 6. References

1. TOGAF version 9 Enterprise edition, <http://www.opengroup.org/architecture/togaf9-doc>
2. <http://www.infoworld.com/t/platforms/open-group-upgrades-enterprise-architecture-402>.
3. <http://www.infoworld.com/d/architecture/togaf-9-means-better-architecture-555>
4. <http://www.zdnet.com/blog/gardner/togaf-9-advances-it-maturity-while-offering-more-paths-to-architecture-level-it-improvement/2808>
5. <http://www.opengroup.org/architecture/togaf9-doc/arch/chap04.html>
6. [http://www.iss.ch/fileadmin/publ/agsa/Security\\_Architecture.pdf](http://www.iss.ch/fileadmin/publ/agsa/Security_Architecture.pdf)
7. <http://www.opengroup.org/architecture/togaf9-doc/arch/>

8. W. Stallings, "Cryptography and Network Security Principles and Practices", fourth edition, Prentice Hall, 2010
9. <http://www.slideshare.net/MVeeraragalloo/togaf-9-security-architecture-ver1-0-5053593>

# Enterprise Security Planning with Department of Defense Architecture Framework (DODAF)

L. Ertaul<sup>1</sup>, J. Hao<sup>2</sup>

<sup>1</sup>Math & Computer Science, CSU East Bay, Hayward, CA, USA

<sup>2</sup>Math & Computer Science, CSU East Bay, Hayward, CA, USA

**Abstract** – *The U.S Government Department of Defense employs DoDAF to develop and document its large and complex enterprise architecture. DoDAF itself has become a sizable and multifaceted subject matter. This paper is an aggregation of information about DoDAF, serves as a high lever overview and introduction to DoDAF, introducing key terms, concepts and development methodologies, as well as its application in dealing with enterprise security planning related issues.*

**Keywords:** Enterprise Architecture Frameworks, DoDAF, Enterprise Security Planning

## 1. Introduction

The Department of Defense Architecture Framework (DoDAF) is an enterprise architecture framework designed to model large and complex enterprises and systems, where their integration and interoperability pose challenges. DoDAF is especially designed to address the six core processes of the Department of Defense (DoD) [1]:

### 1. Joint Capability Integration and Development System (JCIDS) [1]

JCIDS ensures the capabilities needed for war missions are identified and met. Where gaps between the required and actual capability are identified, appropriate measures must be taken in order to prioritize and then bridge those gaps.

### 2. Defense Acquisition System (DAS) [1]

In order to achieve the National Security Strategy and Support employment and maintenance of the United States Armed Forces, a huge investment has to be made. The DAS is to manage this investment as a whole.

### 3. System Engineering (SE) [1]

SE looks at family-of-system and system-of systems. Its goal is to balance system performance with total cost while ensuring the developed systems will be the capability requirements.

### 4. Planning, Programming, Budgeting, and Execution (PPBE) [1]

The PPBE plays an important role from initial capability requirement analysis to decision-making for future programs.

### 5. Portfolio Management (PfM) [1]

PfM Primarily deals with IT investments. Its goals include maximizing return on investment while reducing associated risks in doing so.

### 6. Operations [1]

Operations define the activities and their inter-connections that support the military and business operations carried out by DoD.

As explained above, the six processes require decisions to be made at all levels of DoD. The need for an enterprise architecture framework is evident. The next section will outline the history of DoDAF.

## 2. History of DoDAF

In 1996, the first enterprise architecture framework was developed by DoD. It is called C4ISR. C4ISR stand for Command, Control Communication, Computers, Intelligence, surveillance and reconnaissance. It was developed in accordance to the changing face of modern warfare. [2]

After two iterations of C4ISR, DoDAF V1.0 was release in 2003. It broadened the applicability of architecture tenets and practices to all Mission Areas rather than just the C4ISR community [3]. It addressed usage,

integrated architectures, DoD and Federal policies, value of architectures, architecture measures, DoD decision support processes, development techniques, analytical techniques, (DoD) and moved towards a

In 2007, DoDAF V1.5 was release. DoDAF V1.5 incorporated net-centric concepts and elements, in order to service and support globally interconnected, end-to-end set of information, capabilities, associated processes, and personnel for collecting, processing, storing, disseminating, and managing information on demand to warfighters, policy makers, and support personnel [3].

DoDAF V2.0 was published in 2009. In DoDAF V2.0, the major emphasis on architecture development has changed from a product-centric process to a data-centric process designed to provide decision-making data organized as information [4]. The following section will look at this version of DoDAF in more detail.

### 3. DoDAF V2.0

DoDAF V2.0 consists of 3 volumes:

Volume 1 provides general guidance for development, use, and management of DoD architectures. This volume is designed to help non-technical users understand the role of architecture in support of major decision support processes. Volume 1 provides a 6-step methodology that can be used to develop architectures at all levels of the Department, and a Conceptual Data Model (CDM) for organizing data collected by an architecture effort. [5]

Volume 2 describes the construct of architectures, data descriptions, data exchange requirements, and examples of their use in developing architectural views in technical detail, to include the development and use of service-oriented architecture (SOAs) in support of Net-centric operations. Volume 2 provides a Logical Data Model (LDM), based on the CDM, which describes and defines architectural data; further describes the methods used to populate architectural views, and describes how to use the architectural data in DoDAF-described Models, or in developing Fit-for-Purpose Views that support decision-making. [5]

Volume 3 relates the CDM structure with the LDM relationships and associations, along with business rules described in Volume 2 to introduce a PES, which provides the constructs needed to enable exchange of data among users and COIs. [5]

repository-based approach by placing emphasis on architecture data elements that comprise architecture products [3].

### 3.1 Key Terminologies and Concepts

There are several key terminologies used in DoDAF V2.0, which are essential to understanding the framework:

**Models:** Visualizing architectural data is accomplished through models (e.g., the 'products' described in previous versions of DoDAF). Models (Which can be documents, spreadsheets, dashboards, or other graphical representations) serve as a template for organizing and displaying data in a more easily understood format [6].

**Views:** When data is collected and presented in a model format, the result is called a view [6].

**Viewpoints:** Organized collections of views (often representing processes, systems, services, standards, etc.) are referred to as viewpoints [6].

The DoDAF has eight viewpoints as shown in Fig 1. Each viewpoint has a particular purpose. It can be a broad summary information about the whole enterprise, or narrowly focused information for a specialist purpose. It can also be information on the connections and interactions of aspects of an enterprise.

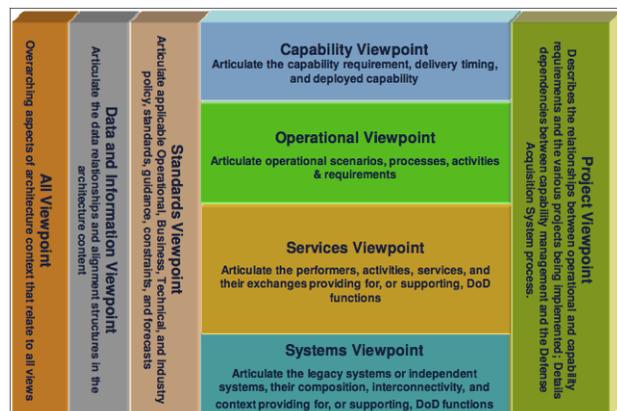


Figure 1. Architecture Viewpoints in DoDAF V2.0

The viewpoints namely are:

**All Viewpoint:** Some overarching aspects of an Architectural Description relate to all the views. The All Viewpoint (AV) models provide information pertinent to the entire Architectural Description, such as the scope and context of the Architectural Description. The scope

includes the subject area and time frame for the Architectural Description. The setting in which the Architectural Description exists comprises the interrelated conditions that compose the context for the Architectural Description. These conditions include doctrine; tactics, techniques, and procedures; relevant goals and vision statements; concepts of operations (CONOPS); scenarios; and environmental conditions [8].

*The Capability Viewpoint:* The Capability Viewpoint (CV) captures the enterprise goals associated with the overall vision for executing a specified course of action, or the ability to achieve a desired effect under specific standards and conditions through combinations of means and ways to perform a set of tasks. It provides a strategic context for the capabilities described in an Architectural Description, and an accompanying high-level scope, more general than the scenario-based scope defined in an operational concept diagram. The models are high-level and describe capabilities using terminology, which is easily understood by decision makers and used for communicating a strategic vision regarding capability evolution [9].

*The Data and Information Viewpoint:* The Data and Information Viewpoint (DIV) captures the business information requirements and structural business process rules for the Architectural Description. It describes the information that is associated with the information exchanges in the Architectural Description, such as attributes, characteristics, and inter-relationships [10].

*The Operational Viewpoint:* The Operational Viewpoint (OV) captures the organizations, tasks, or activities performed, and information that must be exchanged between them to accomplish DoD missions. It conveys the types of information exchanged, the frequency of exchange, which tasks and activities are supported by the information exchanges, and the nature of information exchanges [11].

*The Project Viewpoint:* The Project Viewpoint (PV) captures how programs are grouped in organizational terms as a coherent portfolio of acquisition programs. It provides a way of describing the organizational relationships between multiple acquisition programs, each of which are responsible for delivering individual systems or capabilities [12].

*The Services Viewpoint:* The Services Viewpoint (SvcV) captures system, service, and interconnection functionality providing for, or supporting, operational activities. DoD processes include warfighting, business,

intelligence, and infrastructure functions. The SvcV functions and service resources and components may be linked to the architectural data in the OV. These system functions and service resources support the operational activities and facilitate the exchange of information [13].

*The Standards Viewpoint:* The Standards Viewpoint (StdV) is the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements. Its purpose is to ensure that a system satisfies a specified set of operational requirements. The StdV provides the technical systems implementation guidelines upon which engineering specifications are based, common building blocks established, and product lines developed. It includes a collection of the technical standards, implementation conventions, standards options, rules, and criteria that can be organized into profile(s) that govern systems and system or service elements in a given Architectural Description [14].

*The Systems Viewpoint:* Systems Viewpoint (SV) captures the information on supporting automated systems, interconnectivity, and other systems functionality in support of operating activities. Over time, the Department's emphasis on Service Oriented Environment and Cloud Computing may result in the elimination of the Systems Viewpoint [15].

Under the eight viewpoints, there are 53 models in total. They are provided as pre-defined examples that can be used when developing presentations of architectural data [6]. However, DoDAF is designed as "fit-for-purpose", i.e., all the DoDAF-described models only need to be created when they respond to the stated goals and objectives of the process owner [6].

### 3.2 DoDAF Meta Model

An aid to defining and collecting data consistent with DoDAF V2.0 is provided by the DoDAF Meta-model (DM2). This meta-model (a model about data), replaces the Core Architectural Data Model (CADM), a storage format, referenced in previous versions of DoDAF. DM2 is a replacement for the CADM, but does not provide a physical data model. Instead, a Physical Exchange Specification (PES) is provided as an exchange mechanism, leaving the task of creation of a physical data model to the tool vendors. DM2 provides a high-level view of the data normally collected, organized, and maintained in an architecture effort. It also serves as a roadmap for the reuse of data under the federated approach to architecture development and management.

Reuse of data among communities of interest provides a way for managers at any level or area of the Department to understand what has been done by others, and also what information is already available for use in architecture development and management decision-making efforts. Finally, the DM2 can be used to ensure that naming conventions for needed data are consistent across the architecture by adoption of DM 2 terms and definitions [6].

As shown in Fig 2, DM2 has 3 views[16]:

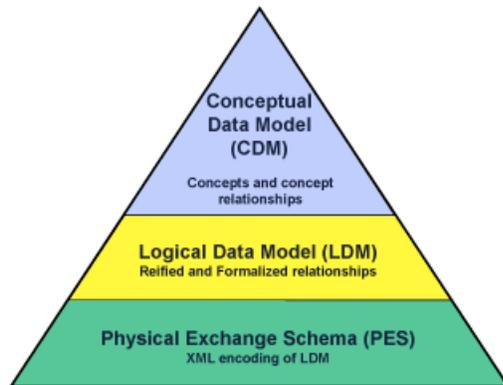


Figure 2. DM2 Views

*Conceptual Data Model (CDM)* defines the high-level data constructs from which architectures are created, so that executives and managers at all levels can understand the data basis of architecture. The CDM defines concepts and describes their relationships in relatively non-technically and easily understood terms [6].

*Logical Data Model (LDM)* adds technical information, such as attributes to the CDM and, when necessary, clarifies relationships into an unambiguous usage definition [6].

*Physical Exchange Specification (PES)* consists of the Logical Data Model with general data types specified and implementation attributes (e.g., source, date) added, and then generated as a set of XSD's, one schema per model/view [6].

In the next section, the DoDAF architecture development methodology is introduced.

## 4. DoDAF Architecture Development Methodology

DoDAF V2.0 is data-centric rather than product-centric (e.g., it emphasizes focus on data, and relationships among and between data, rather than DoDAF V1.0 or V1.5 products).

DoD employs a 6-step process in architecture development:

1. Determine intended use of architecture: the intended use is generally provided by the process owner. As DoDAF V2.0 uses fit-for-purpose models, the purpose and intended use of the architecture is defined in this initial step [6].

2. Determine scope of architecture: the scope of the architecture is determined by its intended use, as well as its linkage and intersection with other architectures. DoDAF also categorizes the scope of architecture into three levels: department, capability/segment and component level [6].

3. Determine data required to support architecture development: the categories of data needed must first be identified. The levels of details of each data category then need to be determined. Finally, the data that is needed to support architecture development is determined. DM2 provides a set of data definition and categories to aid this this.

4. Collect, organize, correlate, and store architectural data: the data can be collected from existing and/or new processes. In either case, the collected data must be validated and analysed by the subject-matter-experts (SMEs)[6].

5. Conduct analyses in support of architecture objectives: architecture-based analytics is a process that uses architectural data to support decision-making through automated extraction of data from a structured dataset, such as the use of a query into a database [6].

6. Document results in accordance with decision-maker needs: The final step in the architecture development process involves creation of architectural views based on queries of the underlying data. Presenting the architectural data to varied audiences requires transforming the architectural data into meaningful presentations for decision-makers [6].

## 5. DoDAF and Security

Security is often an add-on after the system is already built. DoDAF, like many other enterprise architecture frameworks, does not address security specifically. The consequence is that the tools and methodologies to perform security related design is not well conceived. Security is sometimes considered as a nonfunctional or performance system requirement, while sometimes a functional system requirement. In some other instances, security is deemed as an operational mission requirement. [17].

To address security requirement concerns, DoDAF identifies the following measure to counter potential threats and to reduce vulnerabilities:

*Physical* – the counter measures to physical threads, such as break-ins, thefts. Such measures include guards, guard dogs, fences, locks, sensors, including Closed Circuit Television, strong rooms, armor, weapons systems, etc [18].

*Procedural* – to reduce the risk of exploitation by unauthorized personnel, procedural specification is outlined (e.g. to ensure necessary vetting has been carried out for personnel to access security sensitive information and/or system) [18].

*Communication Security (COMSEC)* – data in transit is always under the threat of interception. COMSEC addresses such threats by means of encryption and other techniques to ensure the security in data transmission [18].

*Transient Electromagnetic Pulse Emanation Standard (TEMPEST)* – electronic equipment emit electromagnetic waves purposely or unintentionally. TEMPEST tackles such emission to ensure that no information is disclosed about the equipment, or the data being processed by the equipment [18].

*Information Security (INFOSEC)* – INFOSEC deals with the basic principles of information security: integrity, integrity, availability and confidentiality of data [18].

The utilization of the above measures reduces the security threat, but it also has adverse effect. The protection mechanisms tend to increase the complexity of the capability fulfillment, and therefore make it difficult and expensive to deploy. DoDAF analyzes the following four characteristics in order to assess the risks and apply minimum but necessary security measures:

*Environment* - The level of hostility of the environment the asset is exposed to [18].

*Asset Value* – the cost of the asset to be protected measured by the effect of loss, disclosure and replacement of such asset [18].

*Criticality* - an assessment of the criticality of the asset to enabling the government to undertake its activities [18].

*Personnel Clearance* - a measure of the degree of trustworthiness of the personnel that the government deems suitable to access to the asset [18].

DoDAF V2.0 does not have a separate viewpoint for security. Instead, it treats security like any other requirements [17]. DoDAF V2.0 and DM2 are working together to provide the mapping of viewpoints and concepts to the security characteristics. Below is a segment of the mapping table for the service viewpoint.

Table 1. Service Viewpoints and Concept Mapped to Security Characteristics and Protective Measures

Viewpoint	Concept	Security Characteristics	Protective Measures	Notes
Service	Capability Taxonomy	Security Marking Criticality Environment User Security Profile		The security characteristics of a capability taxonomy are to be derived from the constituent services.
Service		Security Marking Criticality Environment User Security Profile	Physical TEMPEST COMSEC	The environment of a service is derived from the Physical Asset to which is deployed. The User Security Profile is derived from the Organization which uses the service, its Criticality and Security Marking from its Functions.
Physical Asset		Environment	Physical TEMPEST	The environment identifies the worst environment to which the Physical Asset will be deployed.
Activity		Security Marking Criticality	INFOSEC Procedural	The Security Marking identifies the maximum security marking of the data the Function will process and the criticality represents the degree of harm to government operations if disrupted.
Resource Flow		Security Marking	COMSEC	The Security Marking represents the maximum security marking of the Resource Flow.
Performer and Activity		User Security Profile	Procedural	The User Security Profile is the lowest clearance of the user performing the function. This should be derived from Organizations who perform the Function, if the information exists.

The concepts such as activity, resource flow listed in Table 1 [18] are introduced in DM2. They are the data groups that form the building blocks of the architecture description [19]. Security characteristics are mapped to each of those building blocks to enable the assessment of the security risks and appropriate measures of protection.

## 6. Conclusions

From C4ISR to DoDAF, the underlying theme of the existence of such enterprise architecture framework to define concepts and models usable in DoD's core processes. DoDAF does so by providing views (models) to represent and document DoD's complex operations, so the broad scope and complexities of an architecture description can be visualized, understood and assimilated. Despite the lack of dedicated security viewpoint, DoDAF is able to deduce the necessary component and concepts needed to implement security requirements.

## 7. References

- [1] <http://cio-nii.defense.gov/sites/dodaf20/background.html>
- [2] C4ISR AWG, C4ISR Architecture Framework Version 2.0, 1997
- [3] Department of Defense, DoD Architecture Framework Version 1.5, Volume 1, 2007
- [4] [http://cio-nii.defense.gov/sites/dodaf20/whats\\_new.html](http://cio-nii.defense.gov/sites/dodaf20/whats_new.html)
- [5] Department of Defense, DoD Architecture Framework Version 2.0, Volume 1, 2009
- [6] Department of Defense, DoD Architecture Framework Version 2.0, The Essential DoDAF: A User's Guide to Architecture Description Development, 2009
- [7] <http://cio-nii.defense.gov/sites/dodaf20/viewpoints.html>
- [8] [http://cio-nii.defense.gov/sites/dodaf20/all\\_view.html](http://cio-nii.defense.gov/sites/dodaf20/all_view.html)
- [9] <http://cio-nii.defense.gov/sites/dodaf20/capability.html>
- [10] <http://cio-nii.defense.gov/sites/dodaf20/data.html>
- [11] <http://cio-nii.defense.gov/sites/dodaf20/operational.html>
- [12] <http://cio-nii.defense.gov/sites/dodaf20/project.html>
- [13] <http://cio-nii.defense.gov/sites/dodaf20/services.html>
- [14] <http://cio-nii.defense.gov/sites/dodaf20/standards.html>
- [15] <http://cio-nii.defense.gov/sites/dodaf20/systems.html>
- [16] <http://cio-nii.defense.gov/sites/dodaf20/DM2.html>
- [17] G.C. Dalton, J. Colobi, R. Mills Modeling "Security Architectures for the Enterprise"
- [18] <http://cionii.defense.gov/sites/dodaf20/security.html>
- [19] <http://cionii.defense.gov/sites/dodaf20/logical.html>

# Enterprise Security Planning using the Zachman Framework – Builder's Perspective

L. Ertaul<sup>1</sup>, S.Vandana<sup>2</sup>, K. Gulati<sup>2</sup>, G. Saldamli<sup>3</sup>

<sup>1</sup>Mathematics and Computer Science, CSU East Bay, Hayward, CA, USA

<sup>2</sup>Mathematics and Computer Science, CSU East Bay, Hayward, CA, USA

<sup>3</sup>MIS, Bogazici University, Istanbul, Turkey

**Abstract** - *In recent years enterprise architecture (EA) has acquired recognition as playing a pivotal role in change processes. Purported benefits of having enterprise architecture include improved decision making, improved adaptability to changing demands or market conditions, elimination of inefficient and redundant processes, optimization of the use of organizational assets and effectively achieve current and future objectives of the enterprise. By including security requirements in the EA process and security professionals in the EA team, enterprises can ensure that security requirements are incorporated into priority investments and solutions. Zachman Framework is a simple, logical and comprehensive enterprise architecture framework that can be used for enterprise security planning. This paper gives an overview of how Zachman's Framework helps in designing and implementing a streamlined, integrated enterprise security architecture. Also, discussed in this paper is a detailed specification of the security requirements from the builder's perspective of the Zachman Framework*

**Keywords:** Enterprise Architecture, Enterprise Security Planning, Zachman Framework

## 1 Introduction

In today's high-tech and interconnected world, every enterprise needs a well thought out security planning architecture. Security risks rise with the rise in the sophistication of enterprise products and enterprise as a whole. The rise of cloud computing, advancement of mobile, broadband and wireless communication clearly shows that enterprises need better control over the security mechanism. Threats exist from both within the walls of each enterprise as well as from external sources such as hackers, competitors and foreign governments. The goal of enterprise security planning is to have a detailed representation of the procedures, guidelines and practices for configuring and managing security in your environment. By enforcing enterprise security planning, enterprises can minimize their risks and show due diligence to their customers and shareholders.

Enterprise architecture provides a framework for reducing enterprise system complexity and enabling enterprise information sharing. In today's environment, each department or system usually has a vertically integrated approach to data, process, and technology. For example, department A has an application with its own database and runs on its own computer. Department B has another application with its own database and runs on its own computer. The same is true for department C. The Zachman Framework, named after John Zachman, has emerged as a way to develop enterprise-wide architecture. This framework moves from this vertical, departmental approach to a completely opposite horizontal approach. Instead of representing the data, process and technologies as entirely separate entities; he organized them around the points of view taken by various players [1] [2].

The Zachman Framework would seem to provide a sensible way to approach the security of an enterprise as it can accommodate different players involved in securing the enterprise and each player's view of the enterprise security. Its overall simplicity belies its use. Each of the framework's thirty six cells produces at least one output document to describe the system from that particular viewpoint.

This paper is organized as follows. In section 2, we briefly describe the enterprise architecture. In section 3, we briefly describe the Zachman framework which is most popular enterprise architecture framework. In section 4, we briefly describe how the Zachman framework can be applied for enterprise security planning. In section 5, we discuss the tools, technologies and security specifications from the builder's perspective of the Zachman Framework.

## 2 Enterprise Architecture

Enterprise Architecture (EA) is a rigorous description of the structure of an enterprise, which comprises enterprise components (business entities), the externally visible properties of those components, and the relationships (e.g. the behavior) between them. EA describes the terminology, the composition of enterprise components, and their relationships with the external environment, and the guiding principles for the requirement (analysis), design, and evolution of an enterprise [3][4][5].

This description is comprehensive, including enterprise goals, business process, roles, organizational structures, organizational behaviors, business information, software applications and computer systems.

An Enterprise Architecture Framework (EA Framework) is a framework for an Enterprise Architecture which defines how to organize the structure and views associated with an Enterprise Architecture [6].

The three basic components of the enterprise architecture framework are:

**-Views:** provide the mechanisms for communicating information about the relationships that are important in the architecture [6].

**-Methods:** provide the discipline to gather and organize the data and construct the views in a way that helps ensure integrity, accuracy and completeness [6].

**-Training/Experience:** support the application of method and use of tools [6].

In the next section we discuss basic overview of the Zachman Framework which is the most popular enterprise architecture framework. We will also discuss the different perspectives and abstractions of the Zachman framework.

### 3 Zachman Framework Overview

The Enterprise Architecture Framework (EA) frequently called the Zachman Framework, introduced in 1987 by John Zachman and extended by Sowa in 1992 (Sowa and Zachman 1992), as it applies to enterprises is a logical structure for classifying and organizing the descriptive representations of an enterprise that are significant to the management of the Enterprise as well as to the development of the enterprise's systems. It was derived from analogous structures that are found in the older disciplines of Architecture/Construction and Engineering/Manufacturing that classify and organize the design artifacts created over the process of designing and producing complex physical products (ex., buildings or airplanes.) [7].

The units of the Framework can also be understood as organization scheme for all kinds of metadata involved in building and using an information system and have therefore become widely recognized during the last years [7].

The Zachman Framework provides the thirty-six necessary categories for completely describing anything; especially complex things like manufactured goods (e.g., appliances), constructed structures (e.g., buildings), and enterprises (e.g., the organization and all of its goals, people, and technologies). The framework provides six increasingly

detailed views or levels of abstraction from six different perspectives as shown in Fig. 1[8].

It allows different people to look at the same thing from different perspectives. This creates a holistic view of the environment [8]. This Framework is intended being neutral in the sense that it's defined totally independents from tools or methodologies and therefore any tool or any methodology can be mapped against it to understand what they are doing, and what they are NOT doing. The Zachman Framework cannot be considered as either a modeling language, or a methodology, or a modeling notation [7].

In the next section we will have a detailed description of the different perspectives (rows of the Zachman Framework) of viewing any complex thing like an enterprise.

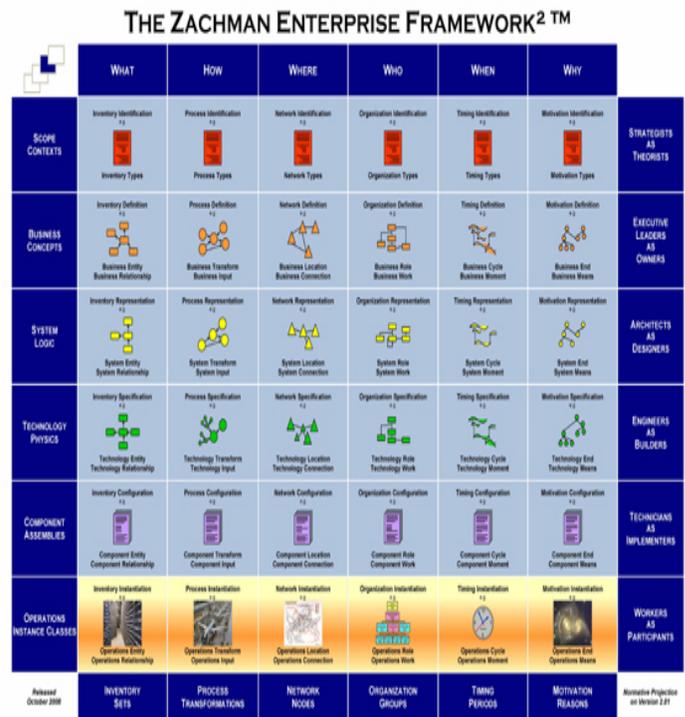


Figure 1. Zachman Framework published in year 2008 [8].

#### 3.1 Rows of the Zachman Framework

John Zachman's "Framework" is diagrammed in Figure 1. The rows represent the points of view of different players in the systems development process, while columns represent different aspects of the process [8]. The players are:

**- Scope (Ballpark view):** Definition of the enterprise's direction and business purpose. This is an industry view, concerned with the things that define the nature and purpose of the business. This is necessary to establish the context for any system development effort [8].

- **Model of the business (Owner's view):** This defines in business terms the nature of the business, including its structure, functions, organization, and so forth [8].

- **Model of the information system (Designer's view):** This defines the business described in step 2, but in more rigorous information terms. Where row two described business functions, for example, as perceived by the people performing them, row three describes them specifically as transformations of data. Where row two described all the things of interest to the enterprise, row three describes those things about which the organization wishes to collect and maintain information, and begins to describe that information [8].

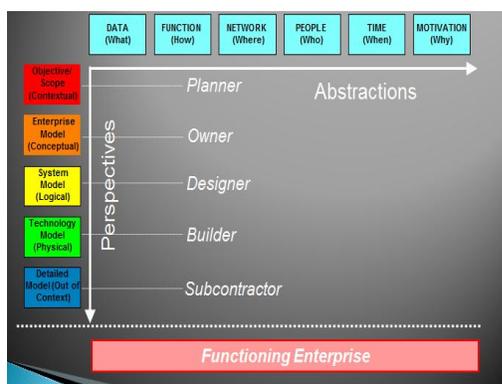
- **Technology model (Builder's view):** This describes how technology may be used to address the information processing needs identified in the previous rows. Here relational databases are chosen over network ones (or vice versa), kinds of languages are selected and program structures are defined, user interfaces are described, and so forth [8].

- **Detailed Description (Sub-Contractor's View):** This is a view of the program listings, database specifications, networks, and so forth that constitute a particular system and implementation of the system is done. These are all expressed in terms of particular languages [8].

- **Functioning system:** Finally, a system is implemented and made part of an organization [8].

In the next section we will have a detailed description of the different columns of the Zachman Framework.

### 3.2 Columns of the Zachman Framework



**Figure 2.** Perspectives and Abstractions of the Zachman Framework

The columns in the Zachman framework as shown in Fig 2 represent different areas of interest for each perspective. The columns describe the dimensions of the systems development effort. These are:

- **Data:** Each of the rows in this column address understanding of and dealing with any enterprise's data. This

begins in row one with a list of the things that concern any company in this industry, affecting its direction and purpose. As you pass down through the rows, you move to progressively more rigorous descriptions of the data (row two is the business person's view, and row three is a disciplined translation of this), until you get to row four, where a specific design approach (and a specific database management system) is specified. Row five is the detailed representation of the data on the computer and row six is the working database [8].

- **Function:** The rows in the function column describe the process of translating the mission of the enterprise into successively more detailed definitions of its operations. Where row one is a list of the kinds of activities the enterprise conducts, row two describes these activities in a contiguous model. Row three portrays them in terms of data transforming processes, described exclusively in terms of the conversion of input data into output data. The technology model in row four then converts these data conversion processes into the definition of program modules and how they interact with each other. Pseudo-code is produced here. Row five then converts these into source and object code. Row six is where the code is linked and converted to executable programs [8].

- **Network:** This column is concerned with the geographical distribution of the enterprise's activities. At the strategic level (row one), this is simply a listing of the places where the enterprise does business. At row two, this becomes a more detailed communications chart, describing how the various locations interact with each other. Row three produces the architecture for data distribution, itemizing what information is created where and where it is to be used. In row four, this distribution is translated into the kinds of computer facilities that are required in each location, and in row five, these facilities requirements are translated into specification of particular computers, protocols, communications facilities, and the like. Row six describes the implemented communications facilities [8].

- **People:** The fourth column describes who is involved in the business and in the introduction of new technology. The row one model of people is a simple list of the organizational units and each unit's mission. In row two, this list is fleshed out into a full organization chart, linked to the function column. Here also, requirements for security are described in general terms. In row three, the potential interaction between people and technology begins to be specified, specifically in terms of who needs what information to do his job. In row four, the actual interface between each person and the technology is designed, including issues of interface graphics, navigation paths, security rules and presentation style. In row five, this design is converted into the outward appearance of each program, as well as the definitions of access permissions in terms of specific tables and/or columns each user can have access to. In row six, you have trained people, using the new system [8].

- **Time:** The fifth column describes the effects of time on the enterprise. It is difficult to describe or address this column in isolation from the others, especially column two. At the strategic (row one) level, this is a description of the business cycle and overall business events. In the detailed model of the business (row two), the time column defines when functions are to happen and under what circumstances. Row three defines the business events which cause specific data transformations and entity state changes to take place. In the technology model (row four), the events become program triggers and messages, and the information processing responses are designed in detail. In row five, these designs become specific programs. In row six business events are correctly responded to by the system [8].

- **Motivation:** As Mr. Zachman originally described this column, it concerned the translation of business goals and strategies into specific ends and means. This can be expanded to include the entire set of constraints that apply to an enterprise's efforts. In row one; the enterprise identifies its goals and strategies in general, common language terms. In row two, these are translated into the specific rules and constraints that apply to an enterprise's operation. In row three, business rules may be expressed in terms of information that is and is not permitted to exist. This includes constraints on the creation of rows in a database as well as on the updating of specific values. In row four, these business rules will be converted to program design elements, and in row five they will become specific programs. In row six, business rules are enforced [8].

In the next section we outline the rules of the Zachman Framework.

### 3.3 Rules of the Zachman Framework

The framework comes with a set of rules:

-**The columns have no order:** The columns are interchangeable but cannot be reduced or created [8].

- **Each column has a simple generic model:** Every column can have its own meta-model [8].

- **The basic model of each column must be unique:** The basic model of each column, the relationship objects and the structure of it is unique. Each relationship object is interdependent but the representation objective is unique [8].

- **Each row describes a distinct, unique perspective:** Each row describes the view of a particular business group and is unique to it. All rows are usually present in most hierarchical organization [8].

- **Each cell is unique:** The combination of 2, 3 & 4 must produce unique cells where each cell represents a particular case. Example: A2 represents business outputs as they represent what are to be eventually constructed [8].

- **The composite or integration of all cell models in one row constitutes a complete model from the perspective of that row:** For the same reason as for not adding rows and columns, changing the names may change the fundamental logical structure of the Framework [8].

- **The logic is recursive:** The logic is relational between two instances of the same entity [8].

In the next section we will briefly describe how the Zachman framework can be used for Enterprise Security Planning.

## 4 Security Planning using Zachman Framework.

For security architecture modeling purposes, the columns of the Zachman matrix (data, function, network, people, time and motivation) are extremely useful. They provide the answers to what data assets the organization controls, how they are used, where they are located, the people involved and means to achieve a secured organization.

Similarly, the first five rows of the matrix give a unique perspective of a particular security challenge. The highest level, the Ballpark View, defines a clear and coordinated boundary (domain) of the system for the purposes of identifying the people, subsystems, and needs impacted by the system.

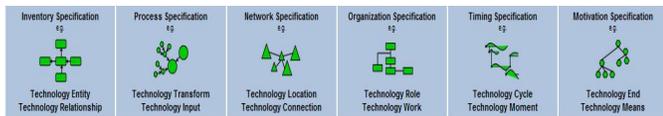
The Owner's View captures the business and organizational relationships, and their external interfaces. It also documents sources of system requirements, including those derived from legacy systems.

The Designer's View defines the functional capabilities of the system and establishes required interactions between subsystems. The Designer's View also establishes and documents the security architectural design and provides a basis for system measurement.

Finally, the Builder's View provides a detailed description of the design and methodology for monitoring and correcting system performance.

Each layer in the framework relates to a tool that can be used to secure the system. For example, an overall organizational security policy would be implemented in the Ballpark View. A tailored security policy and detailed descriptions are handled by the other rows.

## 5 Builder's Perspective



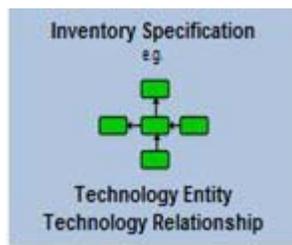
**Figure 3** . Row 4 (Builder's View) of the Zachman Framework

The technology model of the Zachman Framework defines the physical representations of the things in the enterprise as shown in Fig 3.. The builder specifies the technology to solve the problems. The builder applies the physical constraints of what is possible to the designer's artifacts and implements the product or service by understanding its environment [8]. The Builder integrates all the data sources evolved from different platforms and operating systems for providing a common enterprise wide view.

In the next sub section we will describe builder's view of the data column and specify possible tools and technologies from an enterprise security view point

### 5.1 Builder – Data Column

The builder specifies the tools and technology that can be used to ensure the confidentiality, integrity, availability, authenticity and non-repudiation of the designer's logical data model. The builder is concerned about the digital and physical data security.



**Figure 4.** Builder – Data Cell

To ensure data confidentiality and authentication services the builder decides which data has to be encrypted and specifies type of encryption based on the sensitivity of the data. This includes encryption, decryption, certificate management, key management and data recovery services in case the encrypted data is not available for any reason. Standard database security techniques are employed to prevent unauthorized access and denial of service attacks. The builder handles data storage management system and also specifies security mechanism and policies to be adopted if the data is stored in cloud computing environment. The builder specifies data backup and recovery tools to ensure data availability. For non-repudiation services the builder specifies the use of digital signatures and certificates.

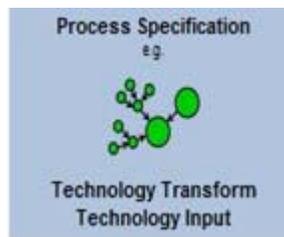
Table 1 specifies some of the tools and technologies that can be used for ensuring data confidentiality, data integrity, data availability, data authenticity and non-repudiation of data .The tools and technologies based on the designer's logical data model and industry security standards.

**Table 1** : Security tools and techniques for Builder – Data cell

<p><b>Data Encryption :</b>  <b>Symmetric encryption :</b> AES /3DES / blowfish  <b>Public key encryption :</b> RSA  <b>Disk/File encryption :</b> Microsoft EFS and Bit locker encryption system.  <b>DBMS column encryption :</b> Oracle 11g transparent data encryption/ Sybase column encryption  <b>Digital Signatures</b>  <b>Data Recovery/Availability</b>                  RAID, Windows Data Recovery, Recuva (Windows)                  Mirrored data servers  <b>Data in the cloud computing environment</b>(storage as a service):                  Cloud safety box, Open solaris VPC gateway, Trend micro secure cloud 1.1  <b>Data logging and monitoring</b>                  SIEM, CA Log manager, Event Viewer  <b>Data Authentication</b>                  eTrust Siteminder                  Single sign on, RSA Secure id, Kerberos Authentication  <b>Physical Security</b> - Use of shredders to dispose data  <b>Data Access Control</b>                  Biometrics, Oracle database vault  <b>Intrusion Detection System , Intrusio Prevention System</b></p>
--

In the next sub section we will describe builder's view of the function column and specify possible tools and technologies from an enterprise security view point

### 5.2 Builder – Process column



**Figure 5.** Builder – Process Cell

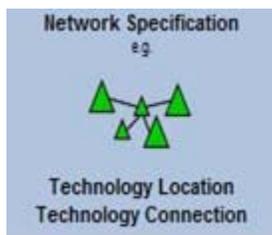
Column two of the Technology model describes the usage and functioning of the system [9]. This cell addresses the disaster recovery plans of the organization. The restoration activities include identifying the internal and external resources to handle damaged equipment and media in order to minimize the loss [9]. It also addresses operational security processes, training processes involved in the organization with asset management and data security processes. Table 2 lists the processes for builder-process cell.

<p><b>Intrusion Detection Process</b>                  Network based – SNORT                  Host based –                  OSSEC(Open Source Host based intrusion detection system)                  Tripwire                  AIDE(Advanced Intrusion detection Environment)                  Prelude Hybrid IDS</p> <p><b>Disaster Recovery Process:</b>                  NetBackup                  NetBackup PureDisk                  NetBackup Real Time                  Cluster Server                  Backup Exec                  Backup Exec System Recovery Server Edition                  Storage Foundation                  Volume Replicator</p> <p><b>Operational Security Processes</b>                  Hardware Controls                  Software Controls                  Input/Output controls                  Media controls</p> <p><b>Data Auditing Process</b>                  ACL                  Audit Exchange                  ActiveData CAAT software                  RemoteSysInfo</p> <p><b>Data Archiving Process</b>                  Tape Storage                  Disk Storage                  Cloud archiving</p> <p><b>Asset Management Process</b>                  Radio Frequency Identification                  GPS                  VANET</p> <p><b>Training Process</b></p>
---

**Table 2.** Security processes for Builder –Process cell

In the next sub section we will describe builder’s view of the network column and specify possible tools and technologies from an enterprise security view point.

**5.3 Builder - Network**



**Figure 6.** Builder – Network Cell

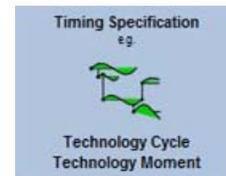
Column 3 of the technology model is concerned about the tools and technologies that can be used to secure the communication between the enterprise entities that are geographically distributed. The builder specifies tools/techniques to secure the communication links, the networking infrastructure, network monitoring and access control mechanism to be employed with the security policies. Table 3 specifies the tools/techniques to be used for builder – network cell.

**Table 3.** Security entries for Builder – Network cell

<p><b>Wireless Security :</b>                  -disable SSID broadcast                  -user authentication – 802.1X,EAP/EAP-fast,ACS for AAA                  -transport encryption -802.11,AES,TKIP,MFP,WPA/WPA2                  -detect and prevent rogue APs,clients,ad-hoc networks –Audits,RF Scanning,wireless IPS                  -VPN’s for remote access</p> <p><b>Wired and Wireless Security :</b>                  -VPN for remote encryption</p> <p><b>Link encryption</b> – Cisco Fiber channel link encryption (uses 128 bit AES)</p> <p><b>Email Security :</b> PGP</p> <p><b>Network device hardening</b></p> <p><b>Network Intrusion detection</b> – SNORT ,Fragrouter</p> <p><b>Network Management</b>                  -SNMP v3                  -Syslog                  -RMon</p> <p><b>Physical Security for Network Infrastructure</b>                  -Biometrics                  -Video Surveillance                  -Sensors                  -Incident Response</p> <p><b>Hardware device specification(eg : routers,switches)</b>                  -Network Availability ,MTTR(Mean time to repair),MTBF(mean time between failures)</p> <p><b>Logistics Security</b>                  - TMW Suite – Enterprise transportation software                  -Roadnet Transportation Software</p>
---

In the next sub section we will describe builder’s view of the Time column and specify possible tools and technologies from an enterprise security view point.

**5.4 Builder - Time**



**Figure 7.** Builder – Time Cell

Time cell is the physical representation of the system events and physical processing cycles expressed as control structures [10].

**Table 4.** Security entries for Builder – Time cell

<p>Regular Patch updates                  Regular Backups                  Password Management-Enterprise password safe                  Key Management                  Monitoring                  People training                  Information life cycle management – SAP Netweaver, Oracle 11g ILM                  Risk/Vulnerability Analysis – SSL Digger,Scuba,Site digger                  Task Management                  Resource Management</p>
---

In the next sub section we will describe builder’s view of the network column and specify possible tools and technologies from an enterprise security view point.

## 5.5 Builder - People



**Figure 8.** Builder – People cell

This cell is concerned with the physical representation of the work flow of the enterprise from security viewpoint. The below Table 5 presents the security entries for the builder – people cell.

**Table 5.** Security entries for builder – people cell

Workflow specification Specification of access privileges -Electronic access – access list , smart cards ,firewalls / routers -Physical access - sensors , alarms , ID Client User interface Metrics – performance,survey,bonuses Role specification
--

In the next section we will discuss the builder's view of the motivation column.

## 5.6 Builder – Motivation



**Figure 9.** Builder – Motivation cell

Column six deals with the constraints implied due to technological limitations and with the availability of resources and product construction[10].

**Table 6 :** Security entries for Builder- Motivation cell

Business Constrained Rules Budget Technological Constraints Availability of Hardware and Software Government Policies,Legal Changes Security Policies Environmental Regulations,Government Regulations Industry Standards
--

## 6 Conclusion

Zachman framework is simple and comprehensive framework and fits well to model the security of the enterprise. The six perspectives and abstractions bring out all the necessary security mechanism and policies to be adopted

for securing the enterprise. Though the Zachman framework seems to be a document heavy approach it still lets the enterprise assess its current state of security and make changes for a more secured environment. It would be better if the framework can be tweaked to fit the enterprise security planning ( for e.g. : we could add a customer row to the framework as well) but this is a limitation of using this framework. Though the Zachman Framework looks comprehensive enough to model the enterprise security other frameworks like Department of Defense Architecture Framework (DODAF) and TEAF can also be considered for enterprise security planning.

## 7 References

- [1] Hay, David C., *THE ZACHMAN FRAMEWORK: AN INTRODUCTION*, Essential Strategies, Inc.;
- [2] <http://www.tdan.com/view-articles/4140/>
- [3] Giachetti, R.E., *Design of Enterprise Systems, Theory, Architecture, and Methods*, CRC Press, Boca Raton, FL, 2010.
- [4] Enterprise Architecture Research Forum, <http://earf.meraka.org.za/earfhome/defining-ea>
- [5] MIT Center for Information Systems Research, Peter Weill, Director, as presented at the Sixth e-Business Conference, Barcelona Spain, 27 March 2007
- [6] Stephen Marley (2003). *Architectural Framework*. NASA /SCI. Retrieved 10 Dec 2008.
- [7] [http://www.mega.com/wp/active/document/company/wp\\_mega\\_zachman\\_en.pdf](http://www.mega.com/wp/active/document/company/wp_mega_zachman_en.pdf)
- [8] <http://www.essentialstrategies.com/documents/zachman2000.pdf>
- [9] <http://www.mcs.csueastbay.edu/~lertaul/ESP/article%252014.pdf>
- [10] <https://apps.adcom.uci.edu/EnterpriseArch/Zachman/ZIFA03.pdf>
- [11] [http://www.sans.org/reading\\_room/whitepapers/modeling/applying-security-enterprise-zac](http://www.sans.org/reading_room/whitepapers/modeling/applying-security-enterprise-zac)
- [12] [citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.4113&rep...](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.4113&rep...)
- [13] <http://www.zachmanframeworkassociates.com/>
- [14] *The Zachman Framework, For Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing*, John A. Zachman, Zachman International, Metadata Systems Software Inc., 2001-2006

# Enterprise Security Planning using Zachman Framework: Designer's Perspective

Levent Ertaul<sup>1</sup>, Archana R. Pasham<sup>2</sup>, Hardik Patel<sup>2</sup>

<sup>1</sup>Mathematics and Computer Science, CSU East Bay, Hayward, CA, USA

<sup>2</sup>Mathematics and Computer Science, CSU East Bay, Hayward, CA, USA

**Abstract** - An effective Enterprise Architecture framework can help an organization or an enterprise deal with the ever-changing business and technology needs and Zachman Framework is one such Enterprise Architecture framework. With Organizations having to operate businesses in a rapid changing climate, security is the biggest concern and an urgent issue for all organisations. Zachman Framework gives a structured tool enabling organizations to manage security at an enterprise level in a systematic, predictable, and adaptable way that fits their unique strategic drivers. This paper discusses how Zachman Framework can be used to secure an enterprise effectively. This paper attempts to present the understandings of the designers' perspective in detail. This paper proposes some entries which can be appropriate for the cells in row 3 from Enterprise security planning point of view.

**Index Terms** - Enterprise Architecture, Zachman Framework, Enterprise Security Planning.

## 1 Introduction

The term "enterprise architecture" is used in many contexts. It can be used to denote both the architecture of an entire enterprise, encompassing all of its information systems, and the architecture of a specific domain within the enterprise. In both cases, the architecture crosses multiple systems and multiple functional groups with the enterprise [4] [5].

Enterprise Architecture is a complete expression of the enterprise; a master plan which "acts as a collaboration force" between aspects of business planning such as goals, visions, strategies and governance principles; aspects of business operations such as business terms, organization structures, processes and data; aspects of automation such as information systems and databases; and the enabling technological infrastructure of the business such as computers, operating systems and networks[1].

The main goal of this paper is to discuss and understand Zachman Framework for enterprise architecture and also the roles and perspective of a designer in the Enterprise security planning. This paper has been organized as follows. Section 2, describes the enterprise architecture framework followed by definition, reason and benefits. Section 3, briefly describes the Zachman framework for enterprise architecture followed by definition, history, reason and brief overview of rows and

columns. Section 4, discusses the row 3 in detail with possible security related entities. Finally, in section 5, conclusion is given.

## 2 Enterprise Architecture Framework

Enterprise Architecture Framework provides a structured tool that manages and aligns an organization's business processes, Information Technology, application, people, operations and projects with the organization's overall strategy and goal. It provides a comprehensive view of the policies, principles, services & solutions, standards and guidelines in an enterprise [6].

### 2.1 Why Enterprise Architecture?

In today's time when the business competition is cut throat and with so many components attached to the business operation, if there is enterprise architecture and a framework that uses this architecture business can survive critical situations and achieve its overall organizational goal. Enterprise Architecture aligns an organization's business processes, Information Technology, application, people, operations and projects with the organization's overall strategy and goal and thus leading the organization to the success. Well defined and properly constructed Enterprise architecture helps an organization for future growth in response to the needs of the business.

### 2.2 Benefits of Enterprise Architecture

A well defined, property constructed and maintained enterprise architecture offers following benefits [3]:

- Highlighting opportunities for building greater quality and flexibility into applications without increasing the cost
- Supporting analyses of alternatives, risks, and trade-offs for the investment management process, which reduces the risks of building systems and expanding resources [3].

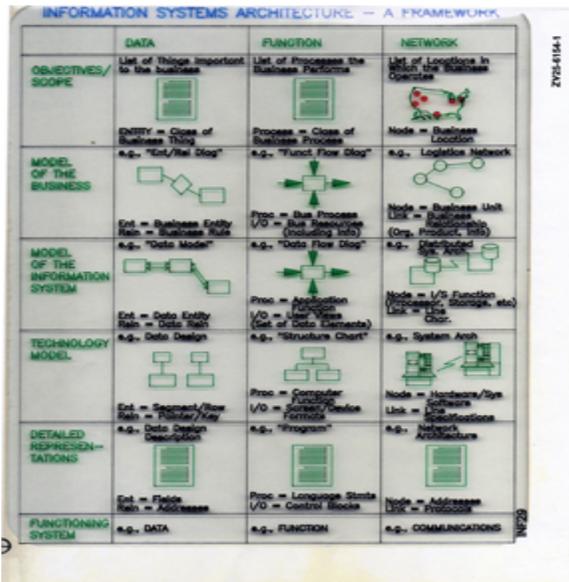
### 3 Zachman Framework for Enterprise Architecture

#### 3.1 Definition

The Zachman Framework™ is a schema - the intersection between two historical classifications that have been in use for literally thousands of years. The first is the fundamentals of communication found in the primitive interrogatives: *What, How, When, Who, Where, and Why* [8][14]. It is the integration of answers to these questions that enables the comprehensive, composite description of complex ideas. *The Zachman Framework™ is not a methodology for creating the implementation (an instantiation) of the object. The Zachman Framework™ is the ontology for describing the Enterprise* [8].

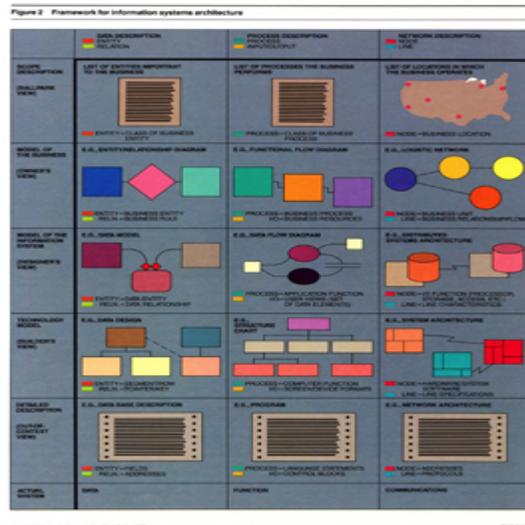
#### 3.2 Zachman Framework Evolutions

History of Zachman Framework dates back to 1984 (see Fig 1). Since the time of the inception to today's time, there have been no change in the basic concepts of the framework but the basic changes that can be seen over the years are related to the graphical representation.



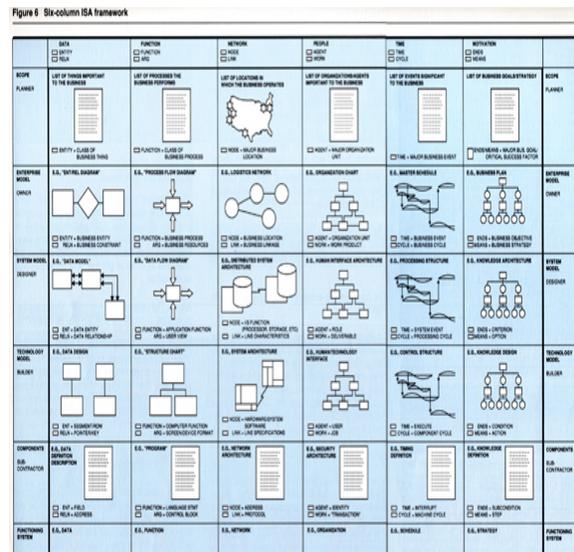
Source: [www.zachmaninternational.com](http://www.zachmaninternational.com)  
 Figure 1- Zachman Framework in 1984

1984: Figure 1 above shows the Zachman Framework in 1984, an original drawing where it has just 3 columns and it was named as "Information System Architecture". John Zachman had an idea of framework of 6 columns but he presented only 3 column framework because at that time people did not know much about Enterprise [14].



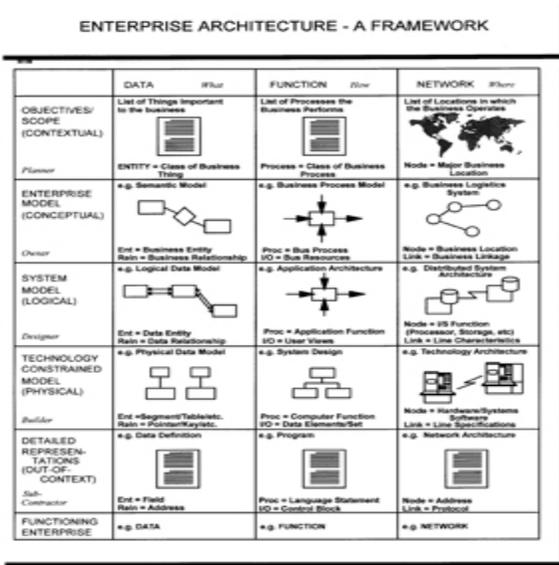
Source: [www.zachmaninternational.com](http://www.zachmaninternational.com)  
 Figure 2- Zachman Framework in 1987

1987: Figure 2 above shows the Zachman Framework in 1987. The original *Framework for Information Systems Architecture*. This is the original version published in the 1987 IBM Systems Journal. Notice that only the first 3 Columns made it in spite of all 6 existing [14].



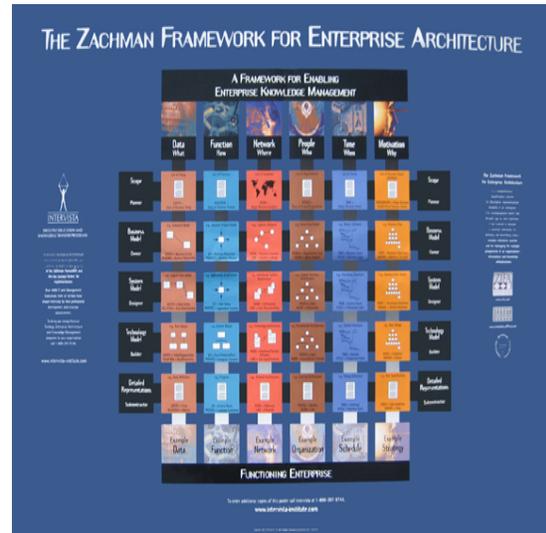
Source: [www.zachmaninternational.com](http://www.zachmaninternational.com)  
 Figure 3- Zachman Framework in 1992

1992: Still called *A Framework for Information Systems Architecture* in this 1992 IBM Systems Journal article. From above Fig 3, Note that John added the words "Owner," "Designer," and "Builder" to Rows 2, 3 and 4 for clarification [14].



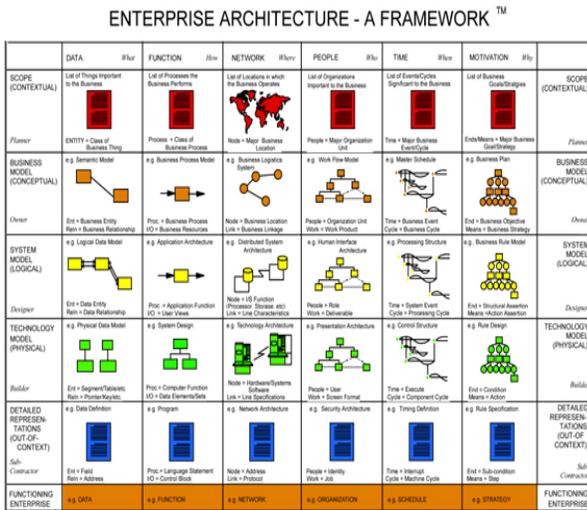
Source: [www.zachmaninternational.com](http://www.zachmaninternational.com)  
 Figure 4- Zachman Framework in 1993

**1993:** It was at this point that John decided to officially call *The Zachman Framework™: Enterprise Architecture - a Framework*. This version is still a minor carry-over from the 1987 article since it is only 3 columns. Notice from figure 4 above, that in this version is the first to use the adjectives "Contextual," "Conceptual," "Logical," "Physical" and "Out-of Context" defining the Rows [14].



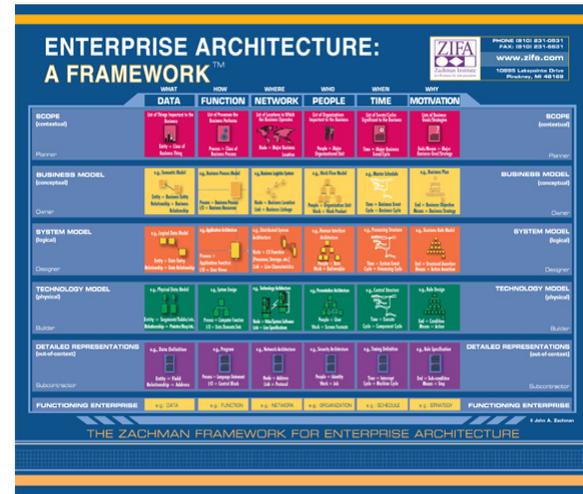
Source: [www.zachmaninternational.com](http://www.zachmaninternational.com)  
 Figure 6- Zachman Framework in 2002

**2002:** As shown in Fig. 6, one significant improvement in this version however, is the use of the black to white gradient between the cells - which works its way down the columns. The movement down each column has nothing to do with granularity; it has everything to do with *transformation* [14].



Source: [www.zachmaninternational.com](http://www.zachmaninternational.com)  
 Figure 5- Zachman Framework in 2001

**2001:** During this time, Enterprise Architecture was really gaining ground based on John's thoughts about the subject. Fully recognized as *The Zachman Framework™*, this version was very widely distributed and had many of the refinements from the previous 10 years of research (See Fig 5) [14].



Source: [www.zachmaninternational.com](http://www.zachmaninternational.com)  
 Figure 7- Zachman Framework in 2003

**2003:** This *Framework* (see Fig 7) does have some significant shortcomings. In addition, the colors of Rows 2 and 3 became inverted. Because of the colors of each Row, this *Framework* illustration emphasizes the Rows. [14].

**THE ZACHMAN ENTERPRISE FRAMEWORK<sup>2</sup>™**

	WHAT	HOW	WHERE	WHO	WHEN	WHY	
<b>SCOPE</b>	Inventory Identification Inventory Types	Process Identification Process Types	Network Identification Network Types	Organization Identification Organization Types	Timing Identification Timing Types	Motivation Identification Motivation Types	<b>STRATEGISTS AS FINANCISTS</b>
<b>BUSINESS</b>	Inventory Definition Business Entity Business Relationship	Process Definition Business Transform Business Input	Network Definition Business Location Business Connection	Organization Definition Business Role Business Work	Timing Definition Business Cycle Business Moment	Motivation Definition Business End Business Reason	<b>EXECUTIVE LEADERS AS OWNERS</b>
<b>SYSTEM</b>	Inventory Representation System Entity System Relationship	Process Representation System Transition System Input	Network Representation System Location System Connection	Organization Representation System Role System Work	Timing Representation System Cycle System Moment	Motivation Representation System End System Reason	<b>ARCHITECTS AS DESIGNERS</b>
<b>TECHNOLOGY</b>	Inventory Specification Technology Entity Technology Relationship	Process Specification Technology Transform Technology Input	Network Specification Technology Location Technology Connection	Organization Specification Technology Role Technology Work	Timing Specification Technology Cycle Technology Moment	Motivation Specification Technology End Technology Reason	<b>ENGINEERS AS BUILDERS</b>
<b>COMPONENT</b>	Inventory Configuration Component Entity Component Relationship	Process Configuration Component Transform Component Input	Network Configuration Component Location Component Connection	Organization Configuration Component Role Component Work	Timing Configuration Component Cycle Component Moment	Motivation Configuration Component End Component Reason	<b>TECHNICIANS AS IMPLEMENTERS</b>
<b>OPERATION</b>	Inventory Instantiation Operations Entity Operations Relationship	Process Instantiation Operations Transform Operations Input	Network Instantiation Operations Location Operations Connection	Organization Instantiation Operations Role Operations Work	Timing Instantiation Operations Cycle Operations Moment	Motivation Instantiation Operations End Operations Reason	<b>WORKERS AS PARTICIPANTS</b>
Released October 2007	<b>INVENTORY SETS</b>	<b>PROCESS TRANSFORMATIONS</b>	<b>NETWORK NODES</b>	<b>ORGANIZATION GROUPS</b>	<b>TIMING PERIODS</b>	<b>MOTIVATION REASONS</b>	Revised June 2011

Source: [www.zachmaninternational.com](http://www.zachmaninternational.com)  
 Figure 8- Zachman Framework in 2004

**2004:** After significant research starting in 2001, this copy of *The Zachman Framework™*, also known as *The Zachman Framework<sup>2</sup>™*, was developed in 2004 and is fairly recognizable (see Fig 8) [14].

### 3.3 Why Zachman Framework

With the use of Zachman Framework the costs are decreased, revenues are increased, processes are improved and business opportunities are expanded. Closer partnership between business and IT groups. Consistently proven itself [14][8]. It helps an organization achieve its business strategy; it gives the organization faster time to market for new innovations and capabilities [16].

### 3.4 Rules of Zachman Framework

- Rule 1:** Columns have no order [17].
- Rule 2:** Each column has a simple, basic model [17].
- Rule 3:** Basic model of each column is unique [17].
- Rule 4:** Each row represents a distinct view [17].
- Rule 5:** Each cell is unique [17].
- Rule 6:** Combining the cells in one row forms a complete description from that view [17].
- Rule 7:** Do not Create Diagonal Relationships between Cells [17].

### 3.5 Zachman Framework Rows Overview

- Row 1** – Scope - External Requirements & Definition of the Enterprise
- Row 2** – Enterprise Model - Business Process Modeling and Function Allocation
- Row 3** – System Model - Logical Models Requirements Definition
- Row 4** – Technology Model - Physical Models Solution Definition and Development
- Row 5** – As Built - As Built Deployment
- Row 6** – Functioning Enterprise - Functioning Enterprise Evaluation

**THE ZACHMAN ENTERPRISE FRAMEWORK<sup>2</sup>™**

	WHAT	HOW	WHERE	WHO	WHEN	WHY	
<b>SCOPE CONTEXTS</b>	Inventory Identification Inventory Types	Process Identification Process Types	Network Identification Network Types	Organization Identification Organization Types	Timing Identification Timing Types	Motivation Identification Motivation Types	<b>STRATEGISTS AS FINANCISTS</b>
<b>BUSINESS CONCEPTS</b>	Inventory Definition Business Entity Business Relationship	Process Definition Business Transform Business Input	Network Definition Business Location Business Connection	Organization Definition Business Role Business Work	Timing Definition Business Cycle Business Moment	Motivation Definition Business End Business Reason	<b>EXECUTIVE LEADERS AS OWNERS</b>
<b>SYSTEM LOGIC</b>	Inventory Representation System Entity System Relationship	Process Representation System Transition System Input	Network Representation System Location System Connection	Organization Representation System Role System Work	Timing Representation System Cycle System Moment	Motivation Representation System End System Reason	<b>ARCHITECTS AS DESIGNERS</b>
<b>TECHNOLOGY PARADIGMS</b>	Inventory Specification Technology Entity Technology Relationship	Process Specification Technology Transform Technology Input	Network Specification Technology Location Technology Connection	Organization Specification Technology Role Technology Work	Timing Specification Technology Cycle Technology Moment	Motivation Specification Technology End Technology Reason	<b>ENGINEERS AS BUILDERS</b>
<b>COMPONENT ASSEMBLIES</b>	Inventory Configuration Component Entity Component Relationship	Process Configuration Component Transform Component Input	Network Configuration Component Location Component Connection	Organization Configuration Component Role Component Work	Timing Configuration Component Cycle Component Moment	Motivation Configuration Component End Component Reason	<b>TECHNICIANS AS IMPLEMENTERS</b>
<b>OPERATIONS INSTANCES CLASSES</b>	Inventory Instantiation Operations Entity Operations Relationship	Process Instantiation Operations Transform Operations Input	Network Instantiation Operations Location Operations Connection	Organization Instantiation Operations Role Operations Work	Timing Instantiation Operations Cycle Operations Moment	Motivation Instantiation Operations End Operations Reason	<b>WORKERS AS PARTICIPANTS</b>
Released October 2007	<b>INVENTORY SETS</b>	<b>PROCESS TRANSFORMATIONS</b>	<b>NETWORK NODES</b>	<b>ORGANIZATION GROUPS</b>	<b>TIMING PERIODS</b>	<b>MOTIVATION REASONS</b>	Revised August 2011

Source: [www.zachmaninternational.com](http://www.zachmaninternational.com)  
 Figure 9- Zachman Framework in 2008

**2008:** Figure 9 is the most current evolution of *The Zachman Framework™* developed and is the version handed out to anyone who attends the *Complete MasterClass* in the *Zachman Certified™ – Enterprise Architect* program, which makes this representation a bit of a collector's item because of it's limited availability through the *Zachman Courses* [14].

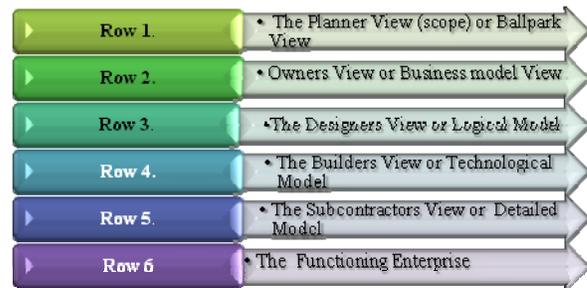


Figure 10- Rows of Zachman Framework

### 3.6 Zachman Framework Columns Overview

The basic model of each column is uniquely defined, yet related across and down the matrix. In addition, the six categories of enterprise architecture components, and the underlying interrogatives that they answer, form the columns of the Zachman Framework. Figure 11 shows clearly the description of each column.



Figure 11- Columns of Zachman Framework

### 4 Designers Role (Row 3) – In Detail

Designer is responsible for designing a part of the system, within the constraints of the requirements, architecture, and development process for the project. This row was originally called “information system designer’s view” in the original version of the ZF (see Fig. 10) [18]. The functionality of this fully attributed model is to reflect the enterprise model of the above (owner) row [2].

**Who is a designer?** The system analyst (Designer) represents the business in a disciplined form. Due to the increase in the number of users and complex IT environment, installing a firewall can no longer be the solution of security measures. Therefore, in this row the Designer hardens the applications and the operating system of the enterprise to ensure reliable security operations [18] [2].

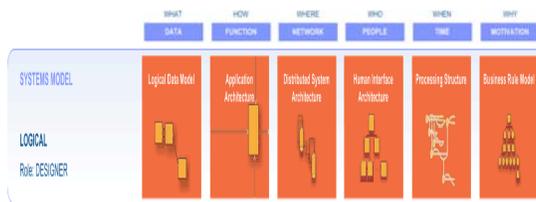


Figure 12- Row 3 of Zachman Framework

#### 4.1 Row3/Column 1 : Data/What

The first cell of Row 3 represents the logical data model, which describes the systems view of interest by transforming the real description of the product into its built in specifications. All the entries from owner go through validation over here. Figure 13 shows the possible entities of logical system model [2]:

**Data Verification Model:** Data Verification is a process wherein the data is checked for accuracy and inconsistencies.

Verification ensures that the specification is complete and that mistakes have not been made in implementing the model [15].

**Data Workflow Model:** A workflow consists of a sequence of connected steps. Workflow may be seen as any abstraction of real work, segregated in work share, work split or other types of ordering.



Figure 13- Entities of Zachman Framework Row 3/ Column 1

**Data Relationship Model:** Relationships are the logical connections between two or more entities .E-R (entity-relationship) Diagrams are used to represent Data relationship Models.

**Data Backup Model:** Data recovery is required because of the following reasons: Disaster recovery, virus protection, hardware failure, application error and user errors.

**Identity-Theft Model:** Identity theft is the wrongful use of another person’s identifying information—such as credit card, social security or driver’s license numbers—to commit financial or other crimes.

**Data Privacy Model:** The main challenge in data privacy is to share some data while protecting personal information. This privacy policy model combines user consent, obligations, and distributed administration [12].

**Data Security Model:** Data security is the practice of keeping data protected from corruption and unauthorized access. The focus behind data security is to ensure privacy while protecting personal or corporate data [12].

#### 4.2 Row3/Column 2 : Function/How

The second cell of Row 3, application architecture, discusses the information security policy function of enterprises which needs to mandate the backups of all data available at all times. The major things under consideration are the overall security of the data including the assurance of no data loss. Figure 14 shows the possible entities of application architecture.

**Disaster Recovery Process:** Figure 15 shows the key elements of disaster recovery planning process. A disaster recovery plan covers both the hardware and software required to run critical business applications and the associated processes to transition smoothly in the event of a natural or human caused disaster [11].



Figure 14- Entities of Zachman Framework Row 3/ Column 2



Figure 15- Disaster Recovery Planning Process

**Access Control Planning:** Access Control is any mechanism by which an authority grants the right to access some data, or perform some action. Access control systems provide the essential services of identification, authentication (I&A), authorization, and accountability [19].

**Data Archiving:** Data archiving is the process of moving data that is no longer actively used to a separate data storage device for long-term retention.

**Confidentiality, Integrity & Availability:** Confidentiality refers to limiting information access and disclosure to authorized users -- "the right people" -- and preventing access by or disclosure to unauthorized ones -- "the wrong people." Integrity refers to the trustworthiness of information resources. Availability refers, unsurprisingly, to the availability of information resources [20].

**Internal and External Processes:** This process is to define and control the value contribution of enterprise architecture and to integrate enterprise architecture into business.

### 4.3 Row3/Column 3 : Network/ Where

The third cell of Row 3, Distributed System Architecture defines the geographical boundaries and specification of the enterprise. The possible entries of this cell are as follows:

**Physical Security:** Physical security describes both measures that prevent or deter attackers from accessing a facility, resource, or information stored on physical media and guidance on how to design structures to resist various hostile acts[11].

**Link Security:** The types of links that fall under this category are Internet, Satellite Internet, Wireless and VPN.

**End to End Security:** End-to-end security relies on protocols and mechanisms that are implemented exclusively on the endpoints of a connection. End-to-End refers to hosts identified by IP (internet protocol) addresses and, in the case of TCP (transmission control protocol) connections, port numbers [12].

**Logistic security:** Logistics is the science of planning and implementing the acquisition and use of the resources necessary to sustain the operation of a system.

### 4.4 Row3/Column 4: People/ Who

The fourth cell of Row 3, Human Interface Architecture defines all the roles of the Individuals which are involved into the Enterprise. Figure 16 below lists all the possible entities [2].



Figure 16- Entities of Row 3/ Column 4

### 4.5 Row3/Column 5: Time / When

The fifth cell of Row 3, Processing Structure will define all the Timeline, Milestones, and Dependencies and other things for the Enterprise.

## 4.6 Row3/Column 6: Constraints/ Why

The sixth cell of Row 3 is a Business Rule Model. Figure 17 below lists the possible constraints for row 3.



Figure 17- Entities of Row 3/ Column 6

## 5 Conclusion

In this paper, Row 3 of Zachman framework (System model) helps organizations to standardize and control the processes that have a great impact upon both technical and non-technical departments. During the course of exploring Zachman framework we realized that though the logical concepts of this framework gives a look and feel of simplicity, it is far beyond that just that. For the effective application of Zachman Framework, We learnt that viewpoint of each player should be clearly defined and well structured. Zachman framework is helpful to achieve a better and stable design for later stage of development specially in situations where important changes are necessary and modifications are performed regularly. It is shown that Zachman Frame work can be used to plan security for Enterprises.

## References

- [1] J. Schekkerman, Institute for Enterprise Architecture Development Extended Enterprise Architecture Framework (E2AF), Essentials Guide, 2004. Available: <http://www.enterprise-architecture.info/>
- [2] L. Ertaul, R. Sudarsanam, "[Security Planning Using Zachman Framework for Enterprises](#)", Proceedings of EURO mGOV 2005 (The First European Mobile Government Conference), July, University of Sussex, Brighton, UK.
- [3] G. A. James, Robert A. Handler, Anne Lapkin, Nicholas Gall, Gartner Enterprise Architecture Framework: Evolution 2005., 2005  
Gartner, Inc. Available: [http://www.alaska.edu/oit/eas/ea/Gartner/gartner\\_enterprise\\_architect\\_130855.pdf](http://www.alaska.edu/oit/eas/ea/Gartner/gartner_enterprise_architect_130855.pdf)
- [4] <http://www.opengroup.org/architecture/togaf7-doc/arch/pl/enterprise.htm>
- [5] <http://www.togaf.org/togaf9/chap01.html>
- [6] Enterprise Architecture Center for Excellence, Available: <http://www.eacoe.org/EnterpriseArchitectureDefined.shtml>
- [7] <http://msdn.microsoft.com/en-us/library/bb466232.aspx>
- [8] Zachman Framework Associates, Toronto, Canada, July 2010. Available : <http://www.zachmanframeworkassociates.com/>
- [9] A Practical Guide to Federal Enterprise Architecture, Chief Information Officer Council, Version 1.0, February 2001. Available: <http://www.cio.gov/Documents/bpeaguide.pdf>
- [10] <http://cefarhangi.iust.ac.ir/download/courses/softwareengineering/E-Books/Ebook/Prentice%20Hall%20-%20A%20Practical%20Guide%20To%20Enterprise%20Architecture.pdf>
- [11] <http://www.cisco.com/warp/public/63/disrec.pdf>
- [12] [http://ksa.securityinstruction.com/index.php?option=com\\_content&view=article&id=83:physical-security-course&catid=3:courses&Itemid=11](http://ksa.securityinstruction.com/index.php?option=com_content&view=article&id=83:physical-security-course&catid=3:courses&Itemid=11)
- [13] [http://www.cisco.com/web/about/ac123/ac147/archived\\_issues/ipj\\_12-3/123\\_security.html](http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_12-3/123_security.html)
- [14] J. P. Zachman, *The Zachman Framework™ Evolution*, April 2009. Available: <http://zachmaninternational.com/index.php/ea-articles/100-the-zachman-framework-evolution>
- [15] <http://www.greenbook.org/marketing-research.cfm/having-faith-in-your-data-03377>
- [16] Zachman Framework Applied to Administrative Computing Services. Available: <http://apps.adcom.uci.edu/EnterpriseArch/Zachman/>
- [17] The Zachman Framework For Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing By John A. Zachman. Available : [http://www.businessrulesgroup.org/BRWG\\_RFI/ZachmanBook/RFIextract.pdf](http://www.businessrulesgroup.org/BRWG_RFI/ZachmanBook/RFIextract.pdf)
- [18] Practical Guide to Enterprise Architecture, A Zachman Framework. Available: <http://flylib.com/books/en/2.843.1.65/1/>
- [19] Active Directory Users, Computers, and Groups Available : <http://technet.microsoft.com/en-us/library/bb727067.aspx>
- [20] [http://www.yourwindow.to/information-security/gl\\_confidentialityintegrityandavailabili.htm](http://www.yourwindow.to/information-security/gl_confidentialityintegrityandavailabili.htm)
- [21] Other Architectures and Frameworks, Available: [http://www.opengroup.org/architecture/togaf8doc/arch/chap37.html#tag\\_38\\_04](http://www.opengroup.org/architecture/togaf8doc/arch/chap37.html#tag_38_04)

# Access Control Model and Algebra of Firewall Rules

Vladimir Zaborovsky, Vladimir Mulukha, Alexander Silinenko  
St. Petersburg state Polytechnical University  
Saint-Petersburg, Russia

*Abstract. The problems of information security are becoming especially important due to complexity and dynamic nature of information resources available through modern computer networks grows. Existing access models are based on the assumption that the security administrator knows the access object model and has full information about the users and their access rights. Obviously, the increasing complexity of networks requires increased functionality of firewalls that requires an adapting of their capabilities to the current state of the access environment. For the realization of adaptation algorithms it is necessary to develop an information access model of network resources. Proposed model has several levels of specification, including the macro-level describing common security policy, meso-level describing access objects and protocols and micro-level description of packet flows and firewall filtering rules, that satisfy macro-level security requirements. The article considers the new approach to formalization of access policy requirements using the algebra of filtering rules that applies to technological virtual connections. Rules parameters are formed using available network resources such as DNS and AD (LDAP).*

**Keywords – traffic management and security, access policy, resource model, firewall**

## I. INTRODUCTION

In modern computer networks there are complex subsystems used for information security purpose. One of the main subsystems is the access policy implementation environment, which is enforced by firewalls.

However, the implementations of access policy are far from simple due to dynamic nature of the information resources and the complexity of describing access policies through filtering rules for firewalls. Nowadays in virtual networks and clouds access control system has to take into account policy restrictions to access to corporate and external information resources [1]. Obviously, for the reliable protection of these resources we must have their formal model description. For the internal (corporate and static ) network resources such formal description is usually can be given by well-known models such as discretionary, mandatory or role-based access control models but for external resources it is required to take into account specific aspects of their operation ( applets, scripts, content and context ) [2].

To generate the filtering rules for firewall it is necessary to take into account such features of modern open information networks as:

- territorial distribution and concurrency;
- static requirement concerning network services;

- non-locality of network resources and parameters;
- “semantic gap” between security policy description and firewall configuration rules.

Actual problem which will be discussed below is description of the access policy requirements that can reduce the semantic gap between the corporate security policy description and filtering rules content that are implemented via security environment (firewalls). This problem can be solved by introducing multiple levels translation which based on describing algebra of firewall rules applied to packet flows.

This idea requires isomorphic description of the access policy requirements in terms of the information flows characteristics. In this case the overall structure of the access policy implementation may have three levels:

- 1) Macro-level – the requirements of common security policy;
- 2) Meso-level – an access objects description and protocols;
- 3) Micro-level – information flows description in terms of packet filters or stateful inspection .

Isomorphism requires a formalization of all description levels, using special algebra whose objects are the actions on the packet flows that do not have a semantic context. In this case many well-known security models of the past have become increasingly inadequate. That is why we proposed to describe traffic security procedure as a sequence of multiple translations of access requirements to the forms that available for firewall implementation. The main advantage of the proposed approach is the possibility of taking into account not only security requirements but also service information about users (AD, or LDAP catalogs), and hardware resources (DNS or SMNP servers), which can be controlled via firewall configuration in static or dynamic modes.

Described approach involves the solution of actual problems of modern algorithmic theory, in particular the practical use of Godel's incompleteness theorems, and the fact that any formal description is incomplete or contradictory. In this context the incompleteness of the access policy requirements means that security policy cannot be defined by the any using formalism of firewall rules, and the inconsistency means that the access objects may be simultaneously marked as the permitted and prohibited.

We offer to resolve this contradiction by using a hierarchical model in which on each level of access description three possible states for the flow of information are used: prohibited, permitted, non-prohibited.

The paper is organized as follows: in Section 2, the architecture of dynamic firewall configuration system and the descriptions of its main components are presented. In Section 3, the usage of the above mentioned algebra is described. The Section 4 is the conclusion and the discussion of the overall results.

## II. SECURITY SYSTEM ARCHITECTURE

Internet security is a main issue of modern information infrastructure. This infrastructure stores information in the form of distributed digital resources which have to be protected against unauthorized access. However, the implementations of this statement are far from simple due to the dynamic nature of network environment and users activity [3]. Below we describe a new approach to configure the security network appliances, that allows an administrator to overcome the semantic gap between security policy requirements and the ability to configure the firewall filtering rules [1]. The architecture of proposed system is presented in Fig.1.

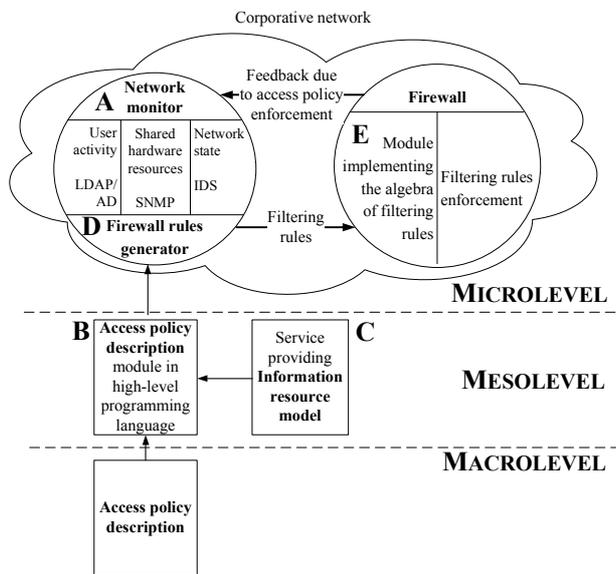


Figure 1. Security system architecture

where:

### A. Network monitor

Network monitor controls the whole system. Network environment state consists of three main parts:

- “User activity” is the information about what computer is currently used by what user. This information can be obtained from Microsoft Active Directory (AD) by means of LDAP protocol.
- “Shared hardware resources” is the information about network infrastructure and shared internal resources that can be described by network environment state vector  $X_k$
- “Network state” is the information about external network channel received from Intrusion Detection Systems (IDS).

### B. Access policy description module

Filtering rules of a firewall in itself are a formalized expression of an access policy. An access policy may simply specify some restrictions, e.g., “Mr. Black shouldn’t work with Youtube” without the refinement of the nature of “Mr. Black” and “work”.

There is a common structure of access policy requirements, which uses the notions of subject, action and object. Thus, the informally described requirement “Mr. Black shouldn’t work with Youtube” can be formally represented as the combination of the subject “Mr. Black”, the action “read”, the object “www.youtube.com” and the decision “prohibit”. This base can also be augmented by a context, which specifies various additional requirements restricting the cases of rule’s application, e.g.: time, previous actions of the subject, attributes’ values of the subject or object, etc.

However, access rules which are based on the notions of subject, action and object are not sufficient alone to implement complex real-world policies. As a result, new approaches have been developed. One of them, Role Based Access Control (RBAC) [4], uses the notion of role. A role replaces a subject in access rules and it’s more invariant. Identical roles may be used in multiple information systems while subjects are specific to a particular system. As an example, remember the roles of a system administrator and unprivileged user that are commonly used while configuring various systems. Administrator-subjects (persons) may be being added or removed while an administrator-role and its rules are not changing.

However, every role must be associated with some subjects as only rules with subjects can be finally enforced. During policy specification roles must be created firstly, then access rules must be specified with references to these roles, then the roles must be associated with subjects.

The OrBAC [5] model expands the traditional model of Role Based Access Control. It brings in the new notions of activity, view and abstract context. An activity is to replace an action, i.e., its meaning is analogous to the meaning of a role for a subject. A view is to replace an object. “Entertainment resources” can be an example of view, and “read” or “write” can be examples of an activity. Thus, the notions of role, activity, view and abstract context finally make up an abstract level of an access policy. OrBAC model allows to specify the access rules only on an abstract level using the abstract notions. Those are called the abstract rules. For instance, an abstract rule “user is prohibited to read entertainment resources”, where “user” is a role, “read” is an activity, and “entertainment resources” is a view. The rules for subjects, actions and objects are called concrete access rules.

To specify an OrBAC policy, a common language, XACML (eXtensible Access Control Markup Language) was introduced. The language maintains the generality of policy’s specification while OrBAC provides additional notions for convenient editing.

The main purpose of security appliances or firewalls development is to increase their performance and accuracy realization of access policy requirements by

means of traffic management (queueing), configuration methods (formal algebra) and environment characteristics identification algorithm or indicator functions. To reach this purpose it is necessary to choose a model of security appliances with customized management parameters which have to be adjusted in accordance of access policy. Nowadays there are some ways to solve this task. The standard in this area is eXtensible Access Control Markup Language (XACML) that based on IETF Framework for Policy-based Admission Control, which components include:

- Policy Decision Point (PDP) – XACML solution that makes the access decisions.
- Policy Enforcement Point (PEP) – the most security-critical component, which protects the resources and enforces the PDP's decision.
- Policy Administration Point (PAP) – the XACML-policy editor.

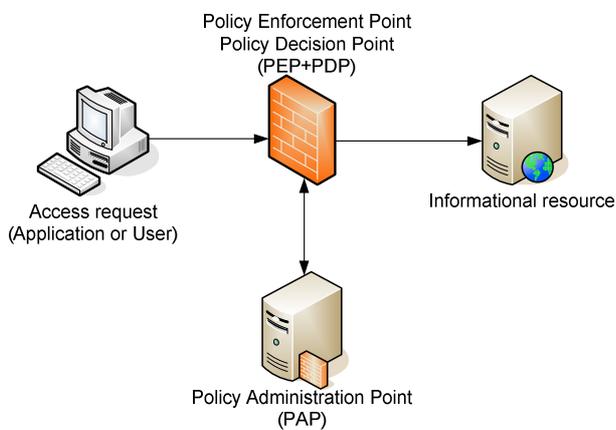


Figure 2. Firewall as a central component of access policy enforcement

In modern computer networks firewall combines PDP and PEP controlling access request and enforcing access decisions in real-time. In this case traffic can be considered as a huge number of data flows from various applications and/or users (Fig. 2). The firewall applies filtering rules for each data flow therefore its performance is strictly depends on the number of rules. To increase firewall performance PAP has to form minimal number of filtering rules that enforce the access policy.

### C. Firewall rules generator

There is a feature common for all firewalls: they execute an access policy. In common representation the main function of access control device (ACD) is to decide whether a subject should be permitted to perform an action with an object. A common access rule “Mr. Black is prohibited to read www.youtube.com”.

As was mentioned above, “Mr. Black” is a subject, “HTTP service on www.youtube.com” is an object, and reading is an action. So the configuration of ACD consists of common access rules that reference the subjects, actions and objects.

Although a firewall as an ACD must be configured with common access rules, each implementation uses its own specific configuration language. The language is often hardware dependent, reflecting the features of firewall’s internal architecture, and usually being represented by a set of firewall rules. Each rule has references to host addresses and other network configuration parameters. An example of the verbal description of a firewall rule may go as follows:

Host with IP address 10.0.0.10 is prohibited to establish TCP connections on HTTP port of host with IP address 208.65.153.238.

The main complexity of this approach is to find out how such elementary firewall rules could be obtained from common access rules.

Each firewall vendor reasonably aims at increasing its sales appeal while offering various tools for convenient editing of firewall rules. However, so far the problem of obtaining firewall rules from common access rules is not resolved in general. Moreover, this problem has not been paid much attention to.

The most obvious issue concerning this problem is that additional information beyond access rules is necessary in order to obtain the firewall rules. This information concerns the configuration of network services and the parameters of network protocols that are used for data exchange – “network configuration”. In general, it can be stored among the descriptions of subjects, actions and objects. An example:

Mr.Black: host with IP-address = 10.0.0.10;

www.youtube.com: HTTP service (port 80) on host with IP-address = 208.65.153.238.

Thus, the final firewall rules can be obtained by addition of the object descriptions to the access rules. It should be noted that even for small and especially for medium and large enterprises it is necessary to store and manage the network configuration separately from the security policy. The suggested approach allows us to achieve this goal: the security officer can edit the access rules with reference to real objects while the network administrator can edit the parameters of the network objects [6].

It should also be noted that there is no need to specify any fixed rules regarding association of the network parameters with the objects. For instance, HTTP port may be a parameter of an object or it may be a parameter of an action. A criterion is that the most natural representation of access policy must be achieved.

While generating the rules, the parameters of network objects can be automatically retrieved from various data catalogs. DNS is the best example of a world-wide catalog which stores the network addresses. Microsoft offers the network administrators the powerful means, Active Directory, to store information about users. Integration with the above mentioned technologies greatly simplifies the work of a security officer as he has only to specify the correct name of an object while forming firewall rules.

D. Information resource model

Interaction between subject and object in computer network can be presented as a set of virtual connections. Virtual connections can be classified as technological virtual connections (TVC) or information virtual connections (IVC). (See Fig.3).

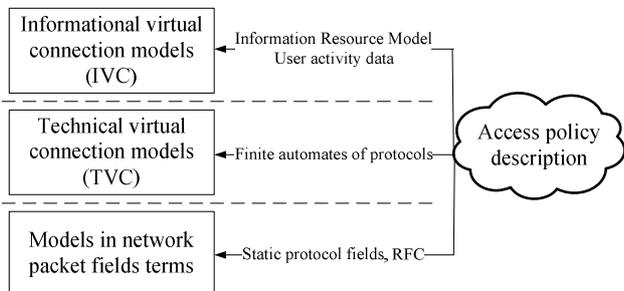


Figure 3. Layers of access control policies.

To implement the policy of access control, the filtering rules are decomposed in the form of TVC and the IVC. These filtering rules can be configured for different levels of the data flow description based on the network packet fields at the levels of channel, transport, and application protocols.

At different layers of access control policy model, the filtering rules have to take into account various parameters of network environment and objects. At the packet filter layer, a firewall considers standards static protocol fields described by RFC. At the layer of TVC, firewall enforces the stateful inspection using finite automata describing states of transport layer protocols. On the upper layer of IVC firewall must consider a-priori information about subject and object of network interaction [7].

As was mentioned above, the information about subject can be obtained from catalog services by LDAP protocol, e.g. Microsoft Active Directory.

According to existing approach [8] a resource model can be presented in:

- 1) logical aspect – an N-dimensional resource space model [9];
- 2) representation aspect - the definition based on standard high-level description languages like XML or OWL [10];
- 3) location aspect – the physical storage model of the resource including resource address.

All these approaches describe the network resource as a whole but don't take into account the specific access control task. Any remote network resource can be fully classified when the connection between this resource and local user would be closed. So it is necessary to control all virtual connections in real time while monitoring traffic for security purpose.

In this paper we propose to implement a special service external to the firewall that would collect, store and renew information about remote network objects. It should automatically create information resource model, describing all informational virtual connections that have to be established to receive this resource. This service

should periodically renew information about resource to keep it alive.

Firewall should cooperate with this external service to receive information resource model and enforce access policy requirements.

E. Algebra of filtering rules

As was mentioned above, the information security is defined by an access policy that consists of access rules. Each of these rules has a set of attributes; the basic ones among them are identifiers of subject and object and the rights of access from one to another. In TCP/IP-based distributed systems access rules have additional attributes that help to identify flows of packets (sessions) between the client and network application server. Generally these attributes identify the network subjects and objects at different layers of TCP/IP interaction model: MAC-addresses at link layer, IP-addresses at network layer, port numbers at transport layer and some parameters of application protocols.

The access policy in large distributed informational system consists of a huge number of rules that are stored and executed in different access control appliances. The generation of the access policy for such appliances is not very difficult: information must be made available for authorized use, while sensitive data must be protected against unauthorized access. However, its implementation and correct usage is a complex process that is error-prone. Therefore the actual problem of rule generation is representation, analysis and optimization of access policy for large distributed network systems with lots of firewall filtering rules. Below we propose an approach to description, testing and verification of access policy by the means of specific algebra with carrier being the set of firewall filtering rules.

According to proposed approach we define a ring as algebraic structure over set of filtering rules or  $R$  [11]. This ring consists of following operations over the elements of the set  $R$ :

1. Commutativity of addition:  $\forall a, b \in R \quad a + b = b + a$ .
2. Associativity of addition:  $\forall a, b, c \in R \quad a + (b + c) = (a + b) + c$ .
3. Zero element of addition:  $\forall a \in R \exists 0 \in R: \quad a + 0 = 0 + a = a$ .
4. Inverse element of addition:  $\forall a \in R \exists b \in R: \quad a + b = b + a = 0$ .
5. Associativity of multiplication:  $\forall a, b, c \in R \quad a \times (b \times c) = (a \times b) \times c$ .
6. Distributivity:  $\forall a, b, c \in R \quad \begin{cases} a \times (b + c) = a \times b + a \times c \\ (b + c) \times a = b \times a + c \times a. \end{cases}$
7. Identity element:  $\forall a \in R \exists 1 \in R: \quad a \times 1 = 1 \times a = a$ .
8. Commutativity of multiplication:  $\forall a, b \in R \quad a \times b = b \times a$ .

Let's define the algebra of filtering rules  $R = \langle R, \Sigma \rangle$ , where  $R$  – the set of filtering rules,  $\Sigma$  – the set of possible operations over the elements of  $R$ . The set of filtering rules  $R = \{r_j, j = \overline{1, |R|}\}$  – the carrier set of algebra  $R$ . Every rule

$r_j = \{X_1, \dots, X_N, A_j, B_1, \dots, B_M\}_j$  consists of a vector  $X_j$  of parameters, a binary variable  $A_j$  and a vector  $B_j$  of attributes.  $A_j \in \{0,1\}$  is a mandatory attribute that defines the action of access control system over packets;  $A_j=0$  means that packets must be dropped (access denied),  $A_j=1$  means that packets must be passed to receiver (access allowed);  $B_{ij} \in DB_j$  is a vector of attribute sets lengths to  $M$  ( $M$  can be 0). An example of elements of  $X_j$ :  $X_{j1}$  will be the set of client IP-addresses, and  $X_{j2}$  the set of server TCP-ports. The rule attributes  $B_j$  define the behavior of access control system that must be applied to corresponding flow of packets (session). The sets of possible values of parameter and attribute vectors are  $DX_1, \dots, DX_N$  and  $DB_1, \dots, DB_M$  in accordance with semantics of every parameter and attribute. For carrier set  $R$  the following expression is right (here “ $\times$ ” is the symbol of Cartesian product):

$$R \subset DX_1 \times DX_2 \times \dots \times DX_N \times DA \times DB_1 \times \dots \times DB_M$$

The set  $\Sigma = \{\varphi_1, \varphi_2\}$  defines the operations that are possible over filtering rules, where  $\varphi_1$  is the operation of addition,  $\varphi_2$  is the operation of multiplication.

The operation of addition for filtering rules is defined by the following expressions [11]:

$$r_3 = r_1 + r_2 = \{X_{11}, X_{12}, \dots, X_{1N}, A_1, B_{11}, \dots, B_{1M}\} + \{X_{21}, X_{22}, \dots, X_{2N}, A_2, B_{21}, \dots, B_{2M}\}$$

$$r_3 = \begin{cases} \{X_{11} \cup X_{21}, \dots, X_{1N} \cup X_{2N}, A_1 \vee A_2, B_{11} \cup B_{21}, \dots, B_{1M} \cup B_{2M}\}, & \text{if } A_1 = A_2; \\ \{X_{11} \Delta X_{21}, \dots, X_{1N} \Delta X_{2N}, A_1 \wedge A_2, B_{11} \Delta B_{21}, \dots, B_{1M} \Delta B_{2M}\}, & \text{if } A_1 \neq A_2, \end{cases}$$

where  $A_i$  is the attribute “the action of rule”,  $\cup$  is the union of sets,  $\Delta$  is the symmetrical difference of sets,  $\vee$  and  $\wedge$  are the logical disjunction and conjunction respectively. In other words the sum of two filtering rules is

- 4) union of sets of the same name parameters and attributes if the attribute “the action of rule” is equivalent in both rules;
- 5) symmetrical difference of sets of the same name parameters and attributes if the attribute “the action of rule” is different in summand rules.

The operation of multiplication for filtering rules is defined by following expressions:

$$r_3 = r_1 \times r_2 = \{X_{11}, X_{12}, \dots, X_{1N}, A_1, B_{11}, \dots, B_{1M}\} \times \{X_{21}, X_{22}, \dots, X_{2N}, A_2, B_{21}, \dots, B_{2M}\}$$

$$r_3 = \{X_{11} \cap X_{21}, X_{12} \cap X_{22}, \dots, X_{1N} \cap X_{2N}, A_1 \wedge A_2, B_{11} \cap B_{21}, \dots, B_{1M} \cap B_{2M}\},$$

where  $\cap$  – intersection of sets. In other words the product of two filtering rules is intersection of sets of the same name parameters and attributes; attribute “the action of rule” for result rule is a conjunction of corresponding attributes of initial rules.

Zero  $0_r$ , identity  $1_r$  and inverse  $-r$  elements of  $R$  are specifies by following expressions:

$$0_r = \{\emptyset, \emptyset, \dots, \emptyset, A, \emptyset, \dots, \emptyset\}, A = 0$$

$$1_r = \{DX_1, DX_2, \dots, DX_N, A, DB_1, \dots, DB_M\}, A = 1$$

$$-r = \{X_1, X_2, \dots, X_N, \bar{A}, B_1, \dots, B_M\},$$

where  $\bar{A}$  – logical inversion of  $A$

The described algebra is distributive commutative ring with identity element that means execution of corresponding axioms.

### III. FIREWALL CONFIGURATION USING PROPOSED ALGEBRA

Let's specify the element of set  $R$  as  $r = \{X_1, X_2, A_1\}$  where  $X_1$  – subset of source IP-addresses,  $DX_1 = [0.0.0.0, 255.255.255.255]$ ;  $X_2$  – subset of destination IP-addresses,  $DX_2 = [0.0.0.0, 255.255.255.255]$ ;  $A_1$  – attribute “the action of rule”,  $DA_1 = \{0,1\}$ , 0 denies access, 1 allows access. It is necessary to define the full and consistent access policy that allows establishing of sessions from Internal network (see schema on Fig. 4, a) to External subnetworks  $0.0.0.0 - 9.255.255.255$ ,  $20.0.0.0 - 49.255.255.255$  and from External subnetworks  $40.0.0.0 - 49.255.255.255$  to the whole Internal network.

For this task a convenient method of representation of access policy is 2-dimensional space  $x_1, x_2$ . Every point of this space is specified by the coordinates  $(x_1, x_2)$ . The set of points  $(x_1, x_2)$  is specified by Cartesian product of sets  $DX_1$  and  $DX_2$  (see on Fig. 4,b).

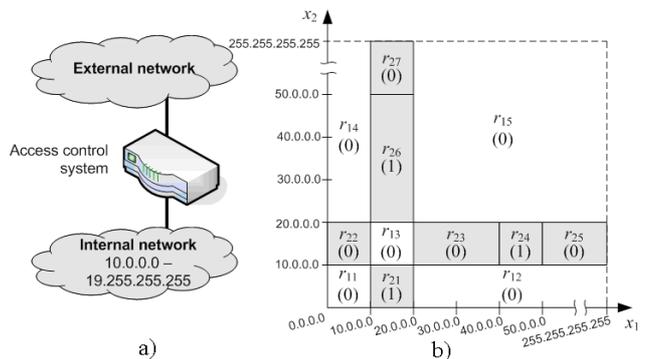


Figure 4. Access control system based on firewall (a) and access policy as a space of parameters (b)

Definition 1. The access policy is full if filtering rules specify the whole of space of parameters:

$$\forall x_1 \in DX_1, x_2 \in DX_2, \dots, x_N \in DX_N (x_1, x_2, \dots, x_N) \in \bigcup_{i=1}^{IR} (X_{i1}, X_{i2}, \dots, X_{iN})$$

Definition 2. The access policy is consistent if any point of space of parameters belongs only to own filtering rule:

$$\bigcap_{i=1}^{IR} (X_{i1}, X_{i2}, \dots, X_{iN}) = \emptyset$$

Obviously that for schema on Fig. 4 there are some forbidden areas as incorrect from the point of view of IP-network functionality. The following rules describe such areas (in Fig. 4,b these areas are colored white):

$$r_{11} = \{0.0.0.0 - 9.255.255.255; 0.0.0.0 - 9.255.255.255; 0\};$$

$r_{12} = \{20.0.0.0 - 255.255.255.255; 0.0.0.0 - 9.255.255.255; 0\};$   
 $r_{13} = \{10.0.0.0 - 19.255.255.255; 10.0.0.0 - 19.255.255.255; 0\};$   
 $r_{14} = \{0.0.0.0 - 9.255.255.255; 20.0.0.0 - 255.255.255.255; 0\};$   
 $r_{15} = \{20.0.0.0 - 255.255.255.255; 20.0.0.0 - 255.255.255.255; 0\}.$

Let's optimize this set of rules by applying the algebra's addition operation to rules  $r_{11}$  and  $r_{14}$ ,  $r_{12}$  and  $r_{15}$ :

$r_{17} = r_{11} + r_{14} = \{0.0.0.0 - 9.255.255.255; 0.0.0.0 - 9.255.255.255, 20.0.0.0 - 255.255.255.255; 0\};$   
 $r_{18} = r_{12} + r_{15} = \{20.0.0.0 - 255.255.255.255; 0.0.0.0 - 9.255.255.255, 20.0.0.0 - 255.255.255.255; 0\}.$

For other areas (colored gray in Fig. 4,b) it is necessary to specify the filtering rules according to the task conditions:

$r_{21} = \{10.0.0.0 - 19.255.255.255; 0.0.0.0 - 9.255.255.255; 1\};$   
 $r_{22} = \{0.0.0.0 - 9.255.255.255; 10.0.0.0 - 19.255.255.255; 0\};$   
 $r_{23} = \{20.0.0.0 - 39.255.255.255; 10.0.0.0 - 19.255.255.255; 0\};$   
 $r_{24} = \{40.0.0.0 - 49.255.255.255; 10.0.0.0 - 19.255.255.255; 1\};$   
 $r_{25} = \{50.0.0.0 - 255.255.255.255; 10.0.0.0 - 19.255.255.255; 0\};$   
 $r_{26} = \{10.0.0.0 - 19.255.255.255; 20.0.0.0 - 49.255.255.255; 1\};$   
 $r_{27} = \{10.0.0.0 - 19.255.255.255; 50.0.0.0 - 255.255.255.255; 0\}.$

These rules may be optimized also by applying of algebra's addition operation:

$r_{28} = r_{21} + r_{26} = \{10.0.0.0 - 19.255.255.255; 0.0.0.0 - 9.255.255.255, 20.0.0.0 - 49.255.255.255; 1\};$   
 $r_{29} = r_{22} + r_{23} = \{0.0.0.0 - 9.255.255.255, 20.0.0.0 - 39.255.255.255; 10.0.0.0 - 19.255.255.255; 0\}.$

As a result the access policy describes by following filtering rule set:

$$R = \{r_{13}, r_{17}, r_{18}, r_{24}, r_{25}, r_{27}, r_{28}, r_{29}\}.$$

Dimension of  $R$  is the main attribute that describes firewall performance characteristics. Usage of the algebraic operations of addition and multiplication allows us to reduce dimensionality of  $R$  and thus to increase the firewall performance while fulfilling requirements of the specific security policy [11]. However the correctness of each rule depends on an environment condition which can vary in real time. Therefore static description of access policy by means of proposed algebra is not enough and according to the telematics approach it is necessary to consider an environment condition with statistical parameters. Development of randomized model of the network environment considering these requirements, allows us to increase accuracy of the description of an access policy by means of filtering rules.

#### IV. CONCLUSION.

1. Each firewall is required to work in compliance with a security policy, user activities and network configuration. Policy requirements cannot be considered separately from methodology of proper firewall configuration and specified security characteristics. Based on OrBAC model it is possible to translate high-level abstract security requirements to low-level firewall configuration.

2. Firewall configuration can be largely automated based on specifying high-level access rules and parameters of corporate DNS, AD/LDAP, SNMP and IDS services. Proposed system architecture can be easily implemented

due to consideration of role-based information access models and characteristics of specific firewalls.

3. Proposed algebra of filtering rules is new mathematical description of access policy and a formal tool for firewall configuration. System approach provides possibility to prove fullness and consistency of an access policy. Proposed algebra is the base of optimization of the set of filtering rules and of the design of dynamic firewall configuration.

#### REFERENCES

- [1] V. Mulukha. Access Control in Computer Networks Based on Classification and Priority Queuing of the Packet Traffic, PhD. Thesis 05.13.19, SPbSPU, Russia, 2010
- [2] V. Zaborovsky, V. Mulukha. Access Control in a Form of Active Queuing Management in Congested Network Environment // Proceedings of the Tenth International Conference on Networks, ICN 2011 pp.12-17.
- [3] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. 2010. A view of cloud computing. *Commun. ACM* 53, 4 (April 2010), pp.50-58.
- [4] D.F. Ferraiolo and D.R. Kuhn. Role-Based Access Control. 15th National Computer Security Conference. (October 1992), pp. 554-563. (<http://csrc.nist.gov/groups/SNS/rbac/documents/ferraiolo-kuhn-92.pdf>)
- [5] <http://orbac.org/index.php?page=orbac&lang=en>
- [6] V. Zaborovsky, A. Titov. Specialized Solutions for Improvement of Firewall Performance and Conformity to Security Policy // Proceedings of the 2009 International Conference on Security & Management. v. 2. pp. 603-608. July 13-16, 2009.
- [7] V. Zaborovsky, A. Lukashin, S. Kupreenko Multicore platform for high performance firewalls. High performance systems // Materials of VII International conference – Taganrog, Russia.
- [8] H. Zhuge, The Web Resource Space Model, Berlin, Germany: Springer-Verlag, 2007
- [9] H. Zhuge, "Resource Space Grid: Model, Method and Platform," *Concurrency and Computation: Practice and Experience*, vol. 16, no. 14, pp. 1385-1413, 2004
- [10] D. Martin, M. Burstein, J. Hobbs, O. Lassila. et al. (November 2004) "OWL-S: Semantic Markup for Web Services," [Online]. Available: <http://www.w3.org/Submission/OWL-S/>.
- [11] A. Silinenko. Access control in IP networks based on virtual connection state models: PhD. Thesis 05.13.19: / SPbSTU, Russia, 2010.

# Information Security Risk Assessment Analysis

Ahmad Ghafarian, Ph.D.  
Dept of Math/Computer Science  
North Georgia College & State University  
Dahlonega, GA 30597  
(706) 864-1498  
aghafarian@northgeorgia.edu

Travis Smith  
Dept. of Information System and Management  
University of Maryland University College  
Adelphi, MD 20783  
(800) 888 8682  
Smith8640@msn.com

## ABSTRACT

*In order to properly reduce risk, you must be able to identify, quantify, and manage the various threats to the organization's assets. Since the results of an analysis will never be better than the various inputs used to conduct the assessment, it is important for the initial evaluation of the organization's assets and threats to be as accurate as possible. Assigning value to assets, assessing the likelihood of attack, calculating risk factors, reviewing controls and documenting findings all are components of a properly conducted risk assessment. It is vital for management to support both a qualitative and quantitative assessment of the organization in order to ensure the safety of the company's assets and its employees. Although current risk analysis procedures have been useful in assisting personnel in analyzing organizational status, improvements to the process are always sought after to provide for more accurate results.*

## Keywords

Risk assessment, threats, qualitative, quantitative.

## 1. Introduction

A risk assessment is conducted by an organization in order to help quantify potential losses that may be incurred from the myriad threats that exist. Current assessment guidelines and procedures are designed to assist personnel in the identification and mitigation of those losses and safeguard the fiscal continuity of the company. In order to assign risk levels to assets, the security professional must be able to accurately analyze the situation and apply the proper controls to mitigate or manage the vulnerabilities or threats. Properly done, this can allow for significant cost savings to a company as well as increasing safety to an organization's employees. After determining which of the numerous methodologies to use, risk assessment personnel must then assign value to organizational assets, identify the vulnerabilities of the assets as well as the likelihood of attack, calculate risk factors, choose and implement the best control, and document the findings for future review. This process allows management to prioritize their funding to controls that protect their company more efficiently and effectively. Although effective, there is always room for improvement to those risk assessment guidelines to allow for more efficient use of company time and resources.

## 2. Risk Assessment Methodologies

There are several different types of risk assessment methodologies that can be utilized to analyze an organization's risk. National Institute of Standards (NIST) and Technology Special Publication SP 800-66 was originally meant to be used in the healthcare field (HIPAA) but it was found to be effective in other types of industries, as well. The NIST approach deals primarily with information technology threats and how they relate to information security risks [4]. Another NIST publication, SP 800-30 [11], is used primarily by security personnel to manage the risk of computer systems [4]. These guidelines can be altered on an as-needed basis, at the discretion of the organization in order to suit their needs. Another approach which focuses more on qualitative process is Facilitated Risk Analysis Process (FRAP) [3]. Companies are able to plug in variables into the testing criteria, covering various aspects or iterations of the methodology and see how changes in the inputs may affect the results. This can allow for a broader understanding of the situation and allow for a more precise decision to be made to deal with specific circumstances. Yet another methodology is called Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) [1]. It is meant to be used on a more local level, with guidelines for companies to follow when they implement and manage information security within their own company. This type of evaluation relies on the individuals within the organization to provide the knowledge and expertise necessary to identify risks within their own departments and recommend potential controls. Employees can also submit their suggestions on the threats that exist and potential controls to be implemented by utilizing what is called the modified Delphi technique. This is a group method that mitigates the peer pressure to follow along with a superior's suggestions by allowing the individual to remain anonymous while the ideas are compiled and reviewed. The Cyber Incident Mission Impact Assessment (CIMIA) process that is focused on the collection and refinement of the mission value assets is described in [10]. This approach is suggested for risk analysis of real-time mission critical systems. In addition, several technological risk analysis tools and their features

have been described in [7]. We found that selecting a risk assessment methodology and a technological risk assessment tool much depends on the type of information systems and the type of organization.

### 3. System Characterization

The first step to conducting a risk assessment is to identify the assets of the organization and then assign value to those assets [15]. This valuation can be based on the physical costs of replacing or maintaining the equipment or facilities or they can be more subjective, such as what the cost would be for the company should sensitive information be leaked or what adversaries would pay to get their hands on the data. The cost of repairing equipment, the loss of productivity, and the costs to maintain and protect it are all tangible factors used in assigning value. Since not all assets can have a physical value, those less tangible components require a best guess effort for value determination. Items such as damage to reputation, corruption of data and value of intellectual property are much harder to assess but just as important, nonetheless. Quantification of all organizational assets, both tangible and intangible, is vital in the risk analysis process in order to identify which are more valuable to the company and may require additional protective measures. Just as important to understand is the cost to the company should controls *not* be put in place to protect the various assets and resulting damages incur. This cost/benefit analysis can assist in the determination of specific safeguards to be implemented, the amount of insurance to obtain and also to comply with legal and regulatory requirements.

In general, risk is the probability of *vulnerability*, multiplied by the *value of the information asset*, minus the percentage of *risk mitigated* by current controls, plus the uncertainty of current knowledge of the *vulnerability* [15]. This defines the calculation needed to perform an accurate assessment of risk and assign a tangible value to it. However, assigning tangible value is not a trivial task and organizations use different techniques and tools to do the valuation. There are many asset valuation techniques that are available to the security risk assessment team. Choosing the appropriate technique requires an understanding of the various techniques and project requirement of security risk assessment [16]. Knowledge of these asset valuation techniques will help the risk assessment team in identification of all physical (and intangible) assets and their individual value to the whole. Automated tools exist that can map network resources or identify all of the hardware and software of an organization, which can be valuable for larger companies to keep track of their physical or logical assets. Some business models are fairly universal in their goals and requirements, so can sometimes find existing tools to meet

their needs. For example, EBIOS is a method for estimating the Information Technology risk and supports the managers in the process of defining the requirements and defining the scope of the analysis. In conjunction with the Common Criteria, and continuous development in the information technology security management, EBIOS becomes overall risk management technique [7]. Other tools such as *COBRA*, *RiskWatch*, and *CRAMM* provide services to government agencies, HIPAA, and Enterprise organizations [7] [11]. These tools can run a company just a few hundred dollars or upwards of tens of thousands, depending on their complexity and functionality and are primarily question-based, with the agencies themselves filling in the appropriate inputs (technology, environment factors, etc.) in order for the tool to run its scenarios and generate a risk profile for the user. Unless an organization falls within the common parameters of a regulated industry or can find a tool model that accurately fits their organization, managers may have to rely on the longer to implement, but more accurate way of assigning value to their assets and then take into account the likelihood of attack into the equation when assigning funds for countermeasures.

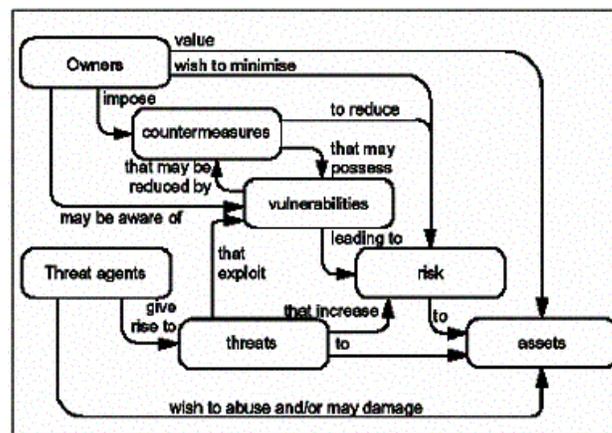


Figure 1: Common Criteria Security Model. Courtesy of John H. Souders

### 4. Likelihood of Attack

Determining the likelihood of attack is all about identifying the vulnerabilities and threats to an organization and figuring out what the chance is that a certain weakness will be exploited. Questions such as what threats exist, how often are they expected to occur and what the impact would be if something did happen should be forefront in the mind of the person (or group) performing the analysis. Threat agents such as malware, hackers, natural disasters, or even employees can all be involved in exploiting the weak links in company armor, resulting in a slew of damages including denial of service

to resources, virus infection, system malfunction or even loss of life, to name a few. Once this identification has been made, a numerical value is assigned to the specific likelihood of attack. This can be an arbitrary number, to be used to gauge the chance that such an event might occur. For example, the chance that a company may experience flooding in Arizona may be slight but it is still a chance, so is assigned a lower value than say, the higher chance that equipment may overheat due to the failure of site HVAC systems.

Organizations oftentimes have the capability to guess the likelihood of certain attacks with some accuracy by reviewing past performance factors or by utilizing already researched factors such as facility fire ratings or e-mail virus occurrences. Professionals can also monitor past historical incidents of various attacks that have occurred outside of their company in order to make a good guess as to whether that same attack is likely to occur within their own organization. Utilizing a specific equation for reference (Annual Loss Expectancy = [Asset Value \* Single Loss Expectancy] \* Annual Rate of Occurrence) [4] analysts can make recommendations on which threats pose the greatest threats to an organization and allocate their funding as appropriate. Determining what the cost would be to recover from a loss if the recognized threat is exploited helps to identify which control should be used based on these formulas.

## 5. Control Analysis

Proper countermeasure selection and implementation is the key to the protection of an organization's assets (to include the human factor). By performing a cost/benefit analysis, management can identify which control would best suit their needs and fall within budget to perform. This can be done by utilizing a simple equation: (Annual Loss Expectancy before the safeguard) – (ALE after the safeguard is in effect) – (the annual cost of the safeguard) = the value of the safeguard to the organization [4]. Basically, if the cost to implement and maintain the control is higher than the annual cost of repairing damage caused from the threat it is meant to protect, than the control is not worth it to purchase and put into effect. Some countermeasures are fairly inexpensive to implement (such as a security sticker that homeowners sometimes put in their windows, indicating they are protected by an alarm system), and other times the safeguard requires significant resources to be effective. Incorporating a safeguard that is highly visible can be an effective deterrent but it is vital that the safeguard is not easily overcome. Knowing that an Intrusion Detection System (IDS) is installed in a network can deter attackers only as long as the IDS do not have a vulnerability that is easily exploited. To deal with risk, a decision must be made to either transfer the risk (insurance), avoid it by

stopping the activity that is creating the risk, accepting it (live with it), or to reduce the risk through the implementation of various controls.

Controls have many different characteristics that are desired by security personnel. They can be modular, are easily upgradable, can be tested, don't inhibit the company assets, or allow user interaction, to name but a few. Improving upon procedures, changing the organizational environment, or installing various security devices are all means of reducing risk through controls. The effectiveness of the control (and its functionality) must be analyzed by those performing the assessment and determine which ones would best suit their needs based on the funding they have available. Once those controls have been implemented, periodic review of those safeguards must be completed in order to guarantee that it still remains the proper control for the situation. At this point, the risk assessment process begins anew, with assets being identified, values assigned, threats/vulnerabilities discovered, and the proper controls put in place, with all aspects of the process documented to allow for an accurate review to be accomplished. The documentation of each stage of the process is necessary in order to justify the control recommendations to senior management. The comparisons that a listed document can show are invaluable in determining the appropriate response to a viewed weakness in an organization. It also serves as a reference for future risk assessments (like a baseline, of sorts) to see the delta from one analysis period to the next and potentially allow for trends to be discovered and possibly predicted.

## 6. Future Assessments

Current risk assessment guidelines and procedures have been fairly effective in assisting personnel in their goal of information security protection. Organizations have several tools available to assist them in their endeavors, but more can be done to prepare for the worst. Limitations of the current system lie in the fact that humans are often involved with identifying and quantifying assets as well as discovering vulnerabilities and threats that need to be addressed. Humans are fallible. If more automated tools become available to deal with more organizational types (industries, company sizes, etc.), then the subjective process of asset identification can be improved upon. The tools may recommend or discover resources that the human factor may have missed. Same can be said for the identification of potential fixes for weaknesses discovered in the organization and recommend the ideal countermeasures for those vulnerabilities based upon a budget range. All of this would need to be programmed and tested by people, of course, but through multiple iterations of

testing and patching, a suitable tool can be developed that could help people with their risk assessment task.

If automated tools are not developed to improve upon the risk assessment process, another potential improvement would be to decrease the re-evaluation period of the process. Instead of companies performing an assessment on an annual basis, they could instead do them every quarter. Technology changes rapidly and responses to those changes may be slow in coming if a review is not completed that recognizes that fact. Decreasing the required review timeframe window and developing it into a more dynamic process may entail the hiring of specialized personnel to oversee the actions but may end up saving the organization in the long run by catching vulnerabilities as they develop and implementing the appropriate countermeasures before they become an issue for the company. It is important to note here that an organization should tailor their assessments to their business goals and not simply incorporate as many controls as possible to bypass the need for continual re-evaluation.

## 7. Conclusion

Risk assessments are meant to provide an organization with a snapshot of their current system and allow for more insightful decision-making regarding the designation of company funds towards the mitigation of potential threats. The process is meant to increase awareness of organizational assets and their respective vulnerabilities and give management and security personnel a better understanding of where bolstering of defenses are needed. The risk assessment process is meant to be step-by-step in order to provide for an easier undertaking, but requires significant manpower to be devoted to the gathering of departmental information for an accurate risk profile. Automated tools can save a company a lot of time and money by performing the risk calculations with only required variables input into them, but organizations need to be vigilant to not rely solely on a program to tell them what they should do to protect them. The human factor comes into play at this stage and reasoning must be used to determine whether the software suggestions are appropriate for their situation or meet the organization's mission focus or goals. The future of the risk assessment process will most likely be one in which companies take an increasingly active role in analyzing their systems and the budgets for security will continue to rise as they realize the importance of proper protections to the safety of their assets. Those larger organizations, or those with smaller funding may need to rely on a more detailed analysis in order to properly match their needs with their capability, but companies that think that they cannot afford to purchase and maintain certain safeguards or

programs may end up completing a risk assessment and find they can't afford not to.

## 8. References

- [1] Albert J. Christopher; Sandra G. Behrens, Richard T. Pethia, and William R. Wilson (1999). *Operationally Critical Threat, Asset and Vulnerability Evaluation (OCTAVE)*. Technical Report CMU/SEI-99-TR-017 and ESC-99-TR-017.
- [2] Baltzan, Paige & Phillips, Amy. (2009). *Business Driven Information Systems: Second Edition*. Boston, MA.
- [3] Elky, Steve (2006). *An Introduction to Information System Risk Management*. SANS Institute, 2007.
- [4] Harris, Shawn. (2010). *All In One CISSP Exam Guide: Fifth Edition*. New York, NY. Kozierok, C. M. 2001. Summary comparison of RAID levels. Retrieved November 25, 2008, from The PC Guide Web site: <http://www.pcguides.com/ref/hdd/perf/raid/levels/com-p-c.html>
- [5] NIST, National Institute of Standard and Technology, Special Publications 800-series. Retrieved February 22, 2011 from: <http://csrc.nist.gov/publications/PubsSPs.html>
- [6] ProQuest Information and Learning Company. (2005). *New Asset Valuation Tools*. Retrieved November 7, 2010 from [http://findarticles.com/p/articles/mi\\_qa5392/is\\_200505/ai\\_n21372751/](http://findarticles.com/p/articles/mi_qa5392/is_200505/ai_n21372751/)
- [7] Rot Artur, 2009, *Computer Support of Risk Analysis in Information Technology Environment*, Proceedings of the 2009 International Conference on Security and Management (SAM'09), pp.67-72
- [8] Saunders, John H. *A dynamic Risk Model for Information Technology Security in a Critical Infrastructure Environment*. Retrieved March 1, 2011 from: <http://www.johnsaunders.com/papers/riskcip/RiskConference.htm>
- [9] Schreider, Tari. (2003). *Risk Assessment Tools: A Primer*. Retrieved November 3, 2010 from <http://www.isaca.org/Journal/Past-Issues/2003/Volume-2/Pages/Risk-Assessment-Tools-A-Primer.aspx>
- [10] Sorrels M. David and Michael R. Grimaila *Towards Predictive Mission Risk Analysis to Operationalize Mission Assured Networking*, Proceedings of the 2010 International Conference on Security and Management (SAM'10), pp. 542-547.
- [11] Schrieder, Tari (2003). *Risk Assessment tools: A primer*. Information System Control Journal, Vol 2.
- [12] Stoneburner, Gary, Alice Goguen, and Alexis Feringa (2002). *Risk Management Guide for Information Technology Systems*. NIST Special Publication 800-30

- [13] Toigo, J. W., 2004. Data at risk. *Network Computing*, 15(1), 37-44. Retrieved November 20, . <http://www.giac.org/resources/whitepaper/planning/132.pdf>
- [14] Wetter, Joern. (2005). *Academic Learning Series: Security+ Certification*. Microsoft Press, Redmond, WA.
- [15] Worrell, C. 2007. Recovery strategies. Retrieved November 22, 2008, from Global Information Assurance Certification Web site:
- [16] Whitman M, and Mattord, 2007. Principles of Incident Response and Disaster Recovery. Course Technology, NJ.

# SAT-based Verification of Data-Independent Access Control Security Systems

Yean-Ru Chen<sup>1</sup>, Jui-Lung Yao<sup>2</sup>, Chih-Sheng Lin<sup>3</sup>, Shang-Wei Lin<sup>4</sup>, Chun-Hsian Huang<sup>5</sup>

Ya-Ping Hu<sup>6</sup>, Pao-Ann Hsiung<sup>7</sup>, Sao-Jie Chen<sup>8</sup> and I-Hsin Chou<sup>9</sup>

<sup>1,8</sup>Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan 106, ROC

<sup>2-7</sup>National Chung Cheng University, Chiayi, Taiwan 621, ROC

<sup>9</sup>Institute of Nuclear Energy Research, Taoyuan, Taiwan 325, ROC

**Abstract**—*The Harrison-Ruzzo-Ullman problem is the verification of a set of policy rules, starting from an initial protection matrix, for the reachability of a state in which a generic access right is granted. Three decades ago, it was shown to be undecidable; however, recently Kleiner and Newcomb (KN) used communicating sequential processes to prove that the model checking of data-independent security systems against universal safety access temporal logic (SATL) is decidable. Nevertheless, this restricted KN problem still lacks an automatic verification method. As a solution, we modeled it as a satisfiability problem such that a set of policy rules can be model checked against a universal SATL property without explicitly constructing the state model a priori. This is made possible by a key technique called permission inheritance. Besides proving the correctness and termination of the proposed method, two real cases namely employee information system and nuclear power plant security system are also used to illustrate the feasibility and efficiency of the proposed method.*

**Keywords:** SAT, access control, verification, security

## 1. Introduction

The importance of ensuring protection and security has grown rapidly since we connected every system to the Internet. For example, *cloud computing* has not only changed the exact computing and storage locations of our data, but has also introduced new concerns for data and computing security. Within the realm of security, *access control* is one of the most widely used and thoroughly studied mechanisms [15]. However, the state-of-the-art techniques in verifying access control policies are mostly based on theoretical analysis or explicit state-based verification [14]. A more symbolic approach is proposed in this work for the automatic verification of access control policies.

Access control prevents unauthorized use of resources by implementing a security policy that specifies who or what may have access to each specific system resource and the type of access that is permitted in each instance [15]. A basic formulation to specify access control policies is called the *access control matrix* which was proposed by Lampson [13] and subsequently refined by Graham and Denning [7]

and by Harrison et al. [9]. This work adopts the *protection matrix* [11] to model access control policies. The Harrison-Ruzzo-Ullman (HRU) problem [9] is the verification of a set of policy rules, starting from an initial protection matrix, for the reachability of a state in which a generic access right is granted. Three decades ago, it was shown to be undecidable; however, recently Kleiner and Newcomb (KN) [11] used communicating sequential processes to prove that the model checking of data-independent security systems against *universal safety access temporal logic* is decidable.

An access control system is said to be *data-independent* [11] with respect to the *type* of objects if objects can be considered *equal* in serving the post of roles or in accessing any data. This constraint induces a symmetry on objects which implies a bisimilarity on the transition system such that the decidability theory proposed by KN is valid [11]. Take a paper reviewing system as a counterexample. Suppose it is required that an author cannot be the reviewer of his/her own paper. Not all objects are *equal* in serving the *reviewer* role for a given paper because some of the objects are precisely the authors of the paper. Thus, such a system is not data independent.

Nevertheless, this restricted KN problem still lacks an automatic verification method. As a solution, we propose a Satisfiability (SAT)-based verification method [5], that is not only automatic, but also incremental and scalable. The decidability of the KN problem was mainly based on a symbolic reasoning of the infinite state space. Together with the finite symbolic state space and the negation of a universal SATL property, we modeled the KN problem into a bounded model checking (BMC) [3] problem and encoded it into a SAT problem. The BMC version of the KN problem allows early detection of property violation, which is significant for verification feasibility because of the state space explosion problem that becomes even more prominent in access control verification. Another issue in access control policy verification is how to generate the state space. A naive method would be to create the *k*-step state graph explicitly. In contrast, a more efficient method is proposed in this work. The SAT encoding for the BMC problem includes the policy rules, an inheritance mechanism for permissions, and a given universal SATL property. The

state graph is not explicitly constructed beforehand, which is made possible due to the implicit encoding of permission inheritance. As a result, the SAT encoding is much more efficient than explicit state encoding.

The rest of this article is organized as follows. Section 2 reviews the state-of-the-art techniques in verifying access control security systems. Section 3 defines the required terminologies. In Section 4, we describe the proposed SAT-based security system verification (S3V) method. Section 5 analyzes the verification results of two real security access control systems. Finally, we give conclusions and future work in Section 6.

## 2. Related Work

Guelev et al. [8] verified access control systems using model checking, but the systems were restricted to bounded numbers of agents and resources. Later, they proposed the synthesis of access control systems modeled in the RW (where R and W stand for access by Reading and Writing, respectively) language into a standard modeling language called XACML (eXtensible Access Control Markup Language) [4]. Their model checking tool was called Access Control Policy Evaluator and Generator (AcPeg), which requires that the numbers of subjects and objects should be identified before verification execution. Bryans [2] used Communicating Sequential Processes (CSP) to formalize and analyze the access control systems. They showed how the core concept in XACML and the properties of the policies can be represented in CSP.

Since the HRU safety problem is undecidable, the above works were mostly proposed for bounded systems. Kleiner and Newcomb [11] removed this restriction by proving that the safety problems for data independent access control policies can be decidable over a newly introduced first-order linear temporal logic, called Safety Access Temporal Logic (SATL) with only universal quantifier at the outermost level.

Though the work of KN showed that the access control safety problem is decidable and a CSP-based analysis could be used for checking the safety of a set of access control policies, yet it is tedious and error-prone [12]. In contrast, our proposed SAT-based verification works automatically, which is like BMC [3] that verifies a given system in  $k$  steps. The difference between our method and BMC is that we do not generate the whole system state graph before verification. The  $k$ -step graph grows incrementally starting from some initial state as each policy rule is applied (variables assigned by SAT solver). Compared with the work [10] proposed by Hughes and Bultan, they proposed four symbols as combinators and several functions to transform the access control policy described in XACML into Boolean logic formulas so that they can verify if such a combination of XACML policies does or does not faithfully reproduce the properties of its sub-policies, and thus discover unintended consequences before they appear in practice. Such a verification task

is like constraint-checking or condition-checking, which is basically a static analysis, where permissions are unchanged in the problem. Our work is like permission-checking, where the permissions change (obtained or removed) when access control policies are applied; thus our work is a dynamic analysis, where state changes are inherited and checked. Moreover, they need to enumerate the subjects, while our method needs not to do so due to KN's symbolic theory [11].

## 3. Preliminaries

Before introducing our method, we define several terminologies used throughout this work.

**Definition 1: Access Control Model** [11]

An access control system  $S$  is modeled by a 5-tuple  $(\Sigma, O, A, P, C)$ , where

- $\Sigma$  is an infinite universe of objects, and  $O$  is a finite subset of  $\Sigma$ .
- $A$  is a finite set of access rights.  $A = R \cup \bar{R}$ ,  $R \cap \bar{R} = \emptyset$ , where  $R$  is a set of access rights representing roles, and  $\bar{R}$  is a set of access rights for data.
- $P = O \times O \times A$  is called a set of permissions, where a permission representing role is denoted by  $(o, o', a)$ , where  $o = o' \in O$ ,  $a \in R$ ; a non-role permission is represented by  $(o, o', a)$ , where  $o \neq o' \in O$  and  $a \in \bar{R}$ .
- $C$  is an access control policy, represented by a finite set of commands, where a command is a 6-tuple of finite sets;  $c(F) = (c_{on}, c_{off}, c_{create}, c_{grant}, c_{take}, c_{destroy})$ , where  $F$  is a set of formal parameters  $\{x_1, \dots, x_n\}$  that can take values in  $\Sigma$ ,  $c_{on}, c_{off}, c_{grant}, c_{take} \subseteq F \times F \times A$ , and  $c_{create}, c_{destroy} \subseteq F$ .

A *state* is a pair  $(O, P)$ . A *transition* denoted by  $(O, P) \rightarrow (O', P')$  is defined between two states  $(O, P)$  and  $(O', P')$ , iff there exists some command  $c_i \in C$  such that all of the following hold:

- The command  $c_i$  is *applicable* at  $(O, P)$ , which means:
  - $c_{on} \cup c_{off} \subseteq O \times O \times A$ , that is, conditional permissions fall within the scope of the current state.
  - $c_{create} \cap O = \emptyset$ , that is, objects to be created do not exist before.
  - $c_{grant} \cup c_{take} \subseteq O'' \times O'' \times A$ , where  $O'' = O \cup c_{create}$ .
  - $c_{destroy} \subseteq O''$ , that is, objects to be destroyed exist after  $c_{destroy}$  has been applied.
- The *guard* of the command is satisfied if  $c_{on} \subseteq P$  and  $c_{off} \cap P = \emptyset$ .
- The *next state* predicates are satisfied if  $O' = (O \cup c_{create}) \setminus c_{destroy}$  and  $P' = ((P \cup c_{grant}) \setminus c_{take}) \cap (O' \times O' \times A)$ .

□

**Definition 2: Safety Access Temporal Logic (SATL) and Universal SATL**

A *Safety Access Temporal Logic* (SATL) formula  $\phi$  has the following syntax:  $\phi ::= x = y \mid x \neq y \mid (x, y, a) \mid \neg\phi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \phi \Rightarrow \phi \mid \exists x \cdot \phi \mid \forall x \cdot \phi \mid \Box\phi$ , where  $x, y \in F_\phi$ ,  $F_\phi$  is a set of all objects in continual existence that satisfy the safety property  $\phi$ ,  $a \in A$ , and  $\Box$  means *always* or *globally*. *Universal SATL* is a fragment of SATL that only contains formulas with the universal quantifier  $\forall$  which may only occur at the outermost level, i.e., formulas of the form  $\forall x_1, \dots, x_n \cdot \phi$ , where  $\phi$  is quantifier free.  $\square$

**Definition 3: Model Checking Access Control System**

**Problem definition.** Given an access control system  $S$  modeled by  $(\Sigma, O, A, P, C)$  and a universal SATL property  $\phi$ , we need to verify if  $S$  satisfies  $\phi$ , which is denoted as  $S \models \phi$ . Note that this problem is decidable; however, it is a complex problem due to the following reasons:

- Both the system  $S$  and the property  $\phi$  are represented as first-order logic over the set of objects  $\Sigma$ .
- Since  $\Sigma$  is an infinite universe of objects, the system state graph of  $S$  cannot be generated before verification. However, an exhaustive exploration is still required to verify  $S$ .
- The problem does not specify any initial state, thus all possible initial states have to be explored.

**KN Method.** The KN method [11] has proved that the concrete traces in  $\Sigma \times \Sigma \times A$  are equivalent to the abstract traces in  $(F_\phi \cup F_C) \times (F_\phi \cup F_C) \times A$  based on the assumption that  $S$  is *data-independent*, where

- $F_\phi$  is a set of all objects in continual existence that satisfy property  $\phi$ .
- $F_C$  is a subset of  $\Sigma \setminus F_\phi$  of size  $h$ , where  $h$  is the greatest number of formal object parameters in any one command. As we can see,  $(F_\phi \cup F_C) \subseteq O$ .

Moreover, we denote  $A_\phi$  as a set of all access rights in continual existence that satisfy property  $\phi$ . They also proved that the abstract system  $(F_\phi \cup F_C) \times (F_\phi \cup F_C) \times A$  and the concrete system  $\Sigma \times \Sigma \times A$  are indistinguishable by any propositional formula  $\phi$  mentioning only objects in  $F_\phi$ . Therefore, model checking for universal SATL is *decidable*. Nevertheless, this restricted KN problem still lacks an automatic verification method. As a solution, we propose a Satisfiability Theory (SAT)-based verification method, that is not only automatic, but also incremental and scalable.

## 4. SAT-based Security System Verification (S3V)

The target problem of model checking access control system in Definition 3 can be solved incrementally by checking for violation of the property  $\phi$  in a bounded number of steps. Given an access control system model  $S = (\Sigma, O, A, P, C)$  and a universal SATL property  $\phi$ , the bounded model checking problem checks if  $S$  violates  $\phi$  in  $k$  steps, i.e.,  $S \not\models_k \phi$ . The target model checking problem can thus be defined as  $S \models_k \neg\phi, \forall k, 1 \leq k \leq k_{max}$ , where

$k_{max}$  will be made explicit in Section 4.2. This BMC version of the target problem can be solved using a SAT solver. Thus we first propose a symbolic encoding method for the BMC problem in Section 4.1, and then analyze the complexity of the S3V method in Section 4.2.

### 4.1 SAT Encoding for the BMC Problem

Given an access control system  $S$  modeled by  $(\Sigma, O, A, P, C)$ , where  $C$  has  $w$  commands, and a command is a 6-tuple of finite sets;  $c(F) = (c_{on}, c_{off}, c_{grant}, c_{take}, c_{create}, c_{destroy})$ , we first introduce how to encode a set of access control policies in an access control system as follows.

Since the actions  $c_{create}$  and  $c_{destroy}$  do not affect the verification result, we do not need to encode them. More specifically,  $c_{create}$  is used for creating new objects that do not possess any permission, and  $c_{destroy}$  is used for destroying some objects, and thus no longer exist (i.e., nonexistent people have no permission). Therefore, both  $c_{create}$  and  $c_{destroy}$  do not affect the permissions that objects may have or not have, and thus they will not affect the verification result that is relevant only to the permissions that objects have or do not have.

Given a finite set  $\Psi$  of elements and  $\delta \in \mathbb{N}$ , the set of natural numbers, let  $Perm(\Psi, \delta)$  denote a *permutation* function of  $\delta$  distinct elements from the set  $\Psi$ . Note that  $Perm(\Psi, \delta)$  is complete in the sense that all permutations of  $\delta$  elements from  $\Psi$  can be found in it. Let  $c_{ij}$  be an independent variable representing that command  $c_i$  is applied at step  $j$ ,  $1 \leq j \leq k$ ,  $\varrho \in \mathcal{D}_i = Perm((F_\phi \cup F_C), n_i)$  represent a unique permutation of  $n_i$  objects from  $F_\phi \cup F_C$ , where  $n_i$  is the number of parameters in command  $c_i$ ,  $B_{i,\varrho} = \{Perm(\varrho, 2)\}$ , for some  $\varrho \in \mathcal{D}_i$ . The notation  $\boxplus$  is the operator for *mutual exclusion*.

The encoding of an access control system  $S$  for  $k$  steps, consists of four parts as follows.

- **Mutual exclusive (ME) application of commands:** This part ensures that (1) only one command is applied at each step, and (2) only one permutation  $\varrho$  of the  $n_i$  objects from  $F_\phi \cup F_C$  is chosen for the application of command  $c_i$  at step  $j$ . ME is encoded as quantification for the following 3 parts:  $(\boxplus_{i \in \{1,2,\dots,w\}}(c_{ij}) \wedge (\boxplus_{\varrho \in \mathcal{D}_i}(c_{ij}, \varrho)))$ , at step  $j = 1$  to  $k$ .
- **Applicability checking (AC):** This part is used to check if some command is applicable. In other words, this is like a pre-condition checking.  $AC = (\bigwedge_{a \in A, b \in B_{i,\varrho}, (b,a) \in (c_i).on} (b, a, j)) \wedge (\bigwedge_{a \in A, b \in B_{i,\varrho}, (b,a) \in (c_i).off} \neg(b, a, j))$ , for some command  $i \in \{1, 2, \dots, w\}$  at some step  $j \in \{1, \dots, k\}$ .
- **Permission inheritance (PI):** This part is used to inherit the permissions from the previous command applications.  $PI = \bigwedge_{p \in (P \setminus (c_i.take) \setminus (c_i.grant))} (\neg((p, j) \oplus (p, j + 1)))$ , for some command  $i \in \{1, 2, \dots, w\}$

at some step  $j \in \{1, \dots, k\}$ . Note that  $\oplus$  means "exclusive or" (XOR).

- Permission changing ( $PC$ ): This part is obtained from the application of the commands  $c_{grant}$  and  $c_{take}$ . As we described above, only these two kinds of commands can change the permissions.  $PC = (\bigwedge_{a \in A, b \in B_{i, \varrho}, (b, a) \in (c_i).grant} (b, a, j + 1)) \wedge (\bigwedge_{a \in A, b \in B_{i, \varrho}, (b, a) \in (c_i).take} \neg(b, a, j + 1))$ , for some command  $i \in \{1, 2, \dots, w\}$  at some step  $j \in \{1, \dots, k\}$ .

Finally, we encode  $S = \bigwedge_{j=1}^k (\boxplus_{i \in \{1, 2, \dots, w\}} (c_{ij}) \wedge (\boxplus_{\varrho \in \varnothing_i} ((c_{ij}, \varrho) \wedge AC \wedge PI \wedge PC)))$ .  $\square$

To identify the initial state of the verification space, we need to initialize the roles for all the objects  $o \in (F_\phi \cup F_C)$ .  $Init = \bigwedge_{o \in (F_\phi \cup F_C)} (\bigvee_{r \in R} (o, o, r, 0))$ .  $\square$

Note that we denote a permission at step  $j$  as  $(b, a, j)$ , where  $b \in Perm(F_\phi \cup F_C, 2)$ ,  $a \in A$ . If such a permission  $(b, a, j)$  is true, we use the notation  $(b, a, j, 1)$ , otherwise  $(b, a, j, 0)$ .

Given a universal SATL property  $\phi$ , we encode the negation of the property ( $\neg\phi$ ) as follows.

- State formula: A state formula needs to be verified against the initial state. There are two types of state formulae. The first type is  $\neg\phi_1 = \forall x \in O, \phi'_1$ . We need to encode  $\phi'_1$ , which is quantifier free for all the objects  $o \in (F_{\phi_1} \cup F_C)$  as follows. Thus, this property is encoded as  $E_1 = \bigvee_{a_1, b_1} (b_1, a_1, 0)$ , for all  $a_1 \in A_{\phi_1}$ , for all  $b_1 \in (F_{\phi_1} \cup F_C)$ . Note that  $b_1$  is used to represent  $\forall x \in O$ . Second, we assume the property  $\neg\phi_2 = \forall x, y \in O, \phi''_2$ . We need to encode  $\phi''_2$ , which is quantifier free for all the objects  $o \in (F_{\phi_2} \cup F_C)$  as follows. Thus, this property is encoded as  $E_2 = \bigvee_{a_2, b_2} (b_2, a_2, 0)$ , for all  $a_2 \in A_{\phi_2}$ , for all  $b_2 \in Perm(F_{\phi_2} \cup F_C, 2)$ . Here  $b_2$  is used to represent  $\forall x, y \in O$ .
- Path formula: The property is of the type  $\forall x, y \in O, \square\phi$ .  $E^k = \bigvee_{a, b} ((b, a, 0) \wedge (b, a, k))$ , for all  $a \in A_\phi$ , for all  $b \in Perm(F_\phi \cup F_C, 2)$ . The reason for encoding at initial state and the step  $k$  is because that our method can guarantee that the property is satisfied at all states from step 1 to  $k - 1$  for a verification bound step  $k$ , thus we only need to verify all states at step  $k$  and the initial state; however, we need to specify all the objects  $o \in (F_\phi \cup F_C)$  by using the symbol  $b$ .

Based on the encoding formula described above, we can finally encode the input problem formula  $I_k$  of SAT as follows.  $I_k = Init \wedge S^k \wedge (E^k \vee E)$ . Given a problem encoding  $I_k$ , a SAT solver assigns true or false to some of the literals  $(c_{ij}, (c_{ij}, \varrho))$  and permissions  $(b, a, j)$  in  $I_k$  such that  $I_k$  is true.

## 4.2 S3V Complexity Analysis

As discussed earlier, there should be a maximum finite verification bound that is required for termination of the

S3V method. This maximum bound can be estimated using the total number of states in a security system because at least one new state is checked for each increment of the verification bound. Thus, we need to estimate the total number of states, i.e., the verification state space size. Further, we also need to estimate the SAT problem size and the complexity of our proposed S3V method.

- Verification state space: the upper bound of the total number of states,  $k_{max}$  is estimated as  $2^{|F_\phi \cup F_C| \times |R|} \times 2^{P_2^{|F_\phi \cup F_C| \times |R|}}$ . Note  $P_v^u$  is the number of permutations of  $v$  elements from  $u$  elements. The first term,  $2^{|F_\phi \cup F_C| \times |R|}$ , represents the total number of combinations of roles for all the objects in  $F_\phi \cup F_C$ . The second term,  $2^{P_2^{|F_\phi \cup F_C| \times |R|}}$ , represents the number of combinations of the non-role permissions, i.e.,  $\{(o, o', a) | o \neq o' \in F_\phi \cup F_C, a \in \bar{R}\}$ .
- The size of the SAT problem  $S \stackrel{?}{=} \neg\phi$  can be estimated by the number of literals and clauses for  $Init$ ,  $S^k$  and  $E^k$ , where  $n_i = |F_{c_i}|$  and  $w = |C|$ .
  - $Init$ : There are  $(|R| \times |F_\phi \cup F_C|)$  literals, and  $|F_\phi \cup F_C|$  clauses.
  - $S^k$ : There are  $\{(w \times k) + (w \times k \times P_{n_i}^{|F_\phi \cup F_C|}) + (4|P| \times w \times k) + (P_2^{|F_\phi \cup F_C|} \times |A| \times k)\}$  literals, and  $\{((2^w - w) \times k) + ((2^{P_{n_i}^{|F_\phi \cup F_C|}} - P_{n_i}^{|F_\phi \cup F_C|}) \times w \times k) + (2|P| \times w \times k) + (P_2^{|F_\phi \cup F_C|} \times |A| \times k)\}$  clauses.
  - $E^k$ : The number of literals and the number of clauses in  $E^k$  depend on the form of the given property. For example, a given property  $\phi ::= (o, o, r_1)$  represents that some object  $o$  has some role  $r_1 \in R$ . The number of literals and clauses of  $E^k$  are both 1. If the given property is  $\phi ::= \forall x, y, (x, y, a_1)$ , where  $x, y \in O, x \neq y$ , some  $a_1 \in A$ , then the number of literals of  $E^k$  are both  $P_2^{|F_\phi \cup F_C|}$ , and the number of clauses is 1.

Without encoding permission inheritance, the verification space size is  $2^{|P|} \times \frac{w^{k+1}-1}{w-1}$ , where  $|P|$  is the cardinality of the set of permissions. Comparing the verification state space size of the S3V method with that of no permission inheritance encoding, the verification state space size of the S3V method is much smaller. The number of literals and clauses are also much smaller. By this empirical analysis, we can conclude that the proposed S3V method is much more efficient.

## 5. Experimental Results

We use two examples, namely Employee Information System (EIS) and Nuclear Power Plant Security (NPPS) System to illustrate the feasibility and benefits of the proposed S3V method, which was implemented using the C programming language. The SAT solver Limmat [1] was used in S3V. The experiments were performed on a PC running Linux 2.6.26

Table 1: Protection matrix of employee information system.

	on	off	create	grant	take	destroy
$c_1(x, y)$	$(x, x, D)$			$(x, y, B)$		
$c_2(x, y)$	$(x, x, D)$				$(x, y, B)$	
$c_3(x, y)$	$(x, x, M)$	$(y, y, M)$ $(y, y, D)$		$(x, y, B)$		
$c_4(x, y)$	$(x, x, M)$	$(y, y, M)$ $(y, y, D)$			$(x, y, B)$	
$c_5(x, y)$	$(x, x, D)$	$(y, y, M)$		$(y, y, M)$		
$c_6(x, y)$	$(x, x, D)$ $(y, y, M)$				$(y, y, M)$	
$c_7(x, y)$	$(x, x, M)$		$y$			
$c_8(x, y)$	$(x, x, M)$	$(y, y, M)$ $(y, y, D)$				$y$

**D:** Director **M:** Manager **B:** Bonus

operating system, with a 64-bit Intel Core i7 2.67 GHz and 6 GB of physical memory.

## 5.1 Employee Information System (EIS)

The EIS system  $S$  is modeled as  $(\Sigma, O, A, P, C)$ , where  $A = \{D(Director), M(Manager), B(Bonus)\}$ ,  $C = \{c_1, \dots, c_8\}$ ,  $|F_\phi| = 2$ ,  $|F_C| = 2$ , and  $|F_\phi \cup F_C| = 4$ . Table 1 shows the protection matrix of EIS.

The command  $c_1(x, y)$  states that object  $x$  is granted permission to give *bonus* to object  $y$  if  $x$  is a  $D(Director)$ . Other commands can be similarly interpreted. A universal SATL property was specified as:  $\forall x, y. ((x, x, Manager) \wedge (y, y, Manager) \wedge \neg(x, y, Bonus) \wedge \neg(y, x, Bonus) \rightarrow \Box \neg((x, y, Bonus) \vee (y, x, Bonus)))$ , which means "Can two managers conspire such that one of them gives a bonus to the other?". The system model and the property were encoded using our proposed method into a SAT problem with 275 literals and 19 clauses. Without the proposed encoding for generalized permission inheritance, one would have to explicitly encode each and every state and transition.

Applying S3V, we found a counterexample at step  $k = 2$ , as follows. Initially object  $a$  is a director,  $b$  and  $c$  are both managers. Applying command  $c_6$ ,  $a$  demotes  $b$  from the post of manager. Applying command  $c_3$ ,  $c$  is granted permission to give bonus to  $b$ . The counterexample was found in 0.23 seconds using 1240 KB memory. Instead of using our S3V method, if we use a conventional unbounded model checking procedure, this example requires checking the full state graph with 1048576 ( $k_{max}$ ) states. Compared to the conventional method, S3V is much more efficient as it checks only 2112 states. Moreover, S3V is more scalable than conventional methods because of the symbolic processing based on the KN theory. If our work is not based on KN, we need to enumerate all the objects. For example, if the number of objects is  $n$ , then the size of the explicit verification state space is  $2^{n^2}$ , which means that it grows

exponentially as shown in Figure 1.

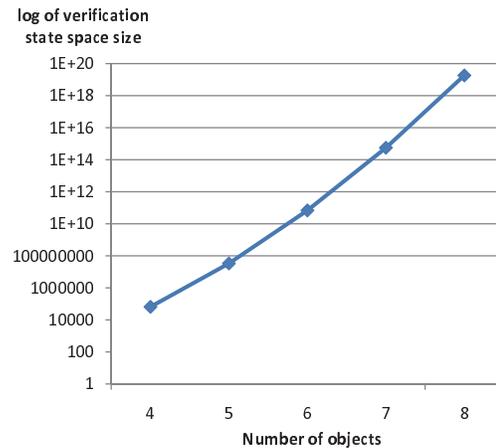


Fig. 1: EIS verification state space size without KN.

## 5.2 Nuclear Power Plant Security (NPPS)

*Nuclear power plants* (NPP) are critical infrastructures that require stringent security control to protect the perimeter of vital operations. However, due to failure of digital instruments such as sensors within the perimeter, technicians must be granted access with an escort personnel such as an expert in radioactivity. The NPP security (NPPS) policy states that only when a protected sector enters the criticality level of *warning*, can a personnel be granted the *escort* permission on-the-fly. A technician is then granted access (*entry*) to the perimeter of vital operations with the help of an escort. This is called the *2-man rule* and can be realized using two RFID tags that must be both read within a pre-defined period of time. The protection matrix is shown in Table 2. Commands  $c_1, \dots, c_5$  define the 2-man rule,  $c_6, \dots, c_{11}$  define the change in criticality level of a protected sector, and  $c_{12}, c_{13}$  constitute the create and destroy actions.

Table 2: Protection matrix for nuclear power plant security system.

	on	off	create	grant	take	destroy
$c_1(x, y)$	( $x, x$ , Normal)				( $y, y$ , Escort)	
$c_2(x, y)$	( $x, x$ , Warning)			( $y, y$ , Escort)		
$c_3(x, y)$	( $x, x$ , Danger)				( $y, y$ , Escort)	
$c_4(x, y)$	( $x, x$ , Warning) ( $y, y$ , Escort)			( $x, y$ , Entry)		
$c_5(x, y)$	( $y, y$ , Escort)				( $x, y$ , Entry)	
$c_6(x, y)$	( $x, x$ , Normal)			( $x, x$ , Warning)	( $x, x$ , Normal)	
$c_7(x, y)$	( $x, x$ , Normal)			( $x, x$ , Danger)	( $x, x$ , Normal)	
$c_8(x, y)$	( $x, x$ , Warning)			( $x, x$ , Normal)	( $x, x$ , Warning)	
$c_9(x, y)$	( $x, x$ , Warning)			( $x, x$ , Danger)	( $x, x$ , Warning)	
$c_{10}(x, y)$	( $x, x$ , Danger)			( $x, x$ , Normal)	( $x, x$ , Danger)	
$c_{11}(x, y)$	( $x, x$ , Danger)			( $x, x$ , Warning)	( $x, x$ , Danger)	
$c_{12}(x, y)$	( $x, x$ , Normal)		$y$			
$c_{13}(x, y)$	( $x, x$ , Normal)					$y$

It is required to check if the 2-man rule is *safe*, where safety requires that a technician who is granted entry can come out again from the protected sector. The corresponding universal SATL property is given as  $\phi : \forall x, y, \neg(x, x, danger) \vee (y, y, escort) \vee \neg(x, y, entry)$ .

The protection matrix and the negated property were encoded using our proposed method into a SAT problem, which has 959 literals and 26 clauses. The property is violated at step  $k = 5$ , where the sector  $x$  has already reached the *danger* level of criticality, the personnel  $y$  is no longer an *escort*, however, the technician is still in the protected sector and can no longer leave it (since the original escort is no longer an escort). The verification took 8.22 seconds, and used 1344 KB memory. Instead of using our S3V method, if we use a conventional unbounded model checking procedure, this example requires checking the full state graph with  $2^{28}$  ( $k_{max}$ ) states.

## 6. Conclusions and Future Work

We proposed a SAT-based verification method for formally verifying access control systems that are specified by a protection matrix. We have shown that the proposed S3V method is much more efficient than a non-permission-inheritance encoding method due to the smaller problem size and it is also more scalable. Two examples on employee information system (EIS) and nuclear power plant security (NPPS) system show the feasibility of our S3V. We are now working on extending the S3V method to data-dependent access control systems, multilevel access control security systems, and role-based access control systems. We believe that this method is scalable in the sense that new security models such as the conventional role-based access control or the contemporary distributed security solutions for cloud computing services can be easily incorporated and extended to be Satisfiability-Modulo Theory (SMT)-based [6].

## References

- [1] A. Biere. The evolution from limmat to nanosat. Technical Report 444, Department of Computer Science, ETH Zurich, 2004.
- [2] J. Bryans. Reasoning about XACML policies using CSP. Technical Report CS-TR-924, University of Newcastle, July 2005.
- [3] E. M. Clarke, A. Biere, R. Raimi, and Y. Zhu. Bounded model checking using satisfiability solving. *Formal Methods in System Design*, 19:7–34, July 2001.
- [4] OASIS committee. eXtensible Access Control Markup Language TC v2.0 (XACML). *OASIS Standard*, 2005.
- [5] M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7:201–215, 1960.
- [6] L. de Moura, B. Dutertre, and N. Shankar. A tutorial on satisfiability modulo theories. In *Proceedings of the 19th International Conference on Computer Aided Verification*, volume 4590/2007 of *Lecture Notes in Computer Science*, pages 20–36. Springer-Verlag, July 2007.
- [7] G. S. Graham and P. J. Denning. Protection – principles and practice. In *Proceedings of the May 16-18, 1972, Spring Joint Computer Conference*, pages 417–429. ACM Press, May 1972.
- [8] D. P. Guelev, M. Ryan, and P. Yves Schobbens. Model-checking access control policies. In *Proceedings of the International Information Security Conference*, volume 3225/2004, pages 219–230. Springer-Verlag, June 2004.
- [9] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman. Protection in operating systems. *Communication ACM*, 19(8):461–471, August 1976.
- [10] G. Hughes and T. Bultan. Automated verification of access control policies using a SAT solver. *International Journal on Software Tools Technology Transfer*, 10:503–520, October 2008.
- [11] E. Kleiner and T. Newcomb. On the decidability of the safety problem for access control policies. In *Proceedings of the Sixth International Workshop on Automated Verification of Critical Systems*, pages 91–103. Elsevier Science Publishers, September 2006.
- [12] E. Kleiner and T. Newcomb. Using CSP to decide safety problems for access control policies. Technical Report RR-06-04, Oxford University Computing Laboratory, January 2006.
- [13] B. Lampson. Dynamic protection structures. In *Proceedings of the November 18-20, 1969, Fall Joint Computer Conference*, pages 27–38. ACM Press, November 1969.
- [14] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.
- [15] W. Stallings and L. Brown. *Computer Security – Principles and Practice*. Pearson Education, 2008.

# Security of the Social Network Site User

Amina Kinane Daouadji<sup>1</sup>, and Sadika Selka<sup>2</sup>

Department of Computer Science, Mohamed Boudiaf University, USTO, Oran, Algeria

**Abstract** - The objective of our method is to guarantee a security, for the social networks site user. Our method makes it possible to classify the social networks site user in classes (class entrusting, and class of threatens ), according to criteria's of confidences, by using the artificial immune networks (AIN). It also makes it possible to warn the user of nature entrusting or threatens of these contacts list in the social network site.

**Keywords:** *Social network, access control, classification, artificial immune networks (AIN).*

## 1 Introduction

The social networks hold a very important place in our life, particularly facebook and twiter, many people integrates these sites in their daily practices. This phenomenon exploded throughout the world, by creating an enormous problem: security. Several methods were carried out to solve this problem, but these methods are based on the site access control [3], [4], and the personal data access [6], [7], our method is carried out to control the access of the social networks users, by a classification based on the artificial immune networks, our algorithm classify the users into tow cluster, one for the trustful users and the second for the users who constitute threat, according to criteria's of confidence selected.

The system will allot to each criterion of confidence selected, a binary value (1 or 0), or a real value (percentage) calculated according to a mathematical formula. An unsupervised artificial immune network will treat these values, to achieve a classification, which allow the social network site user to know the nature of these contacts.

## 2 Social Network

There are several definitions for the social networks:

According to Giles Hogben "a social network consists of a unit finished actors, with the relations defined between them. An actor can be only one person or a group of people. The actors in a social network are bound by relations. The type and the degree of confidence of these relations can vary dependently actors implied. The friends, the family, or the colleagues are as many examples of the types of relations" [1]. and according to Dajana Kapusova Leconte "a social network is a structure formed by relations between people. This social structure made up of nodes, is generally represented by individuals or organizations. The nodes are connected between them by various social knowledge which can go from a simple knowledge until a family bond very extremely" [2].

## 3 Threaten on the social networks

This explosive phenomenon of social networks, became unverifiable, this explosion caused great menaces, various threats comes from the confidence which the users place in the various networks social, the raison to be a social networks is sharing, unfortunately, many users put sensitive information on their subjects like secrets of work and personal photos, that generates also the problem of identity usurpation, there is also the problem of worms, Trojan horses, the problem of the shortened bonds, another type which represents attacks of the type Cross-Site Request Forgery, all these risks lead to the problem of security.

## 4 Methods related to security

Several methods were proposed to cure various problems encountered during exploitation of the social networks, generally, the researchers think to reinforce the control on the social network access and on the data access such as the work of Filipe Beato "Enforcing Access control in social network sites" this work consists in reinforcing the access control in a social network by using cryptographic techniques, more particularly by using public keys shared when two users want to establish a connection [3]. Another idea which is also based on the network access is achieved in the work of Admin Tootoochian, it is a protocol named Lockr, it introduces two concepts, one is for the social attestation and the other for the lists of the social network access control [4], there is also work of Michael Hart, "More Content - Less Control: Access Control in the Web 2.0" this method comprises two needs, a specification of policy usable and an automatic application of the policy, it can be used by any users, and can support the dynamic contents of the blogs, social networks, and other sites with partition of contents [5]. There's also the work of Anna "Collective Privacy Management in Social Networks" The authors of this method have proposed a solution that provides automated means to share images based on an extended notion of ownership of the content. In being pressed on the mechanism of Clarke-Tax, they proposed a simple mechanism which supports the veracity, and which rewards the users who make promotion joint ownership. They incorporated a design inference technique which frees the user from the manually burden of selecting privacy preferences for each picture [6]. And finally the work of Barbara Carminati and Elena Ferrari "Privacy-Aware Collaborative Access Control in Web-Based Social Networks" in this method the authors proposed a solution to respect the privately by mechanisms of access control, able to carry out a division, to control the resources and in the same time, to meet the requirements for confidentiality of the social networks users compared to their

relations by using access rule, they applied an access control by a collaboration of nodes selected in the network; according to the type of relation either friends or colleague therefore the distance and the degree of confidence [7].

## 5 Solution suggested

Each user subscribed in the site, behaves in a different way compared to the other users, our system must classify the users of the social site according to criteria's of confidence selected. These criteria make possible to create two distinct classes; one represent confidence users and another class represent the threat users, by using an artificial immune network. The artificial immune system is the composition of intelligent methods based on metaphors of the natural immune system, principles and models will be applied to solve real world problems[8], [9].

The theory of the immune network is proposed by Jerne in 1974, it suggested that the interactions within the immune system are not limited between antibody and antigens, but also between the antibodies. Therefore we can define an artificial immune network (AIN) as a data-processing model bio-inspired which employs the ideas and the concepts of the theory of the immune networks, mainly the interactions between the B-cells (stimulation and suppression), and the process of cloning and mutation.

### 5.1 Criteria proposed

For the criteria of confidence or menace we thought of extracting them from the behavior of the user in the network, (activity, the behavior with his friends, information and picture sharing degree ...), and from the personal information entered during the creation of its account like the name and first name, geographic area... etc. We chose criteria extracted during and after the account creation:

#### 5.1.1 Criteria extracted during account creation

The criteria extracted during account creation are: the name and first name, the geographic area and the pseudonym. The pseudonym field does not exist in the current social sites but we'll add it to let the user choose either fill in name and first name, or fill out the pseudonym.

#### 5.1.2 Criteria extracted after account creation

Criteria extracted after account creation are:

##### 5.1.2.1 The activity

This criterion will enable us to know if a given user is active or not, in other term if he frequently accede to his account or rarely.

##### 5.1.2.2 Reputation

The field reputation does not exist in the current social sites. We will assume that in our social site there is a field where each user must give notice of (good reputation or bad reputation) on each friend.

#### 5.1.2.3 The documents sharing degree

After the user subscription, we can know if he shares his documents with the other friends or no.

#### 5.1.2.4 The number of invitation accepted

After the user subscription, we can count the number of invitations accepted (by the other users of the network) and compared to the number of invitations sent.

### 5.2 The representation of the criteria

An application must assign each criterion a binary value (1 or 0), or a percentage calculated at real time. So each user is presented in the following form:

User (user ID (name and first name or pseudonym); Area; Activity rate; sharing rate; reputation; invitation).

The percentage is a calculated value, which always depends on the behavior of the user. Percentage calculation is almost similar to each criterion proposed. In this article we will present the percentage calculation of the reputation criterion:

**R<sub>a</sub>**: is the number of friends which mentioned a good reputation on user **i**;

**T<sub>a</sub>**: is the total number of the friends of user **i** ;

**P**: is the calculated percentage of reputation degree.

$$P = (R_a * 100) / T_a \tag{1}$$

If the value of (P) is more than or equal to 50%, the user give a good impression on him in the site, if the value of (P) is less than 50%, then the user give a bad reputation on him in the site.

Tables 1 illustrate the representation of selected values to each criterion.

**Table 1.** Values of the criteria proposed

The criteria	Value suitable
Name & first Name	1
Pseudonym	0
Area	1 (true) or 0 (false)
Activity rate	Percentage %
sharing rate	1 (if he share) or 0 ( if he these not share )
Reputation	Percentage %
Invitation	Percentage %

**Exemple1.** A(hmed) is a user of a social network, when he created his account he used his name and first name, mentioned his true area, he frequently access the site, he shares its information with his friends, he have a good reputation with his friends, and its invitation request was accepted, therefore Ahmed is regarded as entrusting user, his representation is the following one: User A: (1, 1, 0.80, 1,

0.90, 1.00), or User A:( 1, 1, 1.00, 1, 1.00, 1.00), or User A:( 1, 1, 1.00, 1, 0.90, 1.00) .

**Example2.** L(eila) is a user of the same social network, when she created her account she used her pseudonym, and she mentioned a false area, she frequently access the site, she does not share her personal information, the majority of his friends mentioned a negative opinion about her, and half of her invitation were refused, therefore Leila is regarded as threaten user and his representation is the following one: User L: (0, 0, 0.70, 0, 0.30, 0.50), or User L:( 0, 0, 1.00, 0, 0.20, 0.10), or User L:( 0, 0, 0.40, 0, 0.20, 0.30) .

### 6 Structure of the system suggested

An artificial immune network will treat the values of the criteria used, for carried out a classification and give as results the nature (entrusting or threatens) of each user subscribed on the site; figure 1 illustrate the architecture of our system.

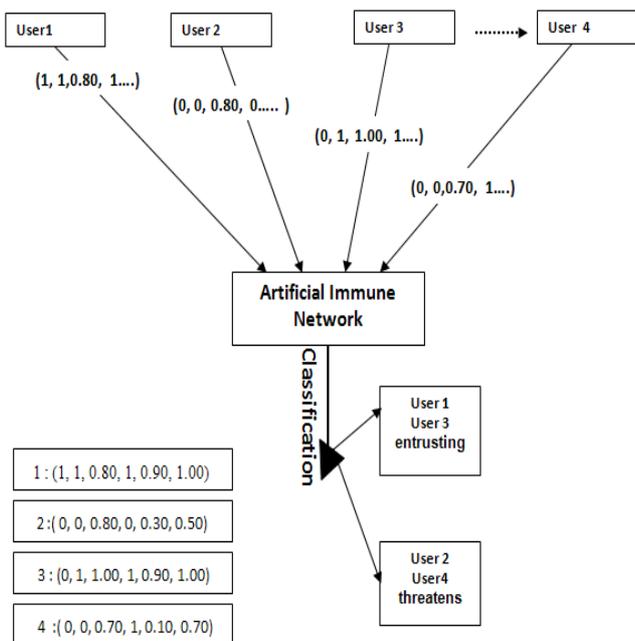


Fig. 1. Architecture of our system

### 7 The algorithm of the artificial immune network unsupervised

**• Initialization**

To initialize the whole of the ARB, Artificial Recognition Ball (Antibodies) by a tiny number of the whole of the individuals (~ 5%). The remainder of the individuals represents the Antigens (~ 95%).

**• Stimulation**

1. To calculate affinity between each ARB and all the whole of Antigens:  $ARB \rightarrow Ag \text{ ps} = 1 - \text{dis}(p) \Rightarrow \text{antigenic reaction}$  .

2. To calculate affinity between ARB cells (neighborhood):  $ARB \rightarrow ARB \text{ ns} = \Sigma \text{dis}(x) \Rightarrow \text{idiotopic reaction}$

**• Cloning**

1. Antigen not recognized ==> primary response: Expansion of the repertory (to integrate a new ARB into the whole of the ARB).
2. Antigen recognized ==> secondary response: cloning of the ARB cells.

**• Resource allocation**

Reduction:  $R_{\text{decayed}} = R_{\text{current}} \times \text{decay\_Rate}$   
 New degree of resources:  $R_{\text{new}} = R_{\text{current}} + (k \times (\text{maxres} - R_{\text{decayed}}) \times S)$

**• Selective Suppression**

To eliminate the ARB which have resources value lower than a threshold and those which resembles each other (Great affinity in the idiotopic reaction).

**Parameters of the algorithm:**

1. Level of stimulation (s).
2. Proximity at the maximum level of resources (maxres - Rdecayed) .
3. Scalar Size (k) (very small).

### 8 Processing of data

Each user i: is represented by six criteria, User i (user ID (name and first name or pseudonym); Area; Activity rate; sharing rate; reputation; invitation), table 2 illustrates the representation of a sample of our database, we used a representation of 100 users, 50 trustful users and 50 threats users .

Table 2. sample of our databases

User 0	1	1	1.00	1	1.00	1.00
User 1	0	0	0.80	0	0.10	1.00
User 4	0	0	0.70	0	0.20	1.00
User 9	1	1	0.90	1	0.90	1.00
User 10	0	1	1.00	1	1.00	1.00
User 26	1	1	0.80	1	1.00	0.90

### 8.1 Results

After the treatment, the algorithm generates 8 ARB; Each ARB has recognized a group of trustful individuals or threat. Table 3 shows the different ARB generated with the class type of each user group:

**Table 3.** Representation of the different ARB generated

ARB Value	Users recognized	The type of class	Recognition	Error rate
0, 2, 5, 7	0, 9, 10, 15, 20, 22, 26, 34, 37, 38,46, 48, 50, 53, 66, 23, 25, 28, 29, 40, 41, 45, 52, 58, 59, 67,69, 71, 27, 54, 70, 21, 35, 36, 39, 47, 49, 51,60, 64, 65, 68, 72, 74, 75, 76, 77, 78, 98, 99	Trustful	100%	0%
1, 3, 4, 6	5, 7, 8, 11,13, 14, 17, 18, 31,33, 43, 44, 55,57, 61, 63,79, 83, 85, 86, 88, 91, 94, 95, 97, 1, 2, 3, 4, 19, 24, 30, 32, 42, 73, 80, 81, 82, 89, 90, 93, 96, 6, 56, 62, 12, 16, 84, 87,92	Menace (threat)	100%	0%

**8.2 Test**

To verify the effectiveness of our system we will test a new user group, and verify the results.

**Table 4.** shows a sample of group of users who do not belong to the database used. To retest the system performed.

User	1	1	0.70	1	1.00	1.00
User 0	1	1	0.70	1	1.00	1.00
User 1	0	0	0.60	0	0.20	1.00
User 2	0	0	0.80	0	0.10	0.70
User 3	0	0	0.40	0	0.30	1.00
User 4	0	0	0.60	0	0.20	1.00
User 5	1	0	0.90	1	0.90	1.00
User 6	0	1	1.00	1	1.00	0.90
User 7	0	0	0.90	0	1.00	0.00
User 8	1	1	0.70	1	0.80	0.70
User 9	1	1	0.90	1	0.90	0.90

Users (0, 5, 6, 8, and 9) are trusted users and users (1, 2, 3, 4, and 7) represent users of threat.

**Table 5.** Shows the results of tests

Users tested	Recognized by ARB number :
User (1, 2, 3, 4)	ARB N° (3)
User (0, 5, 6, 9)	ARB N° (0)
User 7	ARB N° (1)
User 8	ARB N° (7)

According to the results obtained, we can notice that users (1, 2, 3, and 4) were recognized by the ARB N°. (3), which represents the users of class threat. Users (0, 5, 6 and 9) were recognized by the ARB N°. (0), which represents the users of class confidence. User 7 is recognized by the ARB N°. (1), which represents the users of class threat, and the user 8 is recognized by the ARB N°. (7), which represents the users of class confidence.

Each user of the class Trustful must know the nature of these friends and his new contacts.

**• New contact:**

Supposing that A(hmed) is a user in class of confidence and A(li) is a user in class of threatens, A(hmed) wants to invite A(li) to be his friend, the system will warn A(hmed) that A(li) can represented a threat to him, and if he always wants invites him.

If A(hmed) always authorizes the invitation of A(li), the system must save the hour and the date, and the confirmation of Ahmed.

**• A friend in a contact list:**

If A(Li) is already a friend of A(hmed), the system must warn A(hmed) that A(li) can represented a threat to him, and ask him if he always wants to contact A(li) or removed him of his contact list.

**Our system is carried out for:**

1. To classify the users in two classes (confidence and threat).
2. To warn a user of class confidence, the nature of its contacts.
3. Save all the confirmations of acceptance.

If one day a user tackles the social network site in justice, for a contact of class threatens, the social network can defend itself by the information it has already saved.

**9 Conclusion**

Our method is a continuation of the methods which were already proposed; nevertheless the preceding methods based

on the degree of confidence and the type of relation, our method is based on confidence criteria chosen, for carried out a classification intended for the user of the social network so that it can know the nature (trustful or threat) of his contact list. Our results always remain within the experimental framework, it's the theory which should be applied to a real social network in order to detect the gaps of our method, the criteria used can change or to be insufficient in a real social site.

## References

- [1] Giles Hogben & ENISA, “security issues and recommendations for online social networks”, ENISA Report, Greece, [http://www.enisa.europa.eu/doc/pdf/deliverables/enisa\\_pp\\_social\\_networks .pdf](http://www.enisa.europa.eu/doc/pdf/deliverables/enisa_pp_social_networks.pdf), 2007.
- [2] Dajana Kapusova Leconte “développement d’un logiciel de réseau social comme soutien a une communauté de pratique”, Mémoire présenté pour l’obtention du DESS STAF Sciences et Technologies de l’Apprentissage et de la Formation TECFA, Juin 2008.
- [3] Filipe Beato, Markulf kohlweiss etKarel Wouters: “Enforcing Access control” in social Inetworksites,,<http://www.cosicesat.kuleuven.be/publication/article1240.pdf>, 2009.
- [4] Amin Tootoochian, Kiran K. Gollu, Stefan Saroiu, Yashar Ganjali, & Alec Wolman, Lockr : “SocialAccess Control for Web 2.0”, Proceedings ofthe First ACM SIGCOMM Workshop on Online SocialNetworks (WOSN), Seattle,WA, USA,,[http://www.cs.standrews.ac.uk/~trisan/sigcomm08/workshops/wosn/papers/p43. pdf](http://www.cs.standrews.ac.uk/~trisan/sigcomm08/workshops/wosn/papers/p43.pdf), 2008.
- [5]: Michael Hart, Rob Johnson, & Amanda Stent, “More Content – Less Control: Access Control in the Web 2.0”, Seventh International Workshop on Software and Performance (WOSP'08),ACM,NJ,USA.[http://www.cs.stonybrook.edu/~rob/papers/cbac\\_w2sp07.pdf](http://www.cs.stonybrook.edu/~rob/papers/cbac_w2sp07.pdf), 2008.
- [6] Anna C. Squicciarini et Mohamed Shehab et Federica Paci “ Collective Privacy Management in Social Networks ” Track: Security and Privacy / Session: Web Privacy, WWW MADRID, p521, 2009.
- [7] Barbara Carminati and Elena Ferrari “ Privacy-Aware Collaborative Access Control in Web-Based Social Networks ” , University of Insubria 22100 Varese, Italy , 2009.
- [8] Timmis J. et de Castro L. N., “ Artificial Immune System, A new computational intelligence Approach”, ISBN 1-85233-594-7, Eddition Springer, 2002.
- [9] J. Timmis “ Artificial Immune Systems: A novel data analysis technique inspired by the immune network theory ”, PhD Thesis, University of Wales, 2001.

# Source ID Based Security (SIBS) Algorithm for Wireless Sensor Networks

Fahad T. Bin Muhaya, Adeel Akhtar and Fazl-e-Hadi  
 Management Information System, College of Business School,  
 Prince Muqrin Chair (PMC) for IT Security,  
 King Saud University, Riyadh, Kingdom of Saudi Arabia  
 {fmuhaya, adeel, fhadi}@ksu.edu.sa

**Abstract**— Today information security is foremost demand of every application. Data confidentiality is really a challenging task for Wireless Sensor Networks (WSNs). A light weight and full proof security algorithm is introduced in this research work. We have named it “Source ID Based Security (SIBS)” algorithm. SIBS algorithm involves the identification of sender node that is changed after fixed interval of time to mislead the attacker. Results have shown that proposed algorithm provides 7% resistance to different types of well known attacks including Eavesdropping, Denial Of Service (DOS) and false insertion of data. The simulation has been carried out using TOSSIM [2] simulator.

## I. INTRODUCTION

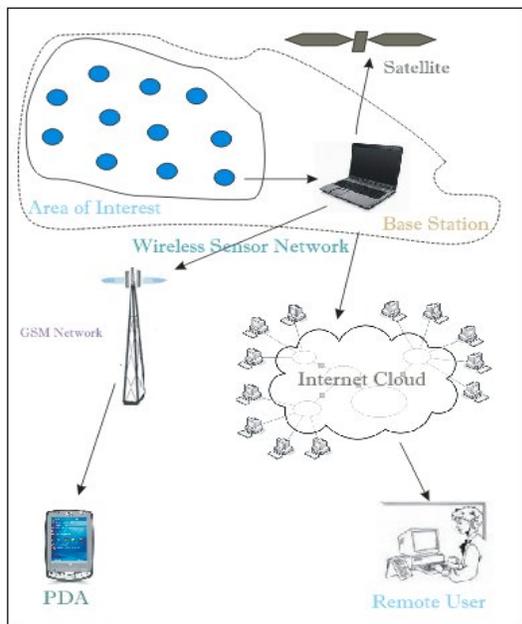


Figure 1. Overview of WSN and its implications [1]

Wireless sensor network is a versatile network for supporting variety of important application. Figure 1 demonstrates an abstract view this network with possible applications. This network is formed by deploying the sensing nodes in the area of interest; the deployed nodes form a self configured networks and starts acquiring the required information. The acquired information is then routed to the Base Station (BS). The BS is highly enriched system having advanced computing capabilities in it. As the figure 1 depicts, this acquired information can be used in any applications. It can be used for GSM networks,

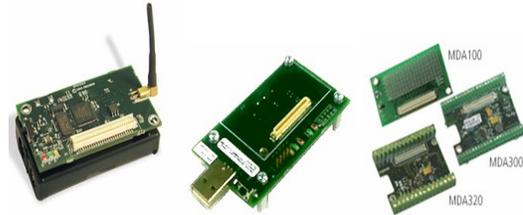


Figure2. Micaz mote with interface and sensor board [3]

remote user facilitation via internet etc. So to get information about a critical area while sitting at a remote location is an interesting capability of wireless sensor network.

The nodes in this network are battery operated and have limited lifetime to operate. Therefore there is a need of energyaware security algorithm. The security algorithm should not perform heavy computation on the nodes because it leads to minimize the network lifetime. A sensor node is composed of five basic components Transceiver, Memory, CPU, Battery and Sensor. The hardware node is called mote.

Figure 2 shows the latest model of a micaz mote, interface board MIB520 and MDA series of sensor boards. Table 1 shows the complete specification of a micaz mote.

Table 1. Specifications on Micaz mote [3]

Processor	Atmel ATmega128L
Program Flash Memory	128 KB
Measurement Serial Flash	512 kb
Serial Communication	UART
Frequency Band	2400MHz to 2483.5 MHz
Transmit Data Rate	250kbps
RF Power	-24dBm to 0dBm
Outdoor Range	75m to 100m
Indoor Range	20m to 30m
Battery	2 AA Batteries
User Interface	Red, Green and Yellow LED
Size	2.25 x 1.25 x 0.25inch
Weight	0.7oz (w/o batteries)
Expansion Connector	51 pin

Table 1 shows the specification of a micaz mote. It is a small and cheap device developed by Crossbow Technology that can be configured to get required information.

## II. LITERATURE SURVEYED

WSNs got very much attraction from the researchers in the recent years. There are a lot of interesting research problems including the hot topic of energy awareness. The latest research of the topic concerned is cited in this section. There are various research areas in this field like routing, localization, security, hardware design etc. Energy aware security technique is the scope of this study. The literature surveyed for this work is also related to routing in WSN. Lot of work has been done for efficient routing in this technology.

Ibriq and Mahgoub in [4] discussed types of routing models for WSN. According to authors the routing models for WSN are one hop model, multihop model and cluster-based model.

### A. One Hop Model

A simple and early mode of routing in WSN, where each node sends its data directly to Base Station.

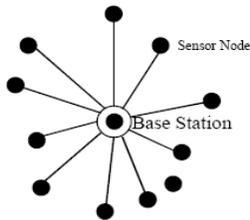


Figure 3 One hop model [4]

### B. Multi-Hop Model

As the direct routing model was not energy efficient model because nodes which are farther from the base station were forced to put more energy to transmit their respective data towards the base station. After the direct model, multihop model was introduced. Rather than sending data directly towards BS the nodes use their neighbors to transmit the data. In this way farther nodes can save their energy which leads to maximize the network lifetime.

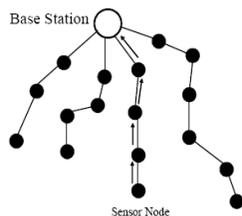


Figure 4 Multi-hop Model [4]

### C. Cluster-Based Model

Multihop routing was also having some issues with it. The nodes closer to the base station consume their energy quickly because they have to forward the whole network data to the BS and vice versa. At this stage cluster routing

came into existence in which clusters are formed in the network. The cluster member nodes send their data towards the cluster head and finally the cluster head forwards the data to the ultimate BS.

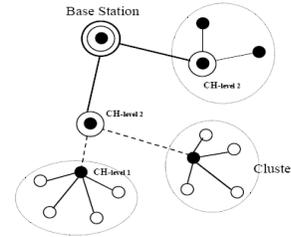


Figure 5 Cluster based model [4]

Researchers have developed many routing protocols for the above mentioned models but still there are numerous issues which need serious consideration.

## III. LIMITATIONS IN LITERATURE SURVEY

WSNs have put the human race in a new era of technology, but still it has many issues which need to be solved. Lot of work has done to make WSNs more secure and robust against all type of attacks. We have gone through state of the art work regarding security in WSNs which revealed that security algorithms need optimization. The study proposes a novel approach of the source id based encryption.

### A. Objectives

After going through the study of latest security issues we can say there is need of such security algorithm that cause less computation of energy for securing data. The objectives of this research work are to develop a security algorithm that is:

1. Suitable for large network size.
2. Secure from both passive and active attacks.
3. Better than traditional security techniques.
4. Should consume less memory and energy for providing solid security.

## IV. PROPOSED SOLUTION

### Source ID Based Security Algorithm

In SIBS when the network starts each node gets a pre calculated file containing randomly generated IDs. Same file is also provided to Base Station. Each node in the network gets its own independent file non identical to other network nodes. When the network starts each node reads the respective file and sets its id accordingly for specific time interval. After expiry of set time interval the file is again read and the ID is changed accordingly. This change is totally synchronized with BS. So all new IDs for all the network members are also known to the BS. So BS will always have updated IDS file of its nodes. Thus whenever an attacker tries to perform passive attack that is eavesdropping, it is not impossible for him to get the accurate data. Because the sensed information in WSNs are of small packets and to get that information one

should get the complete packets. In this scenario when attacker penetrates into the network and starts getting the data packets it will of course keep track of all received data packets, e.g. packets from node x are numbered as 1, 2 etc. Similarly when the sending node will change its id the attacker will simply consider that this packet number 2 belongs to another node y and the attacker will discard the packet. Because the attacker is actually interested in the data packets from the node x. In this way the attacker can be deceived easily and security may be achieved in a very simple way by changing the source ID of the sending node.

A. Pseudo code for SIBS Algorithm

**START**

INITIALIZATION

```

Source ID file generation for each node
Distribute these files to respective nodes
e'=20mJoules
BS=Base Station
// End of Initialization
    
```

```

Get Sensed Info
Data= Info
    
```

**Energy Check**

```

IF (Energy > e') THEN
    Call Data Transmission
ELSE
    Call END
    
```

**Data Transmission**

```

Sense required info
Data =Info
Destination ID selection

WHILE (t ≠ t')
    Send Data to BS
END WHILE

Call Change ID
    
```

**Change ID**

```

Reinitialize t
Read the Source ID file
Set new ID accordingly
Call Energy Check
    
```

**END**

```

Exit simulation
    
```

Figure 6 and pseudo code describe the overall functionality of the enhanced energy aware cluster routing (EEACR). In start all the nodes calculates their respective distances from the cluster head. On the basis of calculated distances the nodes decide the mode of routing. If the distance is less than a predefined threshold value then the mode of routing will be direct. And if the distance is more than the threshold value then the mode of routing will be multihop. So by combining both the techniques we can prolong the network lifetime, can increase packet delivery and can reduce energy consumption.

V. SIMULATIONS

Different security test have been performed to verify the worth of our research work. First test is performed to

check the confidentiality of algorithm. Our algorithm was compared with some well known security algorithms and

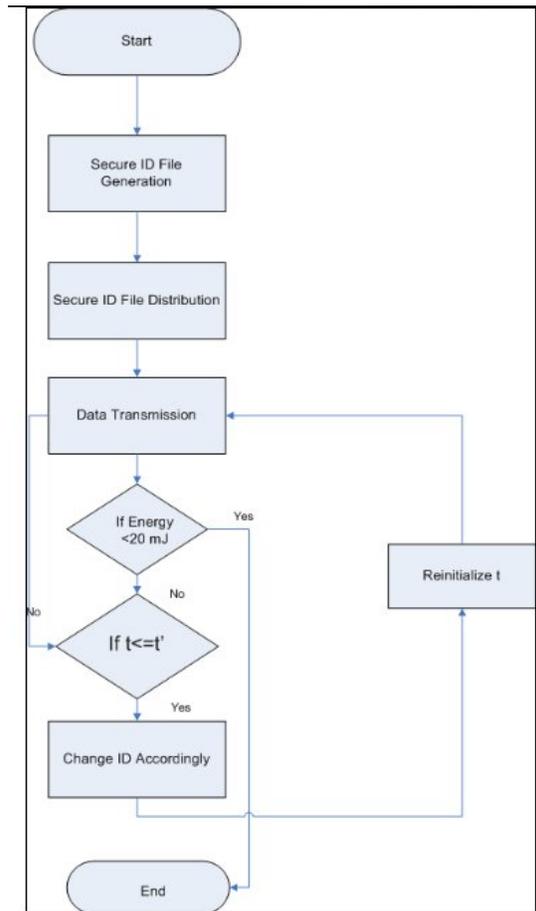


Figure 6. Flow Diagram for Energy Efficient Adaptive Cluster Routing

achieved desirable results.

First test was taken mathematically, suppose an algorithm encrypts *n* number of packets in *t* seconds. The attacker in between sender and receiver receives the packets and needs *r* number of packets to decrypt the information.

Successful Attack:

$$r \rightarrow (n)_t$$

Thus the algorithm must be as secure as it makes impossible to get *r* number of packets. Proposed algorithm is more secure that if attacker gets more than *r* packets even then he will not be able to get the actual information because it totally unknown to him that to whom this received packets belongs to.

Our next test performed is the total number of packets sent to BS. As we are putting some little computation before sending packets towards the BS so our number of packets sent will a little bit less. But on other hand by putting this computation we are able to get maximum security which is more worth full.

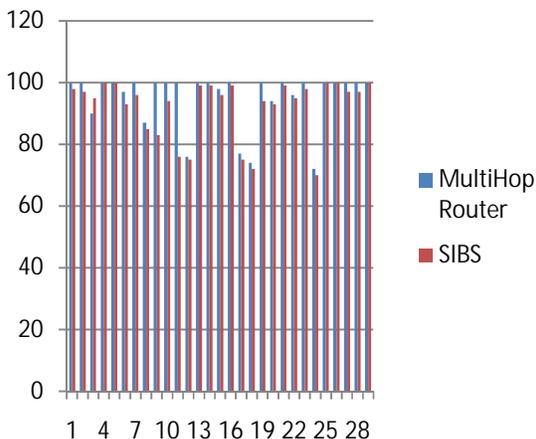


Figure 7. Total no of Packets sent

Figure 7 depicts the total packet sent by each node. The network consists of 30 nodes, node 0 is the Base Station. Our algorithm was checked with a Multihop routing algorithm.

Logical analysis is carried out by comparing proposed algorithms with some well known algorithms. The table mentioned below show how SIBS is more secure and reliable than other algorithms. The reason of such security is misguiding the attacker in such a way that if it is able to get information still will be useless for it. The only way to break is when attacker gets the complete file having mapping keys in it. And that file is not transferred in communication but it is prebuilt file in all the member nodes.

Table2. Comparison of Different Security Algorithms

Algorithm	Key size	Confidentiality	Data Tempering	Security Level
RSS	64bit	Good	Good	High
SAM[11]	256bit	Good	Good	Good
DMA[11]	128bit	Good	Good	Good
SIBA	145bit File	Very Good	Very Good	Very High

VI. CONCLUSIONS

In this research work a new technique is proposed that has increased the data confidentiality of the network. SIBS technique is compared with a well known security algorithm and results are simulated in TOSSIM simulator. While justifying our idea through results of our simulation we have considered the DOS and data

confidentiality attacks. Simulation showed that SIBS increased the security of the network as compared to the other traditional security algorithms.

REFERENCES

1. Adeel Akhtar, Abid Ali Minhas, "Impact of adaptive intra cluster routing on energy efficiency", third symposium on Engineering Sciences, Punjab University Lahore, March 2010.
2. <http://www.tinyos.net>.
3. <http://www.deeds.informatik.tuarmstadt.de/dewenet/images/MICAz.jpg> (micaz mote)
4. J. Ibriq and I. Mahgoub. Cluster-based routing in wireless sensor networks: Issues and challenges. Department of computer science and Engineering, Florida Atlantic University.
5. W. Heinzelman, A. Chandrakasan and H. Balakrishman. 2000. Energy-efficient communication protocol for wireless microsensor networks.
6. Ma and Y. Yang. 2006. Clustering and load balancing in hybrid sensor networks with mobile cluster heads. ACM Third International Conference In Quality Of Service In Heterogeneous Wired/Wireless Networks. Department of Electrical and Computer Engineering State University of New York, Stony Brook, NY 11794, USA.
7. N. Israr and I. Awan. Multihop clustering algorithm for load balancing in wireless sensor networks. University of Bradford, UK.
8. A.T. Hoang and M. Motani. 2007. Collaborative broadcasting and compression in cluster-based wireless sensor networks. ACM Transactions on Sensor Networks. Vol. 3, No. 3, Article 17, August 2007.
9. T. Wu and S. Biswas, "Reducing Inter-cluster TDMA Interference by Adaptive MAC Allocation in Sensor Networks". Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks. Dept. of Electrical and Computer Engineering, Michigan State University
10. Adeel Akhtar, Abid Ali Minhas and Sohail Jabbar, "Adaptive Intra Cluster Routing for Wireless Sensor Networks", International Conference on Convergence and Hybrid Information Technology (ICHIT), 27-29 August 2009, Daejeon Korea.
11. Masahiro Mambo, Yuliang Zheng, "Information security", Second International Workshop, ISW'99, Kuala Lumpur .

**SESSION**  
**AUTHENTICATION + BIOMETRICS**

**Chair(s)**

**TBA**



# Easing Text-based Mobile Device User Authentication Mechanisms

**Dugald Ralph Hutchings**  
Computing Sciences Department  
Elon University  
Elon, North Carolina, USA  
dhutchings@elon.edu

**Abstract** - *We discuss how a variety of techniques relevant to text entry or user identification could be used to facilitate reasonably fast and secure methods for authenticating mobile device users. In particular, we explore how mobile device text-entry methods could be used to speed up both the initial authentication of a mobile device user as well as subsequent password entry for authentication to other systems such as password-protected Web sites. We also explore other authentication mechanisms that involve text entry, maintain device security, but avoid the need for a password. We conclude that all explored techniques are viable approaches to increasing mobile device security while also providing easier access methods, as measured by the predicted speed and accuracy of entered text.*

**Keywords:** mobile device security, passwords, text entry

## 1 Introduction

The ability of mobile devices to conduct commercial, educational, personal, and other tasks continues to increase. Some formal research justifies this claim; for example, the Pew Research Center reported in July 2009, “56% of adult Americans have accessed the internet by wireless means, such as using a laptop, mobile device, game console, or MP3 player,” and “one-third of Americans (32%) have used a cell phone or Smartphone to access the Internet for emailing, instant-messaging, or information-seeking” [3]. As the number of tasks that mobile devices support increases, so too will the amount, the complexity, and in particular the sensitivity of the information stored and transferred by these devices. This situation of increasing information storage and transfer raises the issue of whether the security mechanisms available on these mobile devices are providing adequate protection.

Take for example the act of authenticating a user to any physical computing device. On a typical laptop or desktop system, users must authenticate by providing both a username

and a password. Typical passwords are sequences of 8 or more textual characters, often involving a mix of character types (upper-case letters, lower-case letters, numbers, punctuation, and other special characters) and yielding roughly  $10^{16}$  raw possible combinations. A wide variety of mobile phones and other hand-held devices typically use a much simpler password-like system and do not ask for a username. For example, Nielsen reports that the most popular mobile phone in the United States is the iPhone [23], which prompts the user to enter a simple personal identification number (PIN) consisting of a sequence of 4 numbers (yielding a raw amount of  $10^5$  possible combinations). Other devices ask the user to enter a simple gesture connecting some number of dots on a  $3 \times 3$  grid (typically offering well under  $10^5$  gesture-stroke combinations). While these methods do allow users to have a relatively rapid and simple way to access their personal devices (i.e., they are highly *usable*), they also are not nearly as secure as the standard text-character based password systems that offer substantially more combinations. Likewise, an 8-character text-based password consisting of a mixture of letters (and letter cases), numbers, and special characters is not as appealing as a PIN given the much slower rate at which people can enter text characters on a mobile device. Again, taking the iPhone as an example, users must switch among different virtual screens to access numbers and special characters on its graphical touch-based keyboard.

Although a number of tools have been introduced to allow mobile device users to enter words more quickly, such tools are typically based on matching user actions to an open dictionary of words. As a result, these tools cannot be well applied to password entry, where the password often is not a dictionary word (and in fact, may not be allowed to be such a word).

Even if a superior fast-but-secure authentication method is developed for mobile devices, once a user is authenticated to use the device, then the user often is faced with the task of entering different textual passwords to be able to access corporate networks, to use Web-based information systems, to complete commercial transactions, or to access other secured

information resources. Because of the limited size of a mobile device, the keyboard available on the device typically consists of a number of very small hardware buttons below the screen or a number of small touch-based graphical buttons on the screen. In either case, the small size of the target buttons results in users mistakenly entering erroneous text, making the task of private password entry difficult and time-consuming. As just stated, in this situation, tools such as text auto-correction or dictionary word matching do not assist the user because passwords often are restricted from using words or other textual sequences found in a dictionary. Whether used for initial device authentication or for subsequent access to other systems through the mobile device, it is worthwhile to investigate how the process of text-based authentication on mobile devices could be improved to both maintain the security of the user's data, but also allow the user to enter the password in a reasonably fast manner.

In this paper we discuss the intersection of security and usability by analyzing the possible benefits and drawbacks of adapting text-based authentication strategies to mobile device use. For each strategy, we separately discuss authentication to the device itself and subsequent authentication to other systems. Though none of the discussed text-based techniques specifically have been evaluated in a lab or naturalistic mobile device authentication setting, we report on some past evaluations and provide predictions of performance when possible. Future work will involve targeted testing of the ideas presented in this paper. Before discussing the specific techniques, we briefly analyze the relationship of our work to past research at the intersection of usability and security.

## 2 Related Work

In nearly any introductory textbook on computer security, readers will find a categorization of the available methods for authenticating users:

- *knowledge-based*, in which the user presents a piece of information that only he or she knows (passwords and PINs, e.g.);
- *object-based*, in which the user presents or uses a physical object that only he or she possesses (automated teller machine (ATM) card, hardware token, e.g.); and
- *biometric*, in which the user presents a property of his or her body that is considered to be unique enough for identification (fingerprint scan, iris scan, e.g.).

A single authentication process can involve one, two, or all three of these approaches spread among a variety of methods. With mobile devices, typically the device itself must be possessed to use its functionalities (few devices allow remote access), so it naturally uses an object-based method

for part of its authentication process. A variety of factors (among them cost, bulk, and power consumption) have led popular mobile device manufacturers to avoid fingerprinting, iris scanning, and other biometric approaches which do not place any additional knowledge (i.e., memory) burden on the user to authenticate. Thus knowledge-based approaches have prevailed, forcing users to possess the capability to remember the knowledge-based sequence. Recent research suggests that many users have difficulty remembering standard text-based passwords (see Hutchings's and Komanduri's results, e.g. [6]), leading to the exploration of alternative knowledge-based methods that exhibit higher *memorability* characteristics.

Chief among these explorations has been graphical or image-based approaches that can exploit the *picture superiority effect*, which in essence indicates that images are more memorable because they are stored in two different physical locations in the brain (once as the image itself, and separate as the words that describe the image [2], [11]). Approaches such as having users click on a sequence of images [6], click on a series of points in a single image ([14], [15]), or selecting from among a library of personally meaningful pictures [12] have all been investigated and indeed, the approaches were more memorable than text passwords when used on desktop computing equipment, though entry speed was not measured in any of the cited works. It is questionable how well the techniques can scale down to the mobile device, where screen space is much more limited as compared to the desktop and images may not be as distinguishable.

At SAM 2004, Jansen et al. presented a framework for *mobile* device security which included a method of allowing a variety of "password user interfaces" to be used in a flexible way [5], including a graphical method proposed by Jansen et al. in earlier work [4]. Jansen provides evidence of the feasibility of a graphical approach, but did not provide a user study of whether users could remember graphical sequences or how quickly users could undertake an image-based method. Then at SAM 2010, Citty and Hutchings offered TAPI, an image-based authentication mechanism adapted from previous work [6] but tailored specifically for touch-screen mobile device entry [1]. TAPI offers modestly more security than a PIN (about  $10^8$  possible sequences compared to  $10^5$ ) but not nearly as much as a standard textual password, is highly memorable (about 90% of participants were able to recall their randomly-assigned image sequence after a week on non-use), but is still somewhat slow (in the best possible case, users can expect to enter their image sequences in about 3 seconds). To gain comparable security to a standard text-based password, the number of available images for the sequence would likely need to be increased. However, it is unlikely that users would be able to see all of the images let alone accurately touch them, meaning multiple screens or scrolling would be necessary for image-sequence entry. The

amount of time needed to enter the image-sequence would likely well exceed 6 seconds (double the observed time).

Beyond the problems surrounding the difficulty of remembering and entering passwords, there is also a problem that graphical password entry can be observed and subsequently used to compromise a user's account (typically referred to as *shoulder surfing*). A number of techniques have been proposed at SAM (such as portfolio-based techniques [8]) and elsewhere (such as the Spy-resistant Keyboard [13]) to overcome this problem, but appear to require an amount of screen space not typically available on a mobile device.

In summary, graphical approaches appear to have inherent tradeoffs that may not be able to be overcome. Graphical approaches can be sped up at the cost of security, especially on mobile devices with limited screen space. If indeed security acquires heightened importance on mobile devices, standard textual passwords are likely to persist due to both security and familiarity. Also, textual passwords are likely to be entered beyond the initial device authentication, because users are attempting to access other systems. We have concluded that it is thus worthwhile to investigate how all text-based authentication mechanisms can be improved for entry speed, security, or memorability. We now move to describe a number of recent techniques for facilitating quicker text entry on mobile devices and how those techniques could be adapted for authentication.

### 3 Text-based Authentication

In considering how to build a suitable method of text-based authentication, we explore two research areas: text entry methods designed for mobile devices (but without concern for password entry) and non-password-based authentication designed for typical desktop interaction (but without concern for utility on mobile devices).

#### 3.1 Mobile Device Text Entry

With the rise in popularity of mobile devices, there has been considerable attention paid to improving the ease of use of the devices. The entry of text is one such area. A variety of techniques have been created to increase the speed of overall entry of text by allowing users to execute manual typing actions faster, reduce the likelihood of typing errors, and/or replace typing with alternative actions. In this subsection we explore how two promising techniques could be adapted for both initial device authentication and subsequent entry of standard text-based passwords.

##### 3.1.1 ShapeWriter

Initially named SHARK and commercially available as ShapeWriter [21] or Swype [22], the work principally completed by Zhai and Kristensson allows a user to move his

or her finger across an image of a keyboard on a touchscreen, stopping on each letter of a word that the user intends to type. The movement of the finger is traced and the shape created by the user is matched to a database of shapes that correspond to words. Finally, the matching word is entered as text into the interactive text area [7]. Several studies of the tool have demonstrated that users can achieve rates of text entry on mobile devices that are comparable to standard desktop keyboard entry (in the earliest study of the system, typical rates of 45 words per minute were achieved and exceptional rates of 70 wpm were achieved [7], with very low error rates observed).

Without adaptation, ShapeWriter could be used to increase the security of typical 4-number PIN-based systems without sacrificing entry speed or accuracy by substituting for the PIN with the entry of a dictionary word (in the most recent release there are  $6 \times 10^5$  word combinations). Requiring two or three consecutive words would further increase security but likely diminish entry speed to 2-4 seconds, perhaps not overly burdensome to the user. Traditional advice regarding password construction is that words are to be avoided, though recent research indicates that dictionary words may not be a poor approach if the password is not repeated across users and an account lockout mechanism is used (in which the user account is frozen until an administrator unfreezes it) [10]. However when the mobile device is personally administered and not shared among a large user group, account lockout might not be a realistic approach.

ShapeWriter allows each of its users to create custom words (and their corresponding shapes) that do not match the standard dictionary [21]. This feature further eases the entry of text by creating shapes that match commonly used slang words, domain terms, people's names, acronyms, etc. Though this feature could be used to allow users to create shapes that match non-word password combinations, a main problem is that the word-shape database is not encrypted, thus leaving a plain text version of the password available for an illegitimate user to record (similar to writing a password on a sticky note within reach of a desktop computer). Further, stored words are in all lower-case letters. When the user wants to capitalize a word, additional action is necessary. Since passwords often include mixtures of upper-case and lower-case letters (not to mention numbers and special characters), such actions would slow the user down and ultimately make the technique as slow as or slower than the virtual keyboard technique that ShapeWriter is designed to replace.

Of course, altering ShapeWriter so as to facilitate the distinction of upper- and lower-case letters and to encrypt all word-shape match entries would make it suitable for use when authenticating to other systems through the mobile device (such as Web-based email or a corporate intranet). However, making these alterations could significantly slow the translation from shape to word and reduce the heightened

usability of the approach in the first place. Allowing ShapeWriter to distinguish between password-entry situations and standard text-entry situations would allow only the user-defined shapes to be encrypted and maintain the core elements of usability. The ability (and security) of storing a user's passwords on a system in an encrypted list is well known (see Schneier's Password Safe, e.g. [20]), though the limitation is that the password to the system itself must be stored elsewhere.

In summary, ShapeWriter offers a compelling alternative to standard approaches, as long as the initial entry to the system involves a dictionary word (or sequence of words) for authentication. If a higher degree of security is required for initial entry, a different system is very likely to be necessary.

### 3.1.2 EdgeWrite

Initially designed for stylus-based interaction on a sunken surface, EdgeWrite is a stroke-oriented system for entering individual text characters [16]. The strokes are constrained so as to start and end in a corner of the sunken area, thus minimizing the chance for error and providing a very quick creation and subsequent software analysis of the strokes (the path does not matter, just the end points of the stroke segments). Most characters consist of two to four strokes in sequence. Subsequent research on the overall interaction pattern has shown that the technique is very versatile and has been applied to trackballs [17], eye-tracking [18], and notably among others, touchscreens [19]. Text entry rates for pen-based EdgeWrite are far slower than for ShapeWriter, but on average a single character can be stroked in 0.3 seconds [16] (note that though no formal research specifically on the touchscreen version is yet available). Past research on the various flavors of EdgeWrite show that users can expect to make very few typing errors using very few strokes (under 1% of all attempts in a user study were erroneous) [16]. Thus, we have an expectation that a password consisting of 9 characters could be entered in 3 seconds or less.

Unlike ShapeWriter, EdgeWrite is a character-based approach rather than a word-based approach, with an option for word completion upon a user's explicit request. The main advantage to this approach is that users can enter information one character at a time, eliminating the need to store a plain-text version or encrypted version of the password on the system (and thus the need for the user to take the time to train the system how to match a shape to a password). Further, the technique can be used easily for system authentication as well as subsequent access to other systems such as Web sites. The main disadvantage of this approach is that entry speed may decrease, especially with very long passwords or passwords. Further, the level of training needed to learn the full set of characters in EdgeWrite is somewhat extensive (especially compared to ShapeWriter and the standard virtual keyboard,

where essentially no training is needed), though since the technique has been shown to be usable on multiple devices, that training is portable to essentially any type of standard device, mobile or otherwise.

### 3.1.3 Memory and Observability

An issue with *any* knowledge-based approach (such as a password or PIN) is user memory: should the user forget the password or PIN, then the user's device becomes completely inaccessible. It is worthwhile to investigate not only methods for easing knowledge-based approaches, but also alternatives to the approach.

Another issue with any password-based technique is how easily a malicious individual can observe the authentication process and then repeat it at a later time to gain unauthorized access. If such a person was to video record a password session using a normal desktop keyboard, a small keyboard with physical keys, an on-screen virtual keyboard, or a partial interactive area with no keyboard representation, then that person is fairly likely to be able to determine a user's password. However in a more casual observation environment (such as fellow passengers on a bus or train or a passerby in a busy office), the likelihood of success differs. Take for example the Apple iPhone keyboard. To allow users to more easily determine which virtual key is being pressed, the display shows a zoomed-in version of the character on the screen as the user types. A password entered in this fashion is more likely to be observed than fast-moving gestures, whether those gestures take place on top of a keyboard image (ShapeWriter) or a blank surface (EdgeWrite).

Typically mobile devices can be accessed only by also physically possessing the device itself. As stated previously, technologies do exist to allow remote access to the device, but such technologies are not prevalent. As such, the remaining authentication approach to investigate involves biometrics. Most biometric approaches rely on strictly physical attributes such as fingerprint or iris composition. However a recent line of research suggests that *cognitive* biometric factors can be observed through analysis of physical action and further, such factors are largely immune to observation.

## 3.2 Behavioral Biometrics

The term behavioral biometrics refers to the concept that a person's cognitive or mental capabilities are candidates for biometric authentication, though such capabilities must be measured through observable actions (i.e., behaviors). Then, when combined with other physical properties of the person or his or her actions, the collection can be used in an authentication situation.

The most relevant recent research in this area belongs to Mohammed and Traore [9]. In one of their experiments, the

user trains the biometric system by entering sequences of characters (about 24 to 36 characters in length) on a virtual keyboard. The keyboard keys are shuffled, allowing the system to measure and model how quickly users visually scan the space. Further, the character sequences are set so as to use repeated characters, allowing the system to also measure and model the short-term memory characteristics of the user. By also measuring the areas on the keys that user click with the mouse, how quickly the user moves the mouse from key to key, how long the user hold the mouse button down, etc., the system builds a unique profile of each user. When the user desires to authenticate in the future, then the user enters a character sequence and the system checks the user's actions against the profile. In short, although the user enters text, there is no password and hence no knowledge-based component of the authentication system.

This system is less sensitive to observation than password-entry methods because the virtual keyboard key locations are shuffled for each authentication attempt and the character sequences likewise are randomized for each attempt. Knowing how a user moved between two keys in a previous successful authentication situation is not particularly helpful when the keys are now in different locations or the characters are not used in sequence.

Such a system of behavioral biometrics seems best suited for touchscreen mobile devices where the interaction is most similar to the mouse. While items such as the user path from one virtual key to the next cannot be measured, most of the factors used by Mohammed and Traore can also be measured on the mobile device. A key question to be addressed is the degree of certainty achieved with each subsequent entry of a text character. Similar to the way that the PIN offers a quick and typically accurate authentication at a cost of reduced security, a short text sequence similarly may provide a less certain identification of the user, but at a faster rate than the entry of a long character sequence. The researchers do not report how quickly a user can authenticate using this novel approach, and key research remains to reduce the amount of time as much as possible. In this section therefore we do not predict how quickly the user can authenticate.

The behavioral biometric system can be used only for the initial authentication of the user to the device but clearly cannot be used for direct entry of passwords to other systems. However, the behavioral biometric system can be used indirectly. Once the user enters a password for an external system using the standard text entry method, the device could store (and encrypt) that password for later use, but make access to that stored password dependent upon another biometric authentication session (likely shorter in duration than the initial authentication time).

### 3.3 Mixing Methods

The combination of the methods described in this paper however might be helpful in creating a quick entry method. Suppose that instead of entering an arbitrary sequence of 32 characters, instead the user enters 4 8-character words using the ShapeWriter entry method. Since the user is creating a gesture, these gestures can be analyzed over time to understand the likely shape that a user will make when entering words with certain properties (such as repeated characters) and also to identify words that users seldom or never type, meaning that user cognitive abilities such as short term memory are more likely to be exposed. Likewise the stroke-based EdgeWrite system could be monitored in a similar manner, allowing a smaller number of characters to be used but also enforcing the user to enter seldom used characters which may require searching and memory. In essence, we can leverage both the elements of a text-entry mechanism that the user has learned well and has learned poorly to provide a reasonably secure but rapid authentication mechanism based on biometrics, not knowledge.

## 4 Summary and Discussion

Our goals in this paper are to expose the issues surrounding mobile device authentication, explore usability research that has not yet been applied to security contexts (and vice-versa), and predict how well such research would be able to be applied. In short, the prevailing short PIN or brief gesture methods for mobile devices are much less secure relative to a typical desktop or laptop password system, though mobile devices are beginning to store data that is just as sensitive and run applications that are just as powerful. Adopting a stronger password mechanism for mobile devices can be problematic due to the extended time needed to authenticate to a device that has very small keys, which either slows the user considerably or encourages the pressing of erroneous keys (or both). Typical methods of mitigating typing errors (such as zooming the most recently pressed key) heighten the likelihood that an observer can steal a user's password.

As a result it is worthwhile to explore how to gain more security than a PIN approach without sacrificing too much in usability. One method is to adapt more usable text entry methods (such as ShapeWriter or EdgeWrite) to the password entry scenario. Another method is to adapt desktop behavioral biometric approaches to mobile devices. There are also possibilities to combine text entry methods with a behavioral biometric system, further heightening usability by potentially eliminating the knowledge-based approach (so the user need not be required to remember information).

Since users of mobile devices may also need to connect to other systems through their device, the problem of password entry may remain even if the initial authentication

avoids a password-based approach. Examples of using ShapeWriter, EdgeWrite, and possibly combining behavioral biometrics with a text entry device were explored as faster, more accurate, and more secure methods of password entry than typical techniques that highlight user key presses and increase the possibility of shoulder surfing.

We provided some approximations of the levels of security and levels of usability of the adaptations based on existing research. We propose that the community move to implement more specific prototypes of the ideas and subsequently compare the analysis of the collected data to the predicted results. Such explicit research could significantly advance the low level of security of current approaches while maintaining the high level of usability (notably, entry speed), or even exceeding current usability (notably, by eliminating the user's burden of remembering passwords).

## 5 References

- [1] J. Citty and D. R. Hutchings, "Design & evaluation of an image-based authentication system for small touch-screens," in *Proc. SAM – Int'l. Conf. Security and Management*, Las Vegas, Nevada, USA, 2010.
- [2] J. Deregowski and G. Jahoda, "Efficacy of objects, pictures and words in a simple learning task," *Int'l. J. Psychology*, vol. 10, no. 1, pp.19–25, 1975.
- [3] J. B. Horrigan (2009, July 22). "Press Release: Mobile internet use increases sharply in 2009 as more than half of all Americans have gotten online by some wireless means," Pew Internet & American Life Project [Online]. Available: <http://www.pewinternet.org/Press-Releases/2009/Mobile-internet-use.aspx>
- [4] W. Jansen, S. Gavrilov, V. Korolev, R. Ayers, and R. Swanstrom, "Picture password: a visual login technique for mobile devices," National Institute of Standards and Technology, Gaithersburg, Maryland, USA, Tech. Rep. NISTIR 7030, July 2003.
- [5] W. Jansen, S. Gavrilov, V. Korolev, T. Heute, and C. Séveillac, "A Unified Framework for Mobile Device Security," in *Proc. SAM – Int'l. Conf. Security and Management*, Las Vegas, Nevada, USA, 2004.
- [6] S. Komanduri and D. R. Hutchings, "Order and entropy in picture passwords," in *Proc. Graphics Interface*, Windsor, Ontario, Canada, 2008, pp. 115–122.
- [7] P. O. Kristensson and S. Zhai, "SHARK2: A large vocabulary shorthand writing system for pen-based computers," in *Proc. ACM UIST 2004*, Oct 24-27, Santa Fe, New Mexico, pp. 43–52, ACM Press.
- [8] S. Man, D. Hong, and M. Mathews. "A shoulder-surfing resistant graphical password scheme," in *Proc. SAM – Int'l. Conf. Security and Management*, Las Vegas, Nevada, USA, 2003.
- [9] O. Mohamed and I. Traore, "Cognitive-based biometrics system for static user authentication", in *Proc. International Conference on Internet Monitoring and Protection*, May 24-28, 2009, Venice/Mestre, Italy.
- [10] S. Schechter, C. Herley and M. Mitzenmacher, "Popularity is everything: a new approach to protecting passwords from statistical-guessing attacks," in *Proc. USENIX HotSEC*, August 11-13, Washington, D.C., 2010.
- [11] J. Snodgrass and B. McCullough, "The role of visual similarity in picture categorization," *J. Experimental Psychology: Learning, Memory, and Cognition*, vol. 12, no. 1, pp. 147–154, 1986.
- [12] T. Takada, T. Onuki, and H. Koike, "Awase-E: recognition-based image authentication scheme using users' personal photographs," in *Proc. Innovations in Info. Tech.*, Dubai, United Arab Emirates, 2006, pp. 1–5.
- [13] D. Tan, P. Keyani, and M. Czerwinski. Spy-resistant keyboard: more secure password entry on public touch screen displays. In Proceedings of the 19th conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction, pages 1–10. Narrabundah, Australia, 2005.
- [14] S. Wiedenbeck, J. Waters, J. Birget, A. Brodskiy, and N. Memon, "Authentication using graphical passwords: basic results," In *Human-Computer Interaction International*, Las Vegas, Nevada, USA, 2005.
- [15] S. Wiedenbeck, J. Waters, J. Birget, A. Brodskiy, and N. Memon, "PassPoints: design and longitudinal evaluation of a graphical password system," *Int'l. J. Human-Computer Studies*, vol. 63, no. 1-2, pp. 102–107, 2005.
- [16] J. O. Wobbrock, B. A. Myers, and J. A. Kembel, "EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion," in *Proc. ACM UIST*, Vancouver, British Columbia, Canada November 2-5, 2003, pp. 61-70.
- [17] J. O. Wobbrock and B. A. Myers, "Trackball text entry for people with motor impairments," in *Proc. ACM CHI*, Montréal, Québec, Canada April 22-27, 2006, pp. 479-488.
- [18] J. O. Wobbrock, J. Rubinstein, M. W. Sawyer, and A. T. Duchowski, "Longitudinal evaluation of discrete consecutive gaze gestures for text entry," in *Proc. ACM ETRA*, Savannah, Georgia, March 26-28, 2008, pp. 11-18.

[19] EdgeWrite for iOS (Accessed February 27, 2011)  
[Online] Available:  
<http://depts.washington.edu/ewrite/iphone.html>

[20] Password Safe (Accessed February 27, 2011) [Online]  
Available: <http://www.schneier.com/passsafe.html>

[21] ShapeWriter (Accessed February 27, 2011) [Online]  
Available: <http://www.shapewriter.com/>

[22] Swype (Accessed February 27, 2011) [Online]  
Available: <http://www.swypeinc.com/>

[23] Top Mobile Phones, Sites, and Brands for 2009  
(Accessed February 27, 2011) [Online] Available:  
[http://blog.nielsen.com/nielsenwire/online\\_mobile/top-mobile-phones-sites-and-brands-for-2009/](http://blog.nielsen.com/nielsenwire/online_mobile/top-mobile-phones-sites-and-brands-for-2009/)

# A PASS Scheme in Cloud Computing - Protecting Data Privacy by Authentication and Secret Sharing

Jyh-haw Yeh

Dept. of Computer Science  
Boise State University  
Boise, Idaho 83725, USA

**Abstract** - *Cloud computing is an emerging IT service paradigm. Instead of developing their own IT departments, business sectors purchase on-demand IT service from external providers in a per-use basis. From business cost perspective, companies are shifting capital expenses (CapExp) on hardware equipments to operational expenses (OpExp). Many companies, especially those startups, found this cloud IT service is economically beneficial. However, this IT service paradigm still requires to overcome some security concerns before it can be fully deployed. One of the main security issues is how to protect client's data privacy from cloud employees. This paper proposes a PASS scheme for this purpose using Authentication and Secret Sharing.*

**Keywords:** Cloud Security, Data Privacy, Authentication, Secret Sharing

## 1 Introduction

Cloud computing is an emerging research field to facilitate the deployment of a new IT service paradigm. Cloud providers offer variety of IT services to subscribers and charge them in a per-use basis. Many researches [1-5] have identified the potential benefits of this IT paradigm such as cost saving, innovative technology fast development, better resource utilization, and arguably better security protection.

Data storage is a popular IT service provided by the cloud. For example, the Amazon's S3 - Simple Storage Service [6]. In traditional data management models, people store and protect their own data under their own authority, whereas in cloud computing, the responsibility of data management and protection no longer belongs to the data owner. This fundamental change brings some new security challenges that traditional security solutions might not work. One of the most diffi-

cult challenges is the protection of data privacy [7-11]. Cloud's data center stores data from different clients. These data may physically reside in the same hardware. Thus, cloud providers must have an effective data isolation mechanism to prevent illegal data accesses from outsiders, other clients, or unauthorized cloud employees. The protection against malicious cloud employees is a difficult problem and may require a fully homomorphic encryption algorithm. Unfortunately, cryptologists were not able to find any such encryption algorithm for years. Actually, they are not even sure whether such encryption algorithm exists.

Without homomorphic encryption algorithms, it is unlikely for clients to have 100% data privacy against cloud employees. Thus, the scheme presented in this paper is intended to reduce the data privacy risk to a minimum. The proposed scheme is named "PASS" because it protects clients' data Privacy by Authentication and Secret Sharing. The PASS scheme combines several coherent security components such as public key cryptosystem, key agreement, key management, authentication, and access control. This paper will present the design choices of each component and describe its relationship to others. In general, the design of the PASS scheme is based on the following guidelines:

1. Ensure secret isolation (or secret compartment). That is, a reveal of a secret should not result in a reveal of another secret.
2. Security protection should be considered with a higher priority than efficiency.
3. Select a design choice that would benefit as many security components as possible.

For example, in trying to ensure the main security goal "data privacy", the PASS scheme chooses not to store encryption keys anywhere in the cloud

because of the secret isolation guideline. More detailed explanation of this design choice will be described later in Section 3.

The rest of this paper is organized as follows: The five security components are described from Section 2 to Section 6. Section 7 presents the PASS scheme by providing a sequence of walk through steps to show how to integrate these five security components. Section 8 concludes the paper.

## 2 Public Key Cryptosystem

RSA [12] and elliptic curve [13] are two popular public key cryptosystems. In the proposed PASS scheme, elliptic curve was chosen over RSA. Elliptic curve cryptosystem (ECC) provides the same level of security as RSA, but with much smaller key size [14]. In addition, based on ECC, there is a convenient and efficient key agreement algorithm, which can be used in the key agreement security component (design guideline 3).

### 2.1 ECC Setup

The cloud provider, based on the recommendation from National Security Agency [15], chooses an appropriate prime curve  $E(F_p) : y^2 = x^3 + ax + b$  over a prime  $p$  and a base point  $G$  with an order  $n$ , where  $nG = O$ . The cloud publishes the chosen ECC domain parameters  $\langle p, a, b, G, n \rangle$ .

### 2.2 Public Key Generation

The cloud's server chooses a random number  $d_s \in F_p$  as its private key and then computes its public key  $D_s = d_sG$ . Similarly, each client  $i$  establishes his own private-public key pair  $d_i \in F_p$  and  $D_i = d_iG$ .

### 2.3 ECC cryptography

There are many ECC encryption and signature algorithms available in the literature. All these algorithms are based on the hardness assumption of the Elliptic Curve Discrete Logarithm Problem (ECDLP) that states: *Given two points  $D$  and  $G$  on the curve, where  $D = dG$  for some  $d \in F_p$ , it is computational hard to find  $d$ .*

ECC encryption and decryption will be used in the process of client authentication. Below are the notations used in this paper for these two operations. Let  $c$  be a ciphertext of a message  $x$ , encrypted by a public key  $D_i$ . That is,

$$c = ENC_{D_i}(x) \quad (1)$$

The ciphertext  $c$  can only be decrypted by an entity with a private key  $d_i$ , where  $D_i = d_iG$ . The notation of decrypting the ciphertext  $c$  back to the original message  $x$  is

$$x = DEC_{d_i}(c) = DEC_{d_i}(ENC_{D_i}(x)) \quad (2)$$

## 3 Key Agreement

Though there is a public key ECC established, for the purpose of efficiency, clients' data stored in the cloud should still be encrypted by a symmetric encryption algorithm such as AES [16]. Thus, the cloud's server needs to share a secret symmetric key  $k_i$  with each client  $i$ . The PASS scheme chooses not to store  $k_i$  anywhere in the cloud because of the following two secret isolation advantages:

- If the cloud's server is compromised, the privacy of stored data can still be preserved.
- If there are malicious employees inside the cloud, not storing the shared keys in the cloud reduces their chances of stealing sensitive data.

Of course, in response to a client's query, the server needs to derive the shared key so that the client's data can be retrieved for further processing. Only during this time frame, the malicious employees may have the chance to steal the key. To further protect keys during these time frames, it requires a strong access control component in the PASS scheme as a second defense, which will be described later in Section 6.

Now, the remaining question is how the server derives the shared key while receiving a client's query. The PASS scheme uses Shamir's secret sharing algorithm [17] to accomplish this. For each client  $i$ , the server has a secret share  $SS_i$  and the client has the other secret share  $CS_i$ . With both  $SS_i$  and  $CS_i$ , any entity is able to derive the shared secret key  $k_i$ . However, lack of any one of these two shares,  $k_i$  cannot be recovered.

For each client  $i$  and the server, the key agreement component in the PASS scheme is responsible for generating  $k_i$  and its two secret shares  $SS_i$  and  $CS_i$ . From the established ECC, the two parties (the server and each client  $i$ ) can easily agree on the two secret shares, even without the need to talk to each other. However, the agreement of the shared key  $k_i$  requires exchanging some messages using the Diffie-Hellman algorithm [18].

### 3.1 Secret Key Agreement

Before presenting the secret key agreement algorithm in the PASS scheme, we would like to describe an easy algorithm first, then followed by arguing the use of the easy one may violate a design guideline. It works as follows: Using ECC, the server computes a point  $Q_i = d_s D_i$  and the client  $i$  can compute the same point  $Q_i = d_i D_s$  since  $d_s D_i = d_s d_i G = d_i D_s$ . The shared symmetric key  $k_i$  can just be the x-coordinate of  $Q_i$ . The security of this key agreement scheme is based on the hardness assumption of the elliptic curve Diffie-Hellman problem. Even though this is an efficient key agreement algorithm without any message being exchanged, the PASS scheme decided not to use it because of the violation of the secret isolation design guideline. The violation occurs if the server is compromised and the private key  $d_s$  is revealed, an adversary can then derive the secret key  $k_i$  of any client  $i$ .

Thus, the traditional Diffie-Hellman key agreement algorithm will be used to establish the shared keys in the PASS scheme:

1. Both the server and the client  $i$  choose a random number  $r_s \in F_p$  and  $r_i \in F_p$ , respectively.
2. The server computes and sends a point  $R_s = r_s G$  to the client. Similarly the client  $i$  computes and sends a point  $R_i = r_i G$  to the server.
3. The server computes a point  $Q_i = r_s R_i$  and the client can compute the same point  $Q_i = r_i R_s$  since  $r_s R_i = r_s r_i G = r_i R_s$ . Both the server and the client permanently remove  $r_s$  and  $r_i$  from their storage after  $Q_i$  is computed.
4. The shared secret key  $k_i$  can then be the x-coordinate of the point  $Q_i$ .

Note that this process generates two extra messages. However, the key agreement is a one-time process and will be performed only when a customer subscribes as a new client.

### 3.2 Secret Shares Agreement

After  $k_i$  is agreed, the two secret shares  $SS_i$  and  $CS_i$  can then be generated separately by the server and the client  $i$ , respectively. In this secret share agreement algorithm, there is no extra message required. The algorithm works as follows:

1. Let  $a$  be the x-coordinate of the point  $Q_i + D_i$ , where  $D_i$  is the public key of the client  $i$ .
2. The server and the client  $i$  can construct a same polynomial

$$f(x) = k_i + ax \quad (3)$$

3. Both the server and the client  $i$  randomly choose their secret shares  $SS_i = (x_1, f(x_1))$  and  $CS_i = (x_2, f(x_2))$ , respectively, where these two shares are points on the polynomial.
4. After the secret shares are chosen, both the server and the client remove  $Q_i$  and the polynomial from their storage.
5. In a later query, the client  $i$  presents his share  $CS_i$  to the server. Combining with its own share  $SS_i$ , the server is able to reconstruct the polynomial and thus the secret key  $k_i$  can be recovered.

## 4 Key Management

There are many secret keys and secret shares that the server needs to keep track of. How to manage these keys and other useful information in an integrated fashion is another important design of the PASS scheme.

For each client  $i$ , the server keeps a profile storing the related keys and secret share, along with some other information that can be used later for key derivation, client authentication or access control. Table 1 below shows the contents of a client  $i$ 's profile.

Table 1: Cloud's server keeps a profile for each client  $i$ .

Client ID	$h(k_i)$	$SS_i$	$D_i$
Request Counter	Security Label		

Inside a profile, it does not store the secret key  $k_i$  in a clear form, but in a hashed form  $h(k_i)$ , where  $h$  is a cryptographic hash function. Inclusion of the hashed key is for the purpose of client authentication. Recall that not storing shared secret keys anywhere in the cloud is our major design choice to enhance data privacy. The server also needs to store its secret share  $SS_i$  for a later key derivation when the client  $i$  makes a query with his share  $CS_i$  attached. Note that after the query is processed, the server must discard both  $k_i$  and  $CS_i$  from storage to ensure a better data

privacy protection. It is also useful to keep the client's public key  $D_i$  in the profile. If the server would like to initiate a short but sensitive talk to the client  $i$ , it can use  $D_i$  to encrypt the message since the server does not have the shared secret key  $k_i$  at that moment. For the request counter field in the profile, initially it is set to 0. The counter increments by one each time the client  $i$  makes a request to the server. The purpose of this counter is to prevent replay attacks. Using this counter approach, the client needs to keep a counter on his side too. The authentication procedure will be described in Section 5. Finally the security label in a profile is used to implement mandatory access control. Section 6 has more discussion on this.

## 5 Client Authentication

Upon receiving a service request from a client, it is important for the server to authenticate the client. This authentication can be another defense of data privacy. That is, only allowing the authenticated client to access his own data.

Based on the design choices in other supporting components, the client authentication procedure in the PASS scheme is quite simple without the need of challenge and response handshakes, which are usually required in other typical authentication protocols.

The procedure starts by a client  $i$  sending a service request message with an authenticator

$$Client\ i \xrightarrow{ENC_{D_s}(client's\ counter || CS_i)} Server$$

where  $ENC_{D_s}(x)$  stands for using the server's public key to encrypt  $x$  and the symbol  $||$  means concatenation. Upon receiving the request, the server performs the following steps:

1. Use its own private key  $d_s$  to decrypt the received authenticator, i.e.,  $DEC_{d_s}(ENC_{D_s}(client's\ counter || CS_i))$ , and then retrieve the client's counter and the secret share  $CS_i$ .
2. Use the decrypted  $CS_i$  and the stored  $SS_i$  (in profile) to recover the secret key  $k_i$  using Shamir's secret sharing algorithm as described in Section 3.2, and then compute the hash  $h(k_i)$ .
3. If the computed hash value matches the stored hash value and the client's counter is the same as the server's counter, then the client is authenticated.

Using this request counter approach, the resulting authentication procedure is efficient. However, integrity protection of both counters is a must to prevent deny of service attacks. Including the client's counter inside the authenticator is to prevent replay attacks. A replay attack occurs if an adversary overhears an authenticator from a client's legitimate request and later uses the authenticator to impersonate the client to the server. With the request counter approach in the PASS scheme, an old authenticator will not have a matching counter to the one in the server.

If the authenticator contains only request counter, a milder replay attack could occur. Without  $CS_i$  inside the authenticator, an adversary may overhear the authenticator from a client  $i$  with a counter value  $c_i$  and later use this authenticator to impersonate another client  $j$  when his counter value  $c_j = c_i$ . Of course, the adversary needs to have the capability to keep track of the counter values of both clients  $i$  and  $j$ . Including the secret share  $CS_i$  inside the authenticator, the above minor replay attacks can also be detected.

## 6 Access Control

The access control in the PASS scheme plays another important defense for data privacy, especially for malicious employees inside the cloud. To strongly control accesses, the PASS scheme chooses the mandatory access control model, in which no access right delegation can be granted. To implement such access control in cloud computing, a good security labeling is essential.

A security label consists of two parts (*security level*, *{categories}*), which can be assigned to either data items or subjects. Assigning to a data item, the security level indicates the data's security sensitivity and the set of categories describes kinds of information of the data. While assigning to a subject, the security level is the subject's security clearance and the set of categories describes what kinds of data the subject has right to access. The U.S. DoD has defined four security levels for their applications: top secret, secret, confidential and unclassified. With assigned security labels, a subject is allowed to access a data item if his security clearance is higher than or equal to the data's security level and his set of categories is a super-set of those assigned to the data item. This model is well-known and mature. Thus, no further refinement to the model in the PASS scheme is conducted. However, in the context of cloud

computing, it requires to define appropriate security levels and the granularity of categories, as well as to carefully identify the subjects which may be involved in any cloud operation. All these variables should be considered as a whole so that the resulting security labeling is not complex but flexible enough for access control in the cloud.

To reduce the complexity, the PASS scheme only defines two security levels: secret and non-secret. Table 2 lists the identified subject types. All cloud implementations may require these top-

Table 2: Different subject types in cloud computing.

Clients	Should be allowed to access their own data only.
Query Servers	May include authentication, query, and encryption servers
Query Processes	Processes that are allowed to access clients' data.
Cloud Servers	All other servers in the cloud that are not related to query processing.
Cloud Processes	Processes that are not allowed to access clients' data.
Classified Employee	Few top trusted managers, cloud engineers, and system administrators. Able to revoke and check the status of query processes.
Unclassified Employees	Unclassified employees in the cloud. No right to perform any operations to the query processing black box.

level subject types. However, they can be further divided into subtypes. Different cloud applications could have different subtypes. Thus, each cloud implementation should define subtypes based on their own discretion. All query processes can be invoked by the query server only. They are all built-in functions and should not be altered in any circumstances unless the few top classified employees decide to update them to newer versions. Cloud processes are not designated to manipulate clients' data and they could be any process related to other operations in the cloud. Thus, these processes do not have any access right to clients' data.

Next, what granularity of category is appropriate in cloud computing? Obviously, data belong to different clients should be in different categories. Let  $C_i$  denote a category for each client

$i$ . For security levels, it's clients' responsibility to specify secret or non-secret to each data item. Thus, client  $i$ 's data could be either (*secret*,  $C_i$ ) or (*nonsecret*,  $C_i$ ). In addition, each client  $i$ 's profile is secret and labeled as (*secret*,  $C_i$ ). Data labeled secret needs to be encrypted all the time in the cloud. Finally, the front-end authentication server, and the back-end query server and encryption server are all assumed trusted with the highest label (*secret*, *query\_system*), where the category *query\_system*  $\supset$  {all  $C_i$ }. In order for these servers to be trusted, the cloud providers should put extra effort to secure them such as firewalls, intrusion detection, physical protection, or their mix. Only these query servers are able to derive encryption keys upon receiving clients' queries.

Another notable design for access control in the PASS scheme is to build a black box for query processing. This black box can only be accessed by some selected classified employees with the highest security label (*secret*, *query\_system*). All other unclassified employees, servers and processes that are not related to query processing should absolutely have no right to access anything inside the box.

## 7 The PASS Scheme

With the five security components, Figure 1 shows a diagram that gives an overall picture of the proposed PASS scheme. We walk through the diagram step by step. Step 1 to Step 4 describe how to store a client's data in the cloud. This process includes key agreement, data encryption, and profile creation/update. Step 5 through Step 12 list the sequence of operations for a query processing.

### Key Agreement and Data Encryption Steps:

1. The client  $i$  and the Authentication Server (AS) perform the key agreement procedure as described in Section 3. This procedure can be performed either for a periodic secret key update or when the client  $i$  first subscribes to the cloud.
2. AS creates or updates the client  $i$ 's profile to record the newly agreed key  $k_i$  and the secret share  $SS_i$ .
3. AS sends a data encryption request, along with  $k_i$ , to the Encryption Server (ES).
4. ES encrypts or re-encrypts client  $i$ 's data using  $k_i$ .

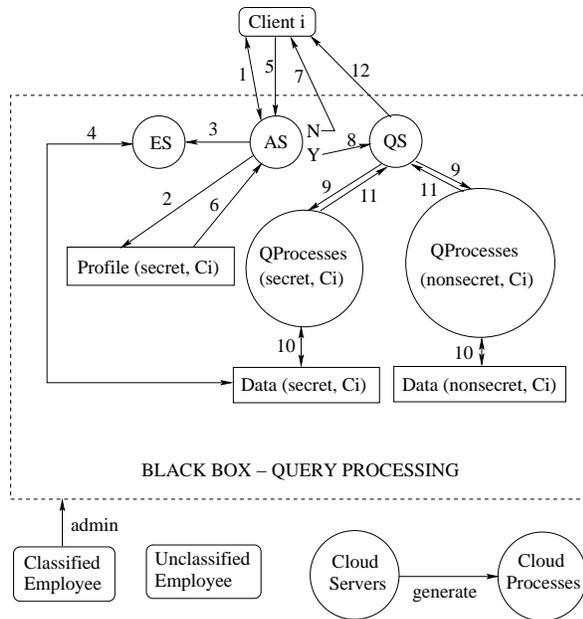


Figure 1: Query processing diagram, which is a black box to most employees in the cloud

### Query Processing Steps:

5. Client  $i$  makes a Query Request QR to AS. An authenticator (see Section 5) is sent along with the request.
6. Upon receiving a QR from the client  $i$ , AS performs the authentication procedure described in Section 5 by first retrieving  $SS_i$  and  $h(k_i)$  from the client  $i$ 's profile. The shared key  $k_i$  will be derived in this step.
7. If the authentication fails, AS either sends back a deny message to the client  $i$  or simply discard the request. AS should forget (i.e., discard)  $k_i$  from its storage right away in this case.
8. If the authentication succeeds, AS forwards the query, along with  $k_i$ , to the Query Server (QS).
9. Depending on the query, QS invokes a query process, which could be either QProcess ( $secret, C_i$ ) or QProcess ( $nonsecret, C_i$ ), to perform operations requested in the query. A secret QProcess will get  $k_i$  from QS.
10. The invoked QProcess may need to access some client  $i$ 's data. Based on the security labels defined in Section 6, QProcess ( $nonsecret, C_i$ ) is only allowed to access data

labeled ( $nonsecret, C_i$ ), whereas QProcess ( $secret, C_i$ ) is able to access data either labeled ( $nonsecret, C_i$ ) or labeled ( $secret, C_i$ ).

11. After the QProcess finishes processing, it returns a Query Result QRT to QS. This QRT should have the same security label as the QProcess. In case the QRT is secret, it will be encrypted before being returned to QS.
12. QS returns either encrypted QRT ( $secret, C_i$ ) or not encrypted QRT ( $nonsecret, C_i$ ) to the client  $i$ . Finally, all servers must discard  $k_i$  from their memory.

Note that data encryption/decryption may be performed by either the ES or a secret QProcess. ES is responsible to encrypt the whole database of a client  $i$  at the time a new shared key  $k_i$  is agreed, whereas each secret QProcess performs encryption/decryption to only related data in a query processing.

## 8 Conclusion

This paper presents a PASS scheme in cloud computing, which aims at protecting data privacy for clients. The scheme consists of five components as described from Section 2 to Section 6. Some designs of the scheme are innovative to the cloud computing field such as not storing data encryption keys anywhere in the cloud by secret sharing, using ECC for key agreement, using request counters and secret shares for authentication, forming a black box for query processing, as well as defining some top-level subject types in the cloud. A sequence of walk through steps for both the initial key agreement and a later query processing are also provided to show how these five components work together.

## 9 References

- [1] G. Boss, P. Malladi, D. Quan, L. Legregni and H. Hall, "Cloud Computing", *IBM Corporation*, 2007.
- [2] "Application Architecture for Cloud Computing", *rPath Inc.*, 2008.
- [3] "Securing Microsoft's Cloud Infrastructure", *Microsoft Global Foundation Services*, 2009.
- [4] "Cisco Cloud Computing - Data Center Strategy, Architecture, and Solutions - Point of view

White Paper for U.S. Public Sector”, *Cisco Systems, Inc.*, 2009.

[5] S. Bennett, M. Bhuller and R. Covington, ”An Oracle White Paper in Enterprise Architecture - Architectural Strategies for Cloud Computing”, *Oracle Corporation*, 2009.

[6] ”Amazon Web Services Launches - Amazon S3”, *amazon.com*, 2006.

[7] C. Wang, Q. Wang, K. Ren and W. Lou, ”Ensuring Data Storage Security in Cloud Computing”, *17th International Workshop on Quality of Service*, 2009.

[8] V.D. Cunsolo, S. Distefano, A. Puliafito and M.L. Scarpa, ”Achieving Information Security in Network Computing Systems”, *8th IEEE International Conference on Dependable, Autonomic and Secure Computing*, 2009.

[9] J. Harauz, L.M. Kaufman and B. Potter, ”Data Security in the World of Cloud Computing”, *IEEE Security and Privacy*, 2009.

[10] C. Wang, Q. Wang, K. Ren and W. Lou, ”Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing”, *Proceedings of IEEE INFOCOM*, 2010.

[11] B. Grobauer, T. Walloschek and E. Stocker, ”Understanding Cloud-Computing Vulnerabilities”, *IEEE Security and Privacy*, 2010.

[12] R. Rivest, A. Shamir and L. Adleman, ”A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, *Communications of the ACM*, 21(2), 1978.

[13] I. Black, G. Seroussi and N. Smart, ”Elliptic Curves in Cryptography”, *Cambridge University Press*, 1999.

[14] ”An Elliptic Curve Cryptography (ECC) Primer, why ECC is the next generation of public key cryptography”, *The Certicom 'Catch the Curve' White Paper Series*, 2004.

[15] ”Recommended Set of Advanced Cryptography Algorithms - Suite B”, *National Security Agency*, 2005.

[16] ”Announcing the Advanced Encryption Standard (AES)”, *Federal Information Processing Standards Publication 197*, 2001.

[17] A. Shamir, ”How to Share a Secret”, *Communications of the ACM*, 22(11), 1979.

[18] W. Diffie and M.E. Hellman, ”New Directions in Cryptography”, *IEEE Transaction on Information Theory*, IT-22, 1976.

# Watermarking-based Image Authentication with Recovery Capability using Halftoning and IWT

Luis Rosales-Roldan, Manuel Cedillo-Hernández, Mariko Nakano-Miyatake, Héctor Pérez-Meana

Postgraduate Section, Mechanical Electrical Engineering School, National Polytechnic Institute of Mexico

**Abstract** – *In this paper we present a watermarking algorithm for image content authentication with localization and recovery capability of the modified areas. We use a halftone image generated by the Floyd-Steinberg kernel as an approximate version of the host image. We adopt this halftone image as a watermark sequence and embed it using the quantization watermarking method into the sub-band LL of the Integer Wavelet Transform (IWT) of the host image. Due to the fact that the watermark is embedded into the sub-band LL of IWT, the proposed method is robust to JPEG compression. Moreover, we employ a Multilayer Perceptron neural network (MLP) in inverse halftoning process to improve the recovered image quality. Using the extracted halftone image, the gray-scale of the modified area is estimated by the MLP. The experimental results demonstrate the effectiveness of the proposed scheme.*

**Keywords:** Watermarking, Content Authentication, Recovery Capability, Integer Wavelet Transform, Multilayer Perceptron

## 1 Introduction

Nowadays the digital age has reached an important development in some technical fields, such as computer and the telecommunications. This development has a strong impact on the people's life, for example it is quite common to take pictures everywhere and every time using his/her cell phones with digital cameras. And also 700,000 pictures per hour are uploaded to any social network to be shared among friends. However these digital pictures can be easily modified using computational drawing tools, such as Photoshop, without causing any distortion. Considering that a scenario where some of these digital pictures could be required as evidence to prove the truth of the statement of a person who is defending his innocence on court, the integrity of these digital images becomes an urgent and important issue.

The Cryptographic Hashing, such as MD5 and SHA-1, have been used to authenticate the digital data, however

it cannot be used for digital images in an efficient manner. The main problem is that there are many different formats, such as JPEG, BMP, TIF, PCX and so on, to save a digital image and besides some of them have their different compression mode. For example, an image could be compressed and converted to another format during its distribution. Although image format or compression mode is changed, the content of the image is conserved totally. Taking these aspects under consideration, the Cryptographic Hashing, digital fingerprinting and other techniques, which cannot tolerate the content conserving modifications, are not adequate for image content authentication.

Among several approaches, a watermarking-based approach is considered as a possible solution. Early image authentication methods [1] result in an integrity decision, which indicates only if the image under analysis is authentic or not. The watermarking-based authenticators can be classified into two schemes: fragile watermarking-based scheme [2] and semi-fragile watermarking-based scheme [3, 4]. The fragile watermarking scheme can be used for complete image authentication in which only those images without any modification are considered as authentic. On the other hand, the semi-fragile watermarking scheme can be used for content authentication, in which those images, that are modified no intentionally and conserved its original content, are considered as authentic. Consequently, content authentication scheme must be robust to content-preserving modification, such as JPEG compression.

Many content authentications methods determine if image has been modified or not, and some of them can localize the modified areas [3]; moreover, only a few schemes have the capability to recover the modified area without using original image [4-8]. In [4] they divide the image into sub-blocks and then mapping the sub-block with a secrete key. With this, a watermark bits sequence is formed by a compressed version of an image block, which is extracted for the quantized DCT coefficients, and then it's embedded into two LSB's of the corresponding image block. This method is classified as a vulnerable scheme to non-intentional modifications, such as image compression, contamination by noise, etc. In [5] they used halftone

representation of the original image as watermark sequence and embedded it into LSB plane of the image. Due to embedding domain is spatial LSB; also this scheme is not robust to JPEG compression. In [6] the authors proposed a hybrid block-based watermarking technique, which includes robust watermarking scheme for self-correction and fragile watermarking scheme for sensitive authentication. In this scheme all alterations, including the content-preserving modification, are detected and the recovery mechanism is triggered; therefore the quality of the final recovered image can be affected. To increase watermark robustness, [7] introduced a concept of region of interest (ROI) and region of embedding (ROE), and the original image is segmented into these two regions. Information of ROI is embedded into ROE in DCT domain. In this scheme, the size of ROI is limited for correctly operation, and for some types of images the segmentation of ROI and ROE can not be done in advance. Due to the fact that the quality of the image is important for further process, in [8] the authors proposed a new watermarking method consisting of the detection and recovery of the modified areas. They used a halftone image from the original one as a watermark sequence and embedded it into the Discrete Cosine Transformation (DCT) using the Quantization Index Modulation (QIM). The QIM applied in DCT domain makes their method be robust to JPEG compression. Also they used a Multilayer Perceptron neural network (MLP) to obtain a high quality of recovered image.

In this paper, we proposed an image authentication and recovery scheme in which a halftone of the original image is embedded as a watermark into the image using quantization-watermarking algorithm. Unlike [8], which used DCT embedding method, in the proposed scheme the watermark sequence is embedded into the sub-band LL of the Integer Wavelet Transform (IWT) domain to increase watermark robustness. We used a similar process of the embedding process in the authentication stage, if the extracted halftone image matches with the embedded one, the image is declared as authentic, otherwise the altered area can be detected and then the recovery process is started to estimate the original gray-scale image of the altered area from the extracted halftone image using the previously trained MLP.

The rest of this paper is organized as follows. Section 2 describes the proposed algorithm and experimental results are presented in Section 3. Finally Section 4 concludes this work.

## 2 Proposed Algorithm

The proposed authentication algorithm is composed by three stages: self-embedding, authentication and recovery stage.

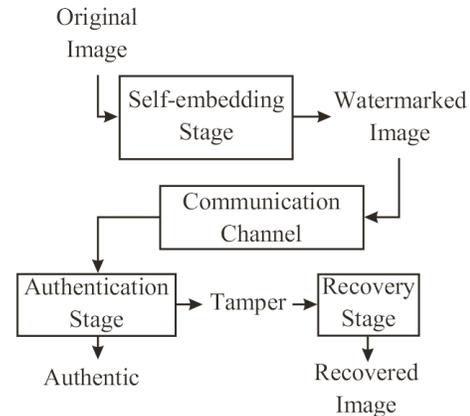


Figure 1. General scheme of the proposed algorithm.

### 2.1. Self-Embedding Stage

In the self-embedding stage, the original image is down-sampled with half size in height and width to generate the watermark sequence. Then we applied the error diffusion halftoning method proposed by Floyd-Steinberg to the down-sampled image to get halftone image. The halftone image is permuted by the chaotic mixing method [9] using user's secret key. On the other hand, the original image is decomposed using the IWT to obtain four sub-bands: LL, LH, HL and HH. The permuted halftone image is embedded into the sub-band LL using quantization watermarking method [10]. The embedding algorithm is given by:

$$w_k = \begin{cases} \tilde{c}_{ij} = v_1 & \text{if } |c_{ij} - v_1| \leq |c_{ij} - v_2| \\ \tilde{c}_{ij} = v_2 & \text{otherwise} \end{cases} \quad (1)$$

where

$$v_1 = \begin{cases} \text{sign}(c_{i,j}) \times \left\lfloor \frac{|c_{ij}|}{2S} \right\rfloor \times 2S, & w_k = 0 \\ \text{sign}(c_{i,j}) \times \left( \left\lfloor \frac{|c_{ij}|}{2S} \right\rfloor \times 2S + S \right), & w_k = 1 \end{cases}$$

$$v_2 = v_1 + \text{sign}(c_{ij}) \times 2S$$

and  $w_k$  is the  $k$ -th watermark bit,  $c_{i,j}$  and  $\tilde{c}_{i,j}$  are the original and the watermarked IWT coefficients, respectively, and  $S$  is the quantization step size. Finally we obtained the watermarked image applying inverse IWT to the watermarked LL sub-band and the rest of the sub-bands (LH, HL and HH). This stage is shown in Figure 2.

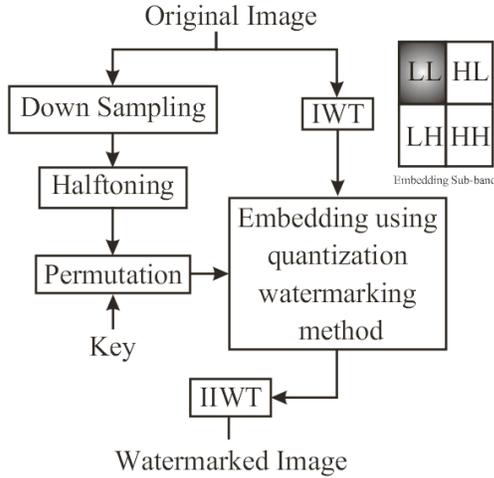


Figure 2. Self-embedding stage.

## 2.2. Authentication Stage

In the authentication stage (see Figure 3) firstly the watermark is extracted for the sub-band LL of the suspicious image and the extracted bits are reordered using the user's secret key given in the embedding stage. The watermark extraction process is given by:

$$\tilde{w}_k = \begin{cases} 0 & \text{if } \text{round}\left(\frac{\hat{c}_{ij}}{S}\right) = \text{even} \\ 1 & \text{if } \text{round}\left(\frac{\hat{c}_{ij}}{S}\right) = \text{odd} \end{cases} \quad (2)$$

where  $\tilde{w}_k$  is extracted watermark bit, and  $\hat{c}_{i,j}$  is IWT coefficient of LL sub-band of the watermarked and possibly modified image.  $S$  is the same quantization step size used in embedding stage. The reordered watermark sequence is the halftone version of the original image and then it is converted to gray scale image using a Gaussian low-pass filter given by (3).

$$F_g = \frac{1}{11.566} \begin{bmatrix} 0.1628 & 0.3215 & 0.4035 & 0.3215 & 0.1628 \\ 0.3215 & 0.6352 & 0.7970 & 0.6352 & 0.3215 \\ 0.4035 & 0.7970 & 1 & 0.7970 & 0.4035 \\ 0.3215 & 0.6352 & 0.7970 & 0.6352 & 0.3215 \\ 0.1628 & 0.3215 & 0.4035 & 0.3215 & 0.1628 \end{bmatrix} \quad (3)$$

Next we generate a halftone image from the suspicious watermarked image and it is re-converted in a gray-scale image using the same Gaussian low-pass filter. This inverse halftoning is the simplest method, even though it produces low quality gray-scale image. In this stage, an accurate detection of the modified areas is important; therefore high quality of the gray-scale image is not necessary. Then both images (gray-scale image generated

from the extracted watermark sequence and gray-scale image generated from suspicious watermarked image) are compared each other to localize the modified areas. To do this we employed a block-wise strategy, in which the comparison is carried out in each block of  $N \times N$  pixels and the mean square error (MSE) of each block is calculated by (4) and it is compared with a predetermined threshold value  $Th$ .

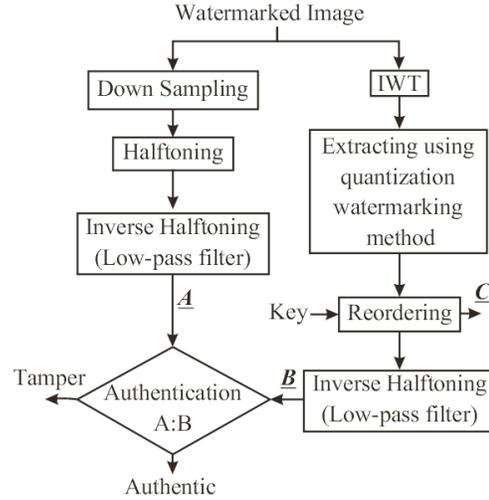


Figure 3. Authentication stage.

$$D = \frac{1}{N_2} \sum_{i=1}^N \sum_{j=1}^N (A(i,j) - B(i,j))^2 \quad (4)$$

where A and B are the blocks of gray-scale image in Figure 3, respectively, and  $N \times N$  is a block size. If  $D \geq Th$  the block is considered as tampered, otherwise the block is authentic.

## 2.3. Recovery Stage

If the authentication stage shows that some blocks of the suspicious image are tampered, then the recovery stage will be triggered. In this stage we will use as input data, the down-sampled suspicious watermarked image, its halftone version, the information about modified blocks and the extracted halftone image (signal C in Figure 3). In this stage we firstly use the down-sampled suspicious image and its halftone version to train MLP by the Backpropagation (BP) algorithm. This recovery stage is shown in Figure 4 and the MLP used to estimate the gray-scale image is shown in Figure 5.

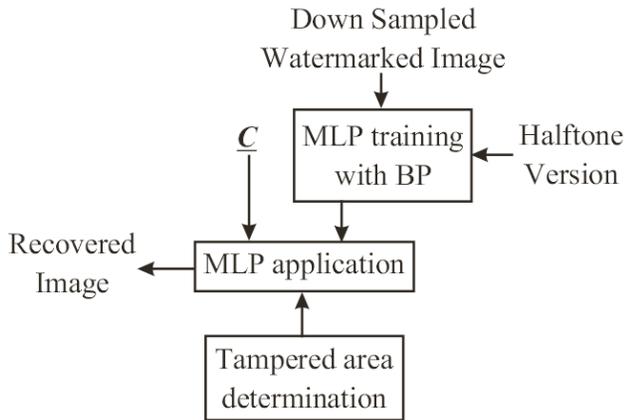


Figure 4. Recovery process.

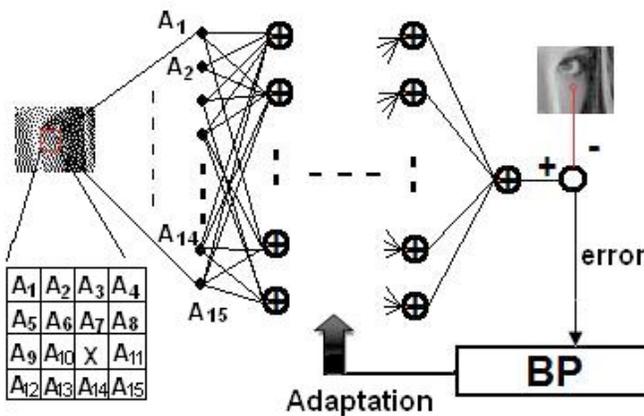


Figure 5. MLP used to estimate the gray-scale image

The 4x4 neighborhood template, shown in bottom-left part of the Figure 5, composed of 16 binary pixels including the center pixel "X", is used to get an input pattern of MLP. The output data is a gray-scale estimated value of the corresponded center pixel "X". The extracted halftone image of the modified area is introduced to this MLP to get a better quality of the recovered region.

In the general case of inverse halftoning, the gray-scale image is not available, therefore the MLP-based inverse halftoning is meaningless, however in this case the non-modified area of the suspicious gray-scale image is available. So we can use the halftone and the corresponded gray-scale image of this non-modified area to generate a high quality image using MLP-based inverse halftoning. Figure 6 shows a comparison between images obtained using Gaussian low-pass filter and MLP.

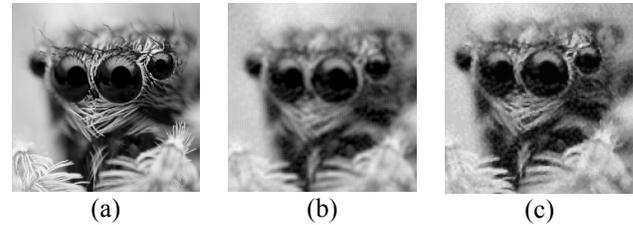


Figure 6. Image quality comparison. (a) Original Image. (b) Gray-scale image by Gaussian low-pass filter (24dB). (c) Gray-scale image by MLP (27dB).

The PSNR of both images respect to the original one are 24 dB and 27 dB, respectively, which indicates that the image generated by MLP can conserve more details of the original image than the gray-scale image generated by a Gaussian low-pass filter.

### 3 Experimental Results

To evaluate the performance of the proposed watermarking scheme, the watermark imperceptibility and robustness are assessed using several images. It is very important to select an adequate value of the quantization step size used in the embedding algorithm, because this value has serious effects on the watermark imperceptibility and robustness. In Figure 7 we show the relationship between watermark imperceptibility and the quantization step size for each sub-band decomposed by IWT. As we can see in the Figure 7, highest sub-band HH shows better watermark imperceptibility compared with other sub-bands. Furthermore the lowest sub-band LL can be used as watermark embedding domain if the step size is lower than 7 from watermark imperceptibility point of view. Considering the watermark robustness, we select the lowest sub-band LL as watermark embedding domain together with step size value 7.

Also, in Figure 8 we show the relationship between quality factor of JPEG compression and BER of the extracted watermark sequence respect to the embedded one. In which the performance of different step sizes are compared. In all cases, the watermark sequence is embedded in the lowest sub-band LL. From Figures 7 and 8, we select the value 7 for the quantization step size. Also the selection of the threshold value  $Th$ , to determine if an area is altered or not, is very important. Figure 9 shows the relationship between the false alarm error rate ( $Fa$ ) and the threshold value when the watermarked image is compressed with JPEG compressor using a quality factor 80. From this figure, the threshold value 0.001 is considered as the best one.

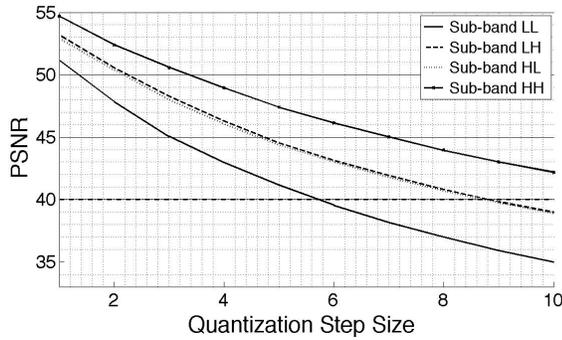


Figure 7. Relationship between quantization step size and PSNR of watermarked image respected to the original one.

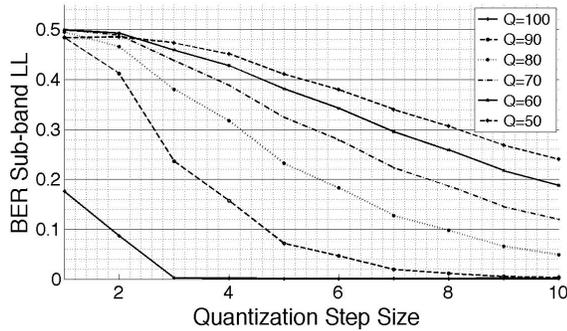


Figure 8. Relationship between quality factor of JPEG compression and BER of extracted watermark respected to the halftone original image.

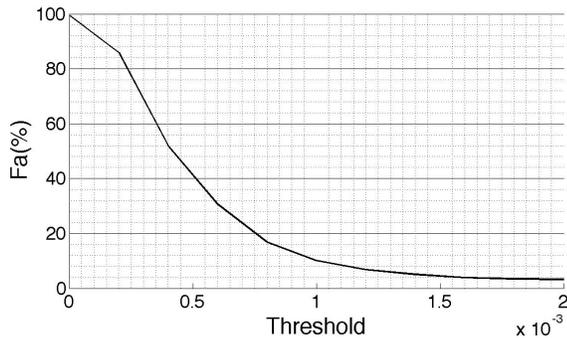


Figure 9. Relationship between threshold value and false alarm error rate with quantization step size equal to 7.

Figure 10 shows the original and the watermarked image generated by the proposed algorithm using step size equal to 7. Here, the average PSNR of the watermarked images respect to their original one is 38.17 dB.

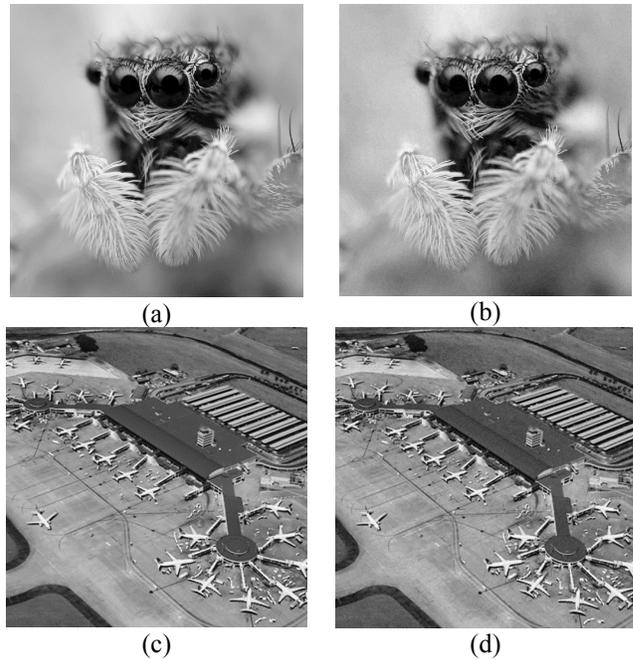


Figure 10. (a), (c) Original Images. (b), (d) Watermarked Images.

Now, the Figure 11 shows an example with modified area; in this case we add extra object to the image and the proposed algorithm is applied to detect and recover the modified area. Figure 12 shows another example with different modified area. In this case we erased an object from the image and the proposed algorithm is applied to detect and recover the modified area. From these figures, the modified areas are detected and recovery correctly.

## 4 Conclusions

In this paper, an image authentication algorithm with recovery capability is proposed, in which a halftone version of the original image is used as a watermark sequence and it is embedded using quantization watermarking method into the LL sub-band decomposed by the IWT.

Important factors, such as the step size value of the embedding algorithm and the threshold value used in the authentication process are estimated taking into account the watermark imperceptibility, robustness and false alarm error rate. The average PSNR of several watermarked image respect to their original versions using an adequate step size value indicates that the embedded watermark is imperceptible by Human Visual System. Also simulation results showed that the embedded watermark is robust to JPEG compression with a quality factor larger than 80%.

The use of the MLP trained by BP algorithm increases the quality of the recovered image and the simulation results showed that the proposed method can detect and recover correctly the modified areas.

## 5 References

- [1] J. Dittmann, "Content-fragile Watermarking for Image Authentication", Proceedings of SPIE, vol. 4314, 2001, pp. 175-184.
- [2] P. Wong, N. Memon, "Secret and Public Key Image Watermarking Scheme for Image Authentication and Ownership Verification", IEEE Trans. Image Processing, vol.10, no.10, 2001, pp. 1593-1601.
- [3] K. Maeno, Q. Sun, S. Chang, M. Suto, "New Semi-Fragile Image Authentication Watermarking Techniques Using Random Bias and Nonuniform Quantization", IEEE Trans. Multimedia, vol. 8, no. 1, 2006, pp. 32-45.
- [4] J. Fridrich, M. Goljan, "Image with Self-Correcting Capabilities", 1999 Int. Conf. on Image Processing, vol. 3, 1999, pp. 792-796.
- [5] H. Luo, S-C Chu, Z-M Lu, "Self Embedding Watermarking Using Halftone Technoque", Cicut Systems and Signal Processing, vol. 27, 2008, pp. 155-170.
- [6] Y. Hassan, A. Hassan, "Tampered Detection with Self Correction on Hybrid Spatial-DCT Domains Image Authentication Technique", Communication Systems Software and Middleware Workshops, 2008, pp. 608-613.
- [7] Clara Cruz, Jose Antonio Mendoza, Mariko Nakano, Hector Perez, Brian Kurkoski, "Semi-Fragile Watermarking based Image Authentication with Recovery Capability", ICIECS 2009 pp. 269-272.
- [8] Jose Antonio Menodza-Noriega, Brian M. Kurkoski, Mariko Nakano-Miyatake, Hector Perez-Meana, "Halftoning- based Self-embedding Watermarking for Image Authentication", 2010 IEEE Int. 53<sup>rd</sup> Midwest Symposium on Circuits and Systems, 2010, pp. 612-615.
- [9] G. Voyatzis, I. Pitas, "Embedding Robust Watermarks by Chaotic Mixing", Int. Conf. on Digital Signal Processing, vol. 1, no. 1, 1997, pp. 213-216.
- [10] B. Chen, G. Wornell, "Quantization Index Modulation: A class of provably good method for digital watermarking and information embedding", IEEE Trans. On Information Theory, vol. 48, no. 4, 2001, pp. 1423-1444.

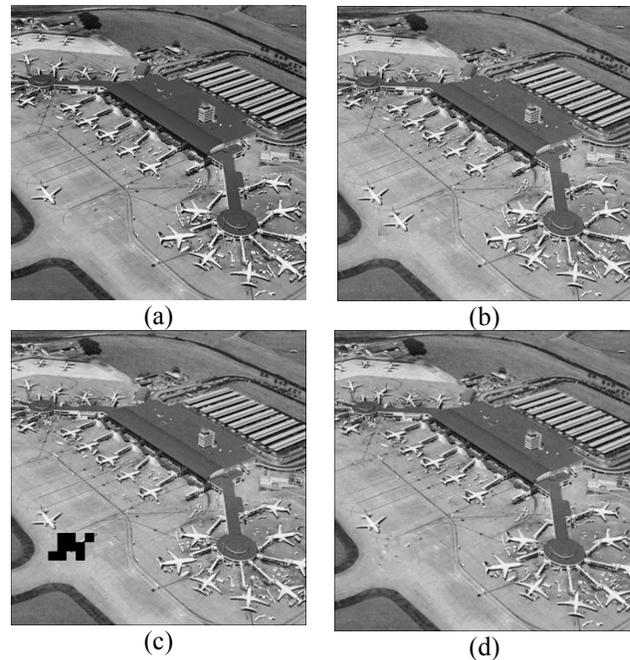


Figure 11. (a) Original Image. (b) Suspicious image, adding extra information. (c) Modified area detection. (d) Recovered image

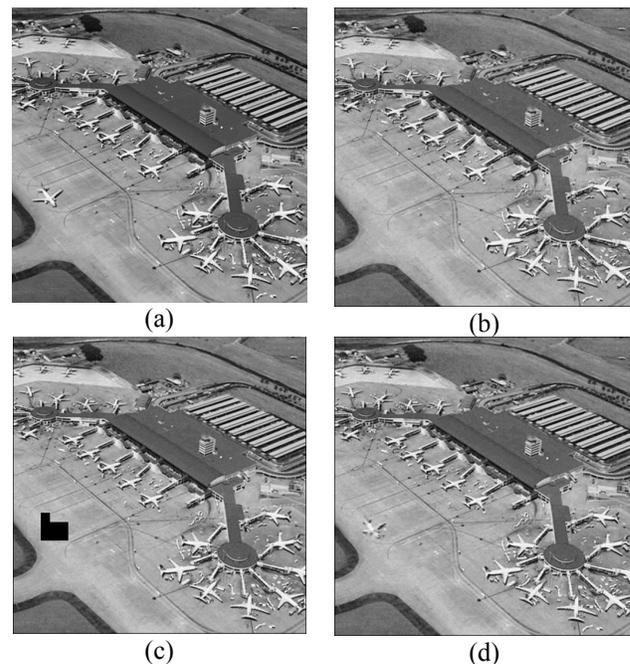


Figure 12. (a) Original Image. (b) Suspicious image, extracting some information. (c) Modified area detection. (d) Recovered image.

# Kerberos, Cryptography and Biometric based Remote Authentication Protocol

Desai Karan Ronak<sup>1</sup>, Ruchir Patwa<sup>2</sup>

<sup>1</sup>School of Information Technology and Engineering, VIT University, Vellore, India

<sup>2</sup>School of Computing Sciences, VIT University, Vellore, India

**Abstract** – We are looking for a very secure method of remote authentication. Biometrics authentication has become popular with the increase in infrastructure facilities and scope of sensor technologies. They are suited due to high security in applications like remote authentication. We are considering a provably secure and blind sort of biometric authentication protocol combined with the advantages of Kerberos's ticket granting. We are using cryptography to make the protocol more secure. It can successfully run over public network for remote access. It can also be implemented to take care of the revoking of registered templates. It is not biometric specific. The main Kerberos part comes in because of the ticket granting mechanism. Kerberos and biometrics are already proven to survive range of attacks. Finally we show a central, already secure, server that can be used for mass authentication and a wide range of applications.

**Keywords:** Biometric authentication, encryption and biometrics, biometric Kerberos.

## 1 Introduction

During remote connection we face certain challenges when it comes to security. The system we are going to propose for overcoming these security issues are using biometrics first of all which is cost effective and secure [1][2]. Thus if such technology/hardware is available at the user end it would be very much efficient for the secure authentication as we will see ahead. Though we are not using remote specifications at all points, we are addressing security issues based on it. We have to take care about a lot of security issues here [3]. We are implementing a blind authentication protocol [4], using Kerberos sort of methodology. This blind authentication crypto-biometric authentication protocol [4] that the authors of the previous work had proposed is the main base and idea of our scheme for remote authentication. We know that how Kerberos has been successful as an authenticating protocol. What we wish to do is make it more secure by integrating it with a crypto-biometric authentication system in place of the password system that Kerberos implements. We will see what problems in Kerberos have been addressed and solved by this kind of methodology. Biometrics authentication has become popular with the increase in infrastructure facilities and scope of sensor technologies. There is a unique way of solving problems for each of these two methodologies by combining them. The user's actual biometric data is also not available with the authenticating server. It's only submitted to the

registration server. The encryption and biometrics with the registration server, authentication server and the token granting server makes this technique unique. It can guard against almost all kind of possible threats in the scenario. We take care of a) Biometric template security b) Privacy of the user c) Trust between user and authenticating server and d) Network security related issues [5]. The previous works were generally based on system that provided security by securing the secret key by biometrics. The proposed system does not follow this lead. We divide our task into three steps 1) Registration 2) Authentication 3) Ticket granting. We will target on strong cryptography for user's original data with the registration server, obviously the authentication should be non-reputable and also the user side attacks and the replay attacks should be taken care of, also in the cases where say the key is compromised. The high performance needed by this level of crypto-biometric system is solved by the token granting system of Kerberos while biometrics takes care of some of the security issues that Kerberos has not been able to solve. In the proposed method we look towards the design of a classifier that also helps us to improve the performance of biometrics and we use the randomization scheme for this purpose. Our main improvements in the previous methods are thus benefits of ticket granting and the single registration and multiple authentication as explained in during the application.

## 2 The Authentication Protocol

The overall authentication procedure as explained is divided into the following three major steps i.e. registration, authentication and ticket granting. A remote client first registers himself, i.e. enrolls himself with the first server. Then it authenticates himself with the authentication server proceeding further with the tickets and session keys. Figure 1 shows how the process is divided for the three servers. Alice is our Client that is situated remotely and wants to access Bob from there. We are assuming that she has that hardware required for biometrics. Alice remotely invokes and authenticates herself to access Bob initially. Alice has to go to a three step procedure initially. Then once a ticket is obtained from the ticket granting server she may skip the initial steps for certain time period because the ticket will work until it expires. Also note the authentication scheme that we are going to publish. We will follow the modulo-operations i.e. all the operations like  $M \text{ operation } N$  are carried out in the encryption domain using the expression  $(M \text{ operation } N) \bmod P$ .  $P$  will be decided by the encryption scheme we employ. Any encryption method can be used but we must be sure it

follows homomorphism for certain operations as explained later during authentication.

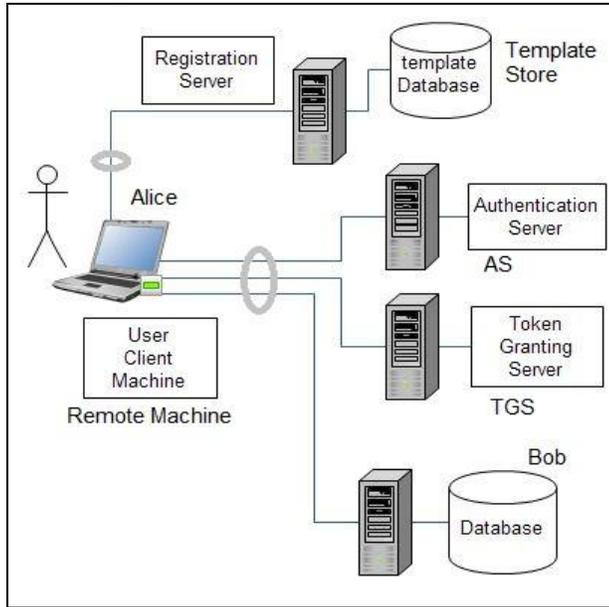


Figure1. The authentication mechanism

## 2.1 Registration Server

This is the basic step of registering the user with the main registration server that has the templates of biometrics provided by all users. The registration server is the trusted server here. We are here assuming that this third party server i.e. the registration server is already safe enough for us.  $K$  is the public key of Alice that it tell the server. During the registration, the client/Alice sends samples of her biometric data to the registration server, which generates the classifier for Alice. The parameters generated by the registration server are encrypted and sent back to Alice. The biometric sample from Alice to registration server was digitally marked by the client and encrypted using the public key of the server to protect it hence making it secure. Finally what we send to the authentication server is the Alice's identity, her public key, the encrypted parameters and the threshold value. The algorithm1 shows how the process of registration takes place.

### 1: Registration

1. Alice collects biometric information on the available hardware device .
2. Alice creates the data  $x$  from vectors obtained from the biometrics.
3. Alice sends the data  $x$  with her identity and her public key  $K$  to the registration server.
4. Registration server uses  $x$  to compute a parameter  $(w, t)$  for the user.
5. These parameters are encrypted using Alice's public key:  $K(w)$

6.  $K(w)$  that are generated with Alice's identity, public key  $K$  and threshold  $t$  are sent to authentication server.
7. Alice is notified about the successful registration process.
8. The connection with the registration server is terminated. (Authentication server is approached for further process)

## 2.2 Authentication Server

Now we need to compute a value  $w \cdot x$  that requires multiplication. We can consider simple scalar multiplication and then the addition of the values obtained. Over here we are calculating  $x$  based on the vector values  $y_i$  that we may obtain from the biometrics. For the sake of simplicity we convert this vector to a finite quantity and single value  $x$  rather than dealing with  $i$  values of a single vector derived from the biometric.  $x$  can be the mean of the vectors or Note that we are using RSA in this method, we know that it follows homomorphism for multiplication[6]. Hence we can compute  $K(w \cdot x) = K(w) \cdot K(x)$  at the server side because of this property of homomorphism that RSA follows. Though we cannot add the results to compute the authentication function making it safe. Sending the product answers to Alice to do the addition actually reveals the classifier parameters to the Alice, which obviously we do not want. We are using a randomization technique for this purpose. We generate the parameter  $r_j$  by such randomization. It makes sure that the Alice can do the summation computing while it is not able to decipher any information from the product that she can get hands on. The randomization is done in a way such that the server can compute the final sum to be compared with the value of threshold that was decided earlier. The server here carries out all of its computation in the encrypted domain, and hence does not get any information about the biometric data ( $x$ ) or classifier parameter ( $w$ ). No one can guess our classifier parameters from the products as they are randomized when multiplied with  $r_j$ . The server is able to compute the final sum  $S$  because of the imposed condition on  $r_j$  and  $t_j S$ .

$$\sum_{j=1}^a (k_j r_j) = I \quad (1)$$

This condition as shown in equation 1 is what we have been able to imply to calculations as shown in the next set of equation. We should note that the ability of the server to generate random number here which actually define the privacy of the server. Substituting the equality in the final sum i.e.  $S$  we get the following

$$\begin{aligned} S &= \sum_{j=1}^a (k_j S_j) \quad (2) \\ &= \sum_{j=1}^a (k_j w \cdot x \cdot r_j) \end{aligned}$$

$$=(w x) \sum_{j=1}^a (k_j r_j) = w x \quad (3)$$

The process of authentication follows the steps shown in algorithm2. This products expression is the only thing that the server is able to obtain. This will reveal if the biometric belongs to the Alice or not while it does not actually reveal the biometric data which may sacrifice the security. It hence provides complete privacy to the user and the biometric data are not stored at any place temporary for template matching. Whatever is revealed is such that if obtained by an untrusted third party cannot be used in a way that it can harm.

**2: Authentication**

1. Alice computes  $K(x)$  and sends to the server
2. Authentication server computes  $a$  random numbers,  $r_j$  and  $k_j$  such that they satisfy the condition  $\sum_{j=0}^a (k_j r_j) = 1$
3. Authentication server computes  $K(w x r_j)$   
 $= K(w) K(x) K(r_j)$ .  
 (because of the homomorphism)
4. The products obtained are sent to the Alice.
5. Alice decrypts the product to obtain  $w x r_j$
6. Alice returns  $S_j = w x r_j$  to the server.
7. Authentication server computes  $S = \sum_{j=0}^a (k_j S_j)$  and checks if  $S > t$ , if true then server issues to Alice  $K_{TG}(Alice, K_s)$  and  $K_s$  encrypted with  $K_A$ . Where  $K_A$  is Alice's key,  $K_s$  is the session key and  $K_{TG}$  is the Ticket granting server's key.

**2.3 Token Granting Server**

TGS or the ticket granting server issues a ticket for the Real server (Bob) that Alice wants to access. It provides with the session key  $K_{AB}$  between Alice and Bob. Ticket granting adds to performance factor of the Kerberos environment with our combination. The Kerberos ticket is a certificate issued by an authentication server, encrypted using the server key. Among other information, the ticket contains the random session key that will be used for authentication of the principal to the verifier, the name of the principal to whom the session key was issued, and an expiration time after which the session key is no longer valid. The ticket is not sent directly to the verifier, but is instead sent to the client who forwards it to the verifier as part of the application request. Because the ticket is

encrypted by the server key, known only by the authentication server and intended verifier, it is not possible for the Alice to modify the ticket without detection. We already have an authentication system that quite space and time consuming and this ticket granting will help us reduce that factor. Although Alice verifies the ID just once with the authentication server, she can contact TGS multiple times for different servers and alternatively access the same server again and again. This benefit covers some part of the lag that we may face in the biometric authentication technique that we are suggesting.

**3 : Token Granting**

1. Now TG Server sends two tickets containing
2.  $K_s(Bob, K_{AB})$  and  $K_B(Alice, K_{AB})$ .
3. Alice sends Bob's ticket timestamp encrypted by  $K_{AB}$  i.e.  $K_{AB}(T)$  and  $K_B(Alice, K_{AB})$  it received from TG.
4. Bob confirms with Alice by a response such as  $K_{AB}(T+1)$  and confirms the success in ticket granting.

Hence once this ticket is granted no need to authenticate again and again and we can thus increase the performance of the biometrics. Figure2 shows how the overall process of authentication is done and summarizes the same. It gives an idea about the steps that are carried on the client's side and on the side of the three servers.

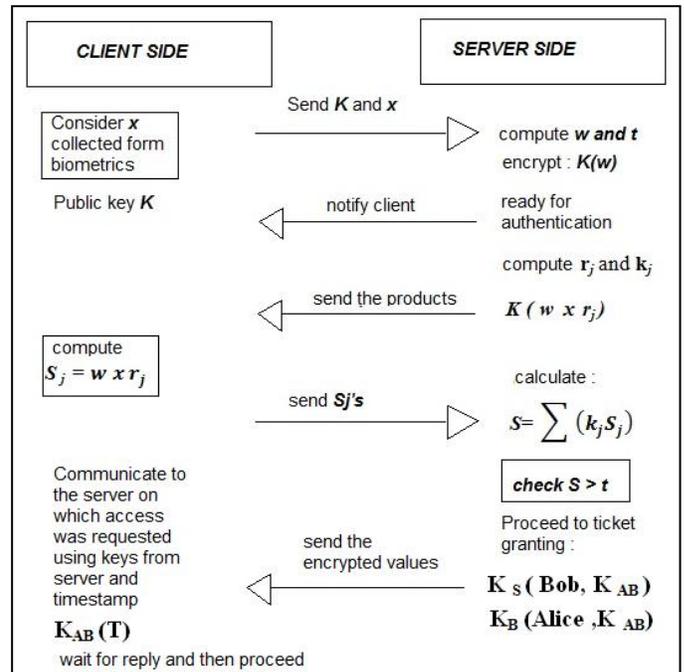


Figure2. Sequential representation of the process.

### 3 Security and Privacy Provided by the Authentication

We analyze all the scenarios to see how the security risks are handled by this authentication technique. First of all the client is to be verified and then the user. The user also has the risk of identity theft due to an unidentified or unsecure server. The database containing the user's information and authentication templates is at risk [3], having the critical information. And finally the network security is to be kept in mind because we will be using an unsecure network.

- a) *The hacker gains access to the template database* - In this case we know that the templates are encrypted by the public key of the respective clients. Hence it's hard to crack the public key algorithm. Moreover if the template is leaked then a new can be created from the new public-private key encryption algorithm. Even brute force for this would be almost impossible given the chances of getting a hit.
- b) *Hacker is in the database server during authentication* – hence the hacker has the total view of the protocol and how things are working. But the hacker cannot learn anything from the  $w x$  or  $x$  values. He can only obtain the  $S_j$  values from which it is almost impossible to derive the original biometric data. It may reveal some information about  $w x$  but still most part of the biometric will remain protective. Even if the hacker is in the server over multiple authentication trial by the same user he will have only multiple values of  $S_j$ . However the values of  $x$  will slightly change during multiple tries. Now his problem is the approximate calculation of  $w x$ . Thus the two points cover how the server will be protected.
- c) *On the client side if the Hacker gains access to the user's biometric or private key* – Over here we should note that we are considering the advantages of not only the biometric authentication but also the security of PKC. He needs the private key of the user to understand the biometric information if somehow he gets his hands on the user's biometric. In practice the private key can be stored on a smart card or such a hardware device to increase security and it is very rare to get both these. Even then if the hacker is successful then it will only affect one user and doesn't mean a threat to the whole system.
- d) *A passive kind of attack on the user's computer* – Hacker is present in user's computer during the login process. But the private key is on the hardware and has no direct access to it. He will thus only come to know the intermediate computation values. He will have  $a$  values with more variables. An effort equivalent to brute force will be needed in this case. Though multiple login attempts can help the hacker to succeed in this way. Though he would not be able to perform an authentication without the private key.
- e) *Network security* - In this case the network can easily be secured using standard cryptographic methods like symmetric cipher and digital signatures. All traffic is encrypted either by clients public key or random number by the server. Thus no information will be deciphered. No replay attack is possible due to the use of random number generation.
- f) *Risks that Kerberos faces* – Kerberos makes assumptions that the servers are secured and the password guessing attack is not possible. Kerberos V also implicitly relies on the servers being secure and software being non-malicious[7][8][12]
- g) The concerns of being tracked at any case during the authentication and revealing personal information to the intruder are secured by the fact that we use different keys for all the three application servers.
- h) *Loose synchronization* - The loose synchronization [7][9] that needs to be done for Kerberos V to avoid replay attack is also not a problem when it comes to our model. Replay attack is taken care by itself as explained earlier.
- i) *The password theft problem* – This problem that Kerberos authentication is vulnerable to [8][9][11] is solved by the method because of the use of crypto-biometric data. Kerberos does not protect against the theft of a password for example say through a Trojan horse login program on the user's workstation.
- j) *DSA and Fast RSA* – The RSA algorithm applied in the protocol can be substituted by either of the DSA or the Fast RSA, using Montgomery algorithm. This will enhance the security provided by the mechanism by several folds.

### 4 Application

*Applications using Multiple AS, One RS* - As we learned from the proposal we can establish a central registration server and once the user is registered the templates are safe with this server. We can now have a lot of remote connections all being authenticated at this single server. We can include a wide area of authentication like a state or even a country. What we need is one time registration on the users end and then through whosoever's remote link he is connecting his biometrics can be authenticated. Many companies and organizations can share such servers and save a huge amount of spending on such a secure server. Then they can all just establish their own authentication server and use the benefits of such a scheme. This will be both economical and effective. Though this main server will need to have a very high tolerance and performance curves, but it is achievable. Also it will give a great amount of security that many small firms may not be able to implement due to economic and other reasons. This feature can be enhanced in many ways and gives a lot of possibilities. As shown in Figure3 there is one trusted central server. We have many Authentication(AS) and ticket granting server(TGS) pairs linked to such central server. The central registration server will need to be something like a server farm. Different authentication servers hold access to the

different databases and other application servers that the user needs remotely. He will have to request for the specific authentication server he wants to access and based on that its request will proceed. This has real time applications like unique identity management and mass authentication systems at public places. Various corporate level authentication mechanisms for employees using remote connections through mobile devices. The algorithm4 shows how this is applied. Keys and variables are explained in the table itself.

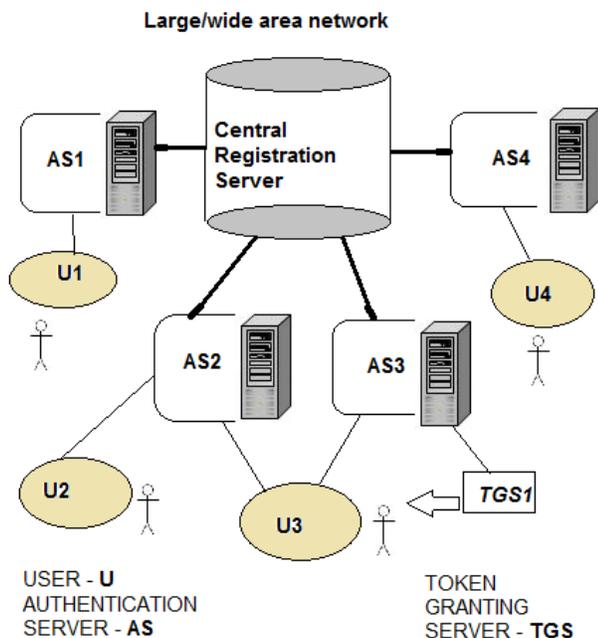


Figure3. Application in large networks

#### 4: Multiple AS, One RS application

1. Users  $1..n$  send their data for registration.
2. User  $i$  calls for authentication .
3. It sends its data collected from biometric samples ,  $x$  with its authentication server  $A_i$  , both encrypted by the registration server's public key  $R$  .
4. The registration server computes  $(w_i, t_i)$
5. Encrypts  $w_i$  ,  $t_i$  and  $K$  with the  $A_i$ 's public key ,and sends to  $A_i$  for authentication.
6. Now the authentication server takes over and direct communication between the user and authentication server takes place and the relative ticket granting server  $T_i$  will do the work of assigning the token .

## 5 Basic Simulation

For evaluating the verifier of our system we implement it based on a very simple client-server architecture using GNU-C. We use simple Linux C or GNU-C compiler for our purpose. First we establish a one client and multiple server, in our case being 3, for the process. This client-server socket connection is done using the TCP-IP method, i.e. connection oriented. We can also test it for connection less UDP protocol using simple network programming. We carry out the bind, connect, accept functions as needed. RSA keys can be generated using the implementation available through XySSL libraries[4]. All experiments can be performed on a average configuration workstation of Intel Corei3 processor, with 1.5GB of RAM. The performance of Kerberos is already tested and verified and we are not making any changes with the basic authentication of Kerberos that may affect performance. The biometrics once collected are in form of vectors that act as simple data vectors and do not give rise to performance issues. Though the initial process of biometric identification on the the client side is a considerably heavy process but it is necessary and doesn't affect our servers. The simulation for remote login process can also be carried out using the Opnet simulator made available by IT guru . It shows how a performance of remote login would work in such an environment having three servers and on client. We can use Ethernet servers and client and a Ethernet Hub to connect them. Make a remote login application and a simple guest user in the simulator to verify the results.

## 6 Future Work

Though the system is very apt for security we can increase the factor of authentication by adding a smart card methodology . This can help us improve the mechanism in many ways. This method gives rise to a very unique idea of a "Central Registration Server" as explained that can act a biometric template matcher for a large number of authenticating server and by doing so a large number of hardware and efficiency costs can be handled. Also many groups together need just one very secure central server. Once such a single server is established then a large number of users can be configured. The system is still vulnerable to Denial of service attack at some points and is one of the drawbacks that need to be handled.

## 7 References

- [1] A.K Jain,A.Ross and S.Prabhakar, "an introduction to biometric recognition," IEEE trans. Circuit systems, Video Technol. ,Vol 14 ,no1 , pp 4-20, Jan 2004.
- [2] Lawrence O'Gorman,Avaya Labs, Basking Ridge "Comparing Passwords, Tokens, and Biometrics for User Authentication" Proceedings of the IEEE, Vol. 91, No. 12, Dec. 2003.

[3] N.K.Ratha , J.H Connell , and R.M.Bolle , “enhancing security and privacy in biometrics-based authentication systems,” IBM Syst. J.,vol.40 , no.3 , pp.614-634, Mar.2001.

[4]Upmanyu, M.; Namboodiri, A.M.; Srinathan, K.; Jawahar,C.V. “Blind Authentication :A Secure Crypto-Biometric Verification Protocol”;; Information Forensics and Security, IEEE Transactions on.Vol 2.Issue2, June 2010:pp 255

[5] Cryptography and network security , William Stallings 4<sup>th</sup> edition.

[6] R.Rivest, A.Shamir , and L.Adleman, “ A method for obtaining digital signatures and public key cryptosystems”, Commun ACM, Vol.21 , no 2 , pp.120-126,1978.

[7] MIT’s online documentation for Kerberos.

[8] B. Clifford Neuman and Theodore Ts'o. “Kerberos: An Authentication Service for Computer Networks”. IEEE CommunicationsMagazine,Volume 32, Number 9, pages 33-38, September 1994.

[9] USC/ISI online material for kerberos.

[10]S.P.Miller,C.Neuman,,J.I.Schiller, “Kerberos authentication System”, Project Athena Technical Plan:Section E.2.1.Cambridge:Cambridge University Press,1987.

[11] S.Yeqin, T.Zhongqun, Z.Xiaorong. “Security Analysis of Kerberos 5 Protocol” ,Computer Knowledge and Technology. Vol.6, no.6, pp.1319-1320, June 2010.

**SESSION**  
**POLICIES AND RELATED ISSUES + INTRUSION**  
**DETECTION**

**Chair(s)**

**TBA**



# Descriptive Analyses of Trusted Security Kernels and Autonomous Systems: Evolution of Security Approaches

**Michael Workman, Ph.D.**

Nathan M. Bisk College of Business  
Florida Institute of Technology  
Melbourne, Florida, USA

**Abstract** - Security countermeasures have been geared toward building fortified systems that focus on prevention and detection of attacks, and recovery from damage. However, according to the Defense Advanced Research Projects Agency (DARPA), given the increasing mobility of computing devices, security approaches must radically change to be effective. Some ways of working towards these ends are in creating systems that can reason and draw inferences and predictions about security vulnerabilities and threats. Among the most important advances are in systems that are adaptive and self-healing deriving from human biology and sociology. Thus security systems are aiming at resilience and adaptation –this is called socio-biologically inspired security. We contrast two approaches: Trusted kernels and sociological agentic systems to help security managers weigh decisions to fit the best approach based on needs and systems architecture.

**Keywords:** Trusted Security Kernels, Biologically Inspired Security, Adaptive-Agentic Systems.

## 1 Introduction

Today, systems are blending telephony with computing devices and thus are becoming more mobile. There are many technologies available and emerging, but the fabric used for these devices are increasingly configured as Peer-to-Peer (P2P) or mobile ad hoc networks (MANET). While multiple approaches have been suggested, there are two prominent ones: a *trusted kernel* versus *sociological agency*. We contrast these to enable security administrators and managers to distinguish which might be most appropriate for a given use, topology, architecture, or computing landscape. In the process we will introduce social exchange theory and agency in resilient security systems and what we define as *socio-biologically inspired adaptive systems*.

### 1.1 Human Body Metaphor

As indicated, adaptive security approaches often uses the human body as a metaphor in terms of building adaptive and resilient systems, particularly in light of the fact that many systems are mobile –such as laptops, notebooks, smart phones and other devices connected to what are called mobile *ad hoc* networks (MANET). These systems are expected to

sustain some damage, but they have the ability to recover (recuperate) from infections and develop synthetic immunization. The idea of detecting system damage was developed from danger theory [1], which explains the response of mammalian immune systems when danger such as a virus or bacteria, are detected. Using that analogy in computing systems, we can, for example, examine system executable files, and the linkages from one file to another in a calling sequence, along with the change events or state transitions, or code changes, and infer the cause of the cause [2]. The security literature presents these characteristics in the concept of biologically inspired security, especially as it fits into a cooperative communications infrastructure with mobile devices peer-to-peer (P2P) and MANETs.

### 1.2 Adaptive Self-Healing Systems

Adaptive systems have meant those systems that have self-healing capabilities. The term “self-healing” indicates the ability for a system to recover from damage. Just as there are different ways that systems adapt to their environments, there are also many mechanisms systems use for healing or repairing themselves. Many self-healing systems apply a model commonly called: Susceptible-Infected-Susceptible (SIS).

Each node or agent in this model represents a system or device, and each connection is a pathway through which an infection can spread. Individual nodes exist in one of a number of states such as “compromised”, “secure”, “infected” or “susceptible”. With each node that becomes compromised or infected the rate at which other systems become compromised or infected increases [3]. Until recently, self-healing has relied mainly on traditional security architecture –providing firewalls that can react to attacks and warn of risky behaviors, having built-in redundancies, standby systems, virus scanners, and recovery utilities. However, the term is evolving to represent more “self correction” methods including the use of genetic algorithms that can propagate information, optimize systems, and instruct changes in a way that resembles how human genetics interact (express) with their environments.

Self-healing then relies on the concept immunology. Immunization has been classified as passive, active, or a hybrid. Passive immunology is the approach previously mentioned, that is, to use firewalls and virus scanning

systems. Active immunization includes using intrusion detection mechanisms that automatically generate an appropriate response to a given signature or type of intrusion or attack. However, an interesting addition to this concept was described by [4] in which they defined an “automated method to detect worm attack, analyze the worm’s malicious code, and then generate an anti-worm” (p. 252). In this mechanism, the generated anti-worm would reproduce the behavior as the malicious worm and spread through the network until it could overtake the malicious worm and neutralize it.

Nevertheless, [4] acknowledged that the anti-worm could be reengineered to neutralize defensive systems. As implied then, hybrid immunology is a combination of active and passive approaches, which gives better defense in depth than either passive or active immunology alone. Research into self-healing approaches generally involve discovery of spreading damage. By tracing the damage along the paths through which the damage is occurring, the systems involved might (1) begin to predict the trajectory or the pathways most likely to be affected next, (2) trace back to an original source and decouple (block) the source from the infrastructure, and (3) reroute traffic while systems are repaired, for example by having virus scanners quarantine and repair the damage [5].

### 1.3 Danger versus Damage

Systems, especially as they become increasingly mobile and interconnected with other systems, are expected to operate in dangerous environments and sustain damage from time to time. The concept of allowing systems to operate in dangerous settings with the recognition that systems may suffer some damage goes against the conventional wisdom that tries to establish bastions.

However, building fortifications that depend on fixed sites and predictable configurations is becoming an untenable security approach (a fortress is not very mobile). The dynamic and trust-based nature of the network infrastructure, the heterogeneity of systems and applications, and the lack of centralized coordination of components –are just some of the issues that greatly complicate the use of conventional security mechanisms and techniques.

As a result, many security architects have embarked on new ways to implement artificial immune systems, which allow computing devices and nodes to participate in dangerous environments where they may receive some damage, and they will either continue to operate (perhaps at degraded performance) while self-healing, or suspend services and hand off work to a trusted peer until the system can recuperate. The analog of this is that of a person who becomes ill and his or her immunology produces antibodies to attack the invading “non-self” pathogens [6]. When ill in this way, a person develops symptoms, such as a cough and/or fever, but unless the illness is lethal, his or her performance may only be degraded to varying degrees and he or she can maintain some level of function.

It is not sufficient, however, for systems to simply respond to infections after the fact. Systems need to be able to detect if danger is present and try to avoid it; but if systems become contaminated, they must recognize when damage has occurred and type of damage that it has sustained in order to initiate the appropriate artificial immunological response [2]. The difficulty this presents is, given that mobile devices are a social collection in which one device may infect another at anytime, how can this be done? Some techniques have used self-contained environments called “trusted security kernels” (TSKs). TSKs have a damage detection engine (DDE), a cause analyzer, and an artificial immune response (AIR) activation [7]; other techniques use protected security configurations, configuration change controls, and anomaly detection schema [2].

## 2 Trusted Bio-Socio Security Kernels

A trusted security kernel (TSK) relies on a general assumption that as a network evolves, machines (i.e. nodes) accrue changes. These changes may or may not introduce security vulnerabilities. A key challenge with mobile devices in particular is that often they are isolated, and unable to query any central repository to determine if a change is benign or malicious. Furthermore, as malware can spread rapidly as with the “*SQL.Slammer*” worm, systems need to be able to respond to threats autonomously, and dynamically reconfigure to adapt [8]. To do this, the TSK monitors the behavior and changes to itself and to its peers, constructing *on the fly*, dynamic trust estimates for the different tasks. TSK can also authenticate the communications with peers. Assuming, for instance, an out-of-channel key exchange between TSKs during the pre-configuration phase, communications between kernels may be assumed to be secure and authenticated for the duration of its use.

### 2.1 TSK and Trust

Trusted security kernels or TSK generally take a pessimistic stance –maintaining a stable configuration and controlling requests for “privileged” operations by combating “pathogens” distinguished as “non-self” from those recognized by the TSK. It is important to recognize that security countermeasures at this level are no different than most any other end-to-end security countermeasure between applications. Yet potentially, these are vulnerable to attacks at the network level, both in regards to multi-hop transports and routing. The main problem with TSK consists of building a secure infrastructure that will create and maintain a trusted computation and communications environment from an end-node perspective, while still supporting dynamic changes in configuration, application and system settings. This relies on “updates” to the TSK, similar to how most people relate to updates to operating systems or virus scanners.

The main goal is to provide needed flexibility for systems, while at the same time, ensuring that potential vulnerabilities are properly identified (or inferred) by peer systems for reporting or behavioral adaptation. The DDE is

central in this role because it monitors mission-critical components in the system to identify degradation or security policy violations that should be classified as “damage” to the system. When an event is identified as damage, a trigger is issued to the *artificial immune response* (AIR) component that in turn performs a causal analysis using statistical correlation between previous events and current ones to identify the probable cause(s) of the reported damage.

## 2.2 TSK and Distrust

A biologically inspired TSK system has no need to find the exact cause of a fault, failure, or attack; therefore, the system uses a probabilistic technique to allow for uncertainty in its conclusions, and consequently is non-linear in creating adaptive immune responses. AIR is unique also in that it can receive input from a number of different devices. For example, if a system is experiencing a known attack, such as “ping of death” ICMP packets, the system doesn’t need to rely exclusively on damage-based input. As local nodes adapt to varying environmental conditions, the behavior of the AIR component is impacted by the collective system. By comparing local conditions with those of similar peers, the adaptive immune response can spread faster through the network than the damage that may be caused.

That said, aside from the differences in scale in relation to human biological systems, computer systems are far more fragile. Unlike in human biology, a single-bit error in a billion-bit program may cause the system to become completely inoperable. By comparison, biological systems generally require far stronger perturbation in order to fail catastrophically—their failure modes are far more flexible and forgiving than synthetic systems. Small changes in programs lead to dramatic changes in an outcome. Thus it is unreasonable to expect that one can naively copy biology and obtain a workable solution. Beyond these issues, the limitations of such a system are that, (1) it relies on an initial trusted configuration that participates in the communications, which may not be the case when new (and uncontrolled) devices join the network (see for instance the problems of security in peer-to-peer networks: [9]), (2) The TSK might have been compromised in any number of ways; and (3) the configuration is not dynamic. These issues leave incomplete the TSK security solution, although it remains a good concept for future security solutions.

## 2.3 Biological Adaptation

In biological terms, organisms can evolve in their social settings by reorienting their behaviors. For example, we learn to adjust our behavior based on the reactions of others. In most “trusted kernel” (TSK) configurations however, a static behavior (configuration) is both assumed and relied upon. Nevertheless, from a biological perspective, basic self-organization capabilities require that systems participate in forming functional security groups and then making socially acceptable adjustments from the feedback they receive; in

other words, we trust some friends more than others based on our experience with them. Ultimately, system security must be able to change according to environmental cues.

In this approach, security policies are often constructed using a graphical user interface and the underlying technology generates the ontology markup (e.g. DAML+OIL or OWL). It then performs policy *deconfliction*, which means to resolve conflicts in the rules that govern the decision-making. Conflicts may come about often in terms of access controls. For example, *Bob* is a member of *Group A*, which only has read access privileges to *Resource B*; but since *Bob* is also a manager, he has full read and write access privileges to that resource and thus must make an exception for him. Yet as suggested, there are limitations here as well. Something more is needed to take these nominal elements and determine whether there are malevolent deviations to form a normative measure.

## 3 TSK and Social Adaptation

The TSK approach has had to evolve and adapt to socially transmitted pathogens, just as in a human analog, where people today are more aware of how flu viruses are spread so they are more likely to wash their hands more frequently than a decade ago (witness the rise in hand-sanitizers). To address the problem of resilience using biological mimicry then needs to combine social approaches to learning about danger, adapting to it, and warning others. Modeling human social interactions in cooperative systems involves the concept of “agency” where systems may share information so as to collectively help trusted friends to avoid danger and relieve some of the workload while a damaged node recuperates (self-heals) from the damage or is taken out of service. Therefore, socially inspired security complements of biologically inspired security have evolved in that a node shares its symptoms of illness and methods for recovery with others it trusts.

### 3.1 Socially Influenced Biological Security

Bandura [10] described human social interaction in terms of *agency*, and he defined “agentic transactions” as the phenomenon where people are producers as well as products of their social systems. As such, agents act on three levels: (1) Direct personal agency in which an agent has goals, makes plans, and takes steps that are governed by certain rules, (2) proxy agency, in which one relies on others to act on his or her behalf to secure desired goals, and (3) collective agency, which is conducted through socially cooperative and interdependent efforts that affect other agents within a social network.

From a synthetic perspective, Sterne, et al. [7] suggested the use of a dynamic hierarchical model in which nodes and services are partitioned up through to an authoritative root. In this configuration, the system relies on clustering to enable scalability, resilience (fault-tolerance), and adaptability to the

dynamic environments. Each node in the communicative peer group maintains responsibility for its own security (e.g. from an Intrusion Detection System perspective), as well as stating some limited responsibilities for its adjacent peers, which is summarized and propagated up the authoritative chain.

Directives, on the other hand, are passed top-downward. In a biological analog, designers of inorganic systems have relied on lessons from human immunology to develop genetic algorithms and self-healing systems [6] for some time. Still, using a sociological analog, digital sociologists have relied on principles shown in social networking that have led to developments such as viral marketing and incentive-based cooperative systems [11]. Taken together, these have inspired modeling biological systems of security transactions in terms of networked systems –especially in highly interdependent and cooperative systems such as cloud computing, grid computing, peer-to-peer networks and MANETs. For this we must introduce the concept of agents and agency.

### 3.2 Agents and Agency

Biologically inspired security systems, agents are analogs of a human organism. Importantly, there are micro and macro –levels of agency. At the micro-level, we have discussed how a human body produces white blood cells to attack “non-self” pathogens, but there is something more. Pathogens are spread from human-to-human, and in the human analog, there are symptoms that may cause a person to take precautions, such as washing one’s hands after shaking the hand of another who is sniffing and sneezing. To the point, micro-level activity in an agent is not sufficient to protect a person. Moreover, we refer to precautions such as hand washing as agentic-behavior.

Thus, people as well as biologically-inspired systems behave socially to exchange information, receive instructions, react to the effects of other agent actions, and provide responses in a cooperative fashion to fulfill individual and collective goals in an adaptable and evolutionary way, while simultaneously healing from and warning others of security violations and violators [9]. For agents to take individual and social action, they must rely on the information they have to make predictions about the consequences of their behavior. In other words, to be proactive, agents require an initial set of knowledge from which agents base their assumptions [12]. As an illustration, security policies may be learned and generated by running an application in a controlled environment to discover its “normal” behavior profile.

For instance, much of the research that has been done in this regard might be attributed to early work during the 1980s/90s at INRIA (Sophia Antipolis, France). The concept of profiling “normal” behavior depends on having a trusted initial run of an application or system, and determining the range of behaviors it performs including requests for privileged (system) calls. When run subsequently, the security system monitors the application to determine whether it deviates from this predefined behavior. If so, the application execution is intercepted by the TSK; for example, if it

attempts to make systems calls that are prohibited [13]. In other words, TSK critically relies on detecting anomalies based on “signatures” that must be updated by the TSK.

For more flexibility, a contemporary method is to import into the security policy ontology (such as codified as OWL) the threat and vulnerability information accumulated by a global security entity. For example, as provided by the Common Vulnerabilities and Exposures (CVE) ontology [14], which captures and updates with common vulnerabilities and incidents reported by the Software Engineering Institute’s CERT. Based on these profiles and configuration information, the agent itself may detect damage and then communicate it to an authoritative other (a trusted proxy), or it may learn of damage or danger by proxy, such as confederated intrusion detection systems [7] that detect changes or monitor events.

While proxy agency such as confederated security offers some clear benefits particularly in terms of administration, in highly dynamic and mobile network topologies, this is not always possible. Something more may be needed, and to the extent that an agent can understand and manage itself in its environment, the more autonomously it can act (i.e. it relies less on a hierarchy of structures or administrative domains), and when critical information is unknown, it is learned, and this learning is most likely to be derived socially; and more specifically, it is learned through collective agency.

### 3.3 Socially Interactive Systems

While biologically-inspired security can be extremely effective in many if not most system and network topologies and configurations, nodes in a highly interdependent and dynamic configuration creates unique challenges that a conventional approach does not address very well in isolation, at least in a static way. For example, in a network that hosts mobile devices, the exchange of information occurs in a flexible topology where devices may join and leave the network in ways that are not predictable.

Paths between any set of communicating devices or nodes may traverse multiple wireless links, which may be comprised of heterogeneous platforms running various applications and consisting of strict limitations, such as in RAM or in network transmission rates. They also vary in their ability to supply underlying security countermeasures such as firewalls, virus scanners, and intrusion detection systems. Eventually, these underlying countermeasures cannot be completely relied upon. Neither is it practical to expect that all nodes will behave in a predictable fashion; or that a centralized or confederated model of security will be able to oversee all of the dynamic activity and responded in an effective and timely manner.

Bandura [10] described human social interaction in terms of agency, and he defined “agentic transactions” as the phenomenon where people are producers as well as products of their social systems. As such, agents act on three levels: (1)

Direct personal agency in which an agent has goals, makes plans, and takes steps that are governed by certain rules, (2) proxy agency, in which one relies on others to act on his or her behalf to secure desired goals, and (3) collective agency, which is conducted through socially cooperative and interdependent efforts that affect other agents within a social network. From a synthetic perspective, Sterne, et al. [7] suggested the use of a dynamic hierarchical model in which nodes and services are partitioned up through to an authoritative root. In this configuration, the system relies on clustering to enable scalability, resilience (fault-tolerance), and adaptability to the dynamic environments.

Each node in the communicative peer group maintains responsibility for its own security (e.g. from an IDS perspective), as well as some limited responsibility for its adjacent peers, which is summarized and propagated up the authoritative chain. Directives, on the other hand, are passed top-downward. From a biological analog, designers of inorganic systems have relied on lessons from human immunology to develop genetic algorithms and self-healing systems [6].

## 4 Autonomous Systems

Many efforts have been made to make biologically-inspired systems more socially interactive. However, the bottom line is that systems (like human beings) do not function in isolation. We interact with our environment. Just because one might become infected with a pathogen does not mean one may not become aware of social contamination that may even lead to a pandemic. While TSK and its adaptors for social sensors have improved how systems better insulate and recover from a pathogen, they lack the broader means that humans in social societies have of warning others about this danger. In this, the final section, we present how socio-agentic systems strive to more accurately conform to Bandura's [10] triadic reciprocity model. It is important that we note that not all systems require such sophistication, many systems may easily be defended (based on cost-benefit) by conventional means. Nevertheless, systems of the future must defy convention to remain relatively secure.

### 4.1 Socially Promiscuous Systems

In a sociological analog, digital sociologists have relied on principles from social networking that have led to developments such as viral marketing and incentive-based cooperative systems [11]. Thus taken together, these have inspired what is currently called *agent security* transactions in networked systems—especially in highly interdependent and cooperative systems such as cloud computing, grid computing, peer-to-peer networks and mobile ad hoc networks (MANET).

In these configurations, agents behave socially to exchange information, receive instructions, react to the effects of other agent actions, and provide responses in a cooperative

fashion to fulfill individual and collective goals in an adaptable and evolutionary way, while simultaneously healing from and warning others of security violations and violators [9]. For agents to take individual and social action, they must rely on the information they have to make predictions about the consequences of their behavior. In other words, to be proactive, agents require an initial set of knowledge from which agents base their assumptions [12].

Security policies may be learned and generated by running an application in a controlled environment to discover its “normal” behavior (a profile). When run subsequently, the security system monitors the application to determine whether it deviates from this predefined behavior. If so, the application execution is intercepted; for example, if it attempts to make systems calls that are prohibited [13]. Another method is to import into the security policy ontology threat and vulnerability information, such as provided by the Common Vulnerabilities and Exposures (CVE) ontology [14]. The CVE captures and updates with common vulnerabilities and incidents reported by the Software Engineering Institute's CERT. Based on these profiles and configuration information, the agent itself may detect damage and then communicate it to an authoritative other (a trusted proxy), or it may learn of damage or danger by proxy, such as confederated intrusion detection systems [7] that detect changes or monitor events.

While proxy agency such as confederated security offers some clear benefits particularly in terms of administration, in highly dynamic and mobile network topologies, this is not always possible. Something more is needed, and to the extent that an agent can understand and manage itself in its environment, the more autonomously it can act (i.e. it relies less on a hierarchy of structures or administrative domains), and when critical information is unknown, it is learned, and this learning is most likely to be derived socially; and more specifically, it is learned through collective agency.

### 4.2 Socio-Biological Security

While biologically inspired security can be extremely effective in many if not most system and network topologies and configurations, nodes in a highly interdependent and dynamic configuration creates unique challenges that the conventional approach doesn't address very well in isolation, at least in a static way. For example, in a network that hosts mobile devices, the exchange of information occurs in a flexible topology where devices may join and leave the network in ways that are not predictable. Paths between any set of communicating devices or nodes may traverse multiple wireless links, which may be comprised of heterogeneous platforms running various applications and consisting of strict limitations, such as in RAM or in network transmission rates.

They also vary in their ability to supply underlying security countermeasures such as firewalls, virus scanners, and intrusion detection systems. Eventually, these underlying countermeasures cannot be completely relied upon. Neither is

it practical to expect that all nodes will behave in a predictable fashion; or that a centralized or confederated model of security will be able to oversee all of the dynamic activity and responded in an effective and timely manner. As with human social interaction [15], [16], systems may use the concept of collective agency in which a system assumes certain responsibilities that others rely upon, such as delivering a service at a certain quality of service (QoS) metric. If those responsibilities are not fulfilled, then the agent becomes distrusted by others [11].

It is important to note that a violation that prohibits a system from living up to a promised QoS metrics such as exceeding a latency threshold may not be maliciously caused; it may be a legitimate operation that causes a temporary condition. Nevertheless, a repeated offense would be treated the same as a malicious attack. Also, in human social systems people tend to trust strangers less than they trust their friends [10]. In making a decision about interacting with a stranger, people often inquire of their friends about the stranger's reputation. In security, this feature is known as a reputation-based system [17]. If friends don't know the stranger, the stranger will not have any reputation.

In a highly interdependent network of systems, distrusting systems that have newly joined the network, that is, strangers, can negatively impact the collective performance and availability of information and computing resources because the more systems that cooperate and share the load, the better the performance. Since penalizing a stranger and giving preference to friends discourages participation in a social exchange, an adaptive stranger policy [11], [17] deals with this by requiring each existing peer to compute a ratio of the amount of resources requested by the stranger. If it is less than an established threshold, then the peers will work with the stranger, so long as the stranger does not try to violate a security policy or attempt a known security threat. If the stranger's request exceeds the threshold, then peers will compute a probability of working with the peer.

This probability is used in determining whether the request can be serviced by shared cooperation, throttled back—such as reduced transmission rate, increasing latency, or lowering bandwidth, or deferring the request to later as a low priority [18]. As devices interact and gain experience with the stranger system, and if the stranger maintains a good reputation over time, the stranger will become a trusted friend—that is unless the stranger inflicts some damage, in which case, the reputation of the stranger will be negatively impacted. Depending on the kind of damage the device inflicts, it may even be labeled an enemy and locked of the communications. In that case, where a device is determined to be an enemy (an attacker), then the enemy may try to rejoin the network as a new stranger—a technique called “white washing.” The adaptive stranger policy mitigates this by carefully watching the stranger and sharing reputation experience with trusted “friends.” Sometimes trusted friends may try to collude with the attacker by giving the enemy a

good reputation, raising the trust of the stranger among the collective nodes. To combat this, each node must carefully watch the resource consumption thresholds, QoS metrics, and only gradually increase trust as a function of the collective experiences of the nodes in the network [2].

### 4.3 Collective Agency and Adaptive Systems

In the movie *Star Trek*, in *The Wrath of Kahn* episode, Mr. Spock said: “Don't grieve, Admiral. It is logical. The needs of the many outweigh the needs of the few... Or the one.” This is a fitting description of the goal of collective agency. As the designers of the Arpanet envisioned, a survivable network requires the collective effort and redundancy to fulfill an overall objective. In most systems that utilize QoS metrics and local versus remote prioritization—such as scheduling algorithms and routing protocols, a condition known as “selfish link” can occur [19].

A “selfish link” reflects agent actions that lead to a lack of cooperation and undesirable network effects. Among these issues is “free riding” [19], [2] where agents disregard their obligations to other agents in favor of self-preservation, for example, to preserve its own compute cycles or communications bandwidth for its own services, such as running local services at the highest priorities and lowering priorities of requests from external nodes. To address these problems, incentive models may be used to encourage more altruistic behaviors; that is, to share resources to better ensure service and resource availability.

To approach the integrity issue, there are at least three behavioral treatments for collective agency: (1) Providing incentives and positive reinforcements for cooperative actions, such as such as allocating queue preferences to efficient nodes, (2) negative reinforcements, such as sending requests to cause an agent to follow through on an obligation, or to cease malevolent or unintentional damaging behavior, and (3) punishments levied against agents that ignore negative reinforcements or that are violators of a security policy or issue a request that is a known threat. All three approaches are needed to create a resilient environment and preserve the availability and integrity of resources and services [2].

Adaptive systems security must be able to combine biologically inspired and socially inspired approaches to security, especially given three trends: (1) computing devices are becoming more compact and mobile, (2) computing devices are increasingly part of sharing resources in a cooperative ad hoc network—such as with P2P and MANET systems, and (3) computing is becoming more virtual, such as distributed through cloud or grid computing infrastructure, which in many cases may be managed and operated by a third party such as with Microsoft's Azure or IBM's set of cloud facilities.

Beyond the abilities that system components can monitor themselves and converse with communicative partners, some of which may be considered “close friends”, and others acquaintances, and still others strangers, they adjust their behavior accordingly in order to adapt to their environments. At one time an agent may be resident on a familiar “in house” system, and at another time, it may be distributed to a foreign node in an unfamiliar environment. Security administrators and programmers cannot possibly reconfigure these systems appropriately –the systems must learn their hosted environment and react correctly.

With each staging event such as relocation from one virtual machine to another, or from one environment to another, the system (or node) must orient itself to its base configuration; and then it then must establish its goals and set its priorities, and begin collecting information about its neighbors. While making such adjustments, peers may offer incentives for cooperating in a new environment, just as a new employee in the workforce might respond better to a colleague who offers advice rather than one who ignores him or her. Gaining cooperative behavior such as to entice selfish agents to participate is called an *incentive based security system*, such as giving one node an incentive for routing and forwarding data by giving it preference over others when it makes a request [20]. This method, however, does not discourage malevolent behaviors such as nodes that continue transmissions even after ICMP source quench requests have been made [21]. Still, from this example, danger might be inferred if an agent that issues a notification that is subsequently ignored.

#### 4.4 Novelty as Potential Danger

Given this social interactivity in determining danger and realizing damage to others, the inclusion of a reputation-based system mitigates potential hazards by maintaining and collecting votes from other agents about their favorable or unfavorable history with the requestor. If the requestor has an unfavorable reputation as determined by the agent’s local history, an agent may resort to punishing the malevolent requestor by simply adding it to its service prohibitions. On the other hand, if the other agents report an unfavorable reputation, but an agent’s local history is favorable, it may choose to allow it, but monitor the behavior of the request such as its consumption of the agent resources (CPU compute cycles, bandwidth, memory) or attempt to make changes to the configuration such as copying, modifying, or deleting a file. If an agent with a good reputation begins to cause damage to the system, such as through requests of a provider that absorb most or all available resources, the agent may issue negative reinforcement demands that the requester reroute or throttle its requests, and then monitor for compliance and update the reputation history and report to other agents accordingly.

If the requestor ignores the demands, the agent may switch to punishments such as queuing to low priority for

instance, if the request is not a known threat but is exceeding a threshold for some resource such as bandwidth. If a request is determined to be a threat, then the system would automatically block the attacker agent or drop its packets. From a security policy perspective, an agent is granted rights according to a given role, but in an *ad hoc* communicative environment, rights and roles can be very dynamic.

Behavioral role conformance to a well-defined set of behaviors based on allocated rights is the usual case and considered benign. Benign behaviors do not need to consume system resources that need to be closely monitored, since monitoring consumes its own resources. However, when something novel is encountered, it presents potential danger. Novelty may be such that an agent attempts to perform an unauthorized function, or an agent performs an authorized function, but the function attempts to perform some novel behavior. Danger therefore can equate to novelty in terms of security policy enforcement.

From this frame of reference, danger can also be viewed as a continuum on a severe (X coordinate) and imminent (Y coordinate) axis. When danger is encountered, it is monitored according to its coordinates on these axes. A threshold can be set in the continuum in which an intervention may be required to preclude damage from occurring. Damage in this context may be defined as any action that would impinge on mission execution, including negative effects on mission parameters such as exponential consumption of network bandwidth, an application that normally does not copy files tries to copy a file, or negative impact on any QoS parameters needed for successful mission execution.

The structures of the actions could consist of goals, plans, steps, and rules, which are malleable and negotiable. That is, an agent assembles its own set of rules dynamically while reacting to social and environmental events, and selecting appropriate qualifying plans to achieve its current goal. The agent may try other qualifying plans if initial plan fails, or if events or exceptions cause the agent to change plans. Consequently, agents are adaptive to a given situation in that they select the next plan based on current information that is available, either from the environment or from its local knowledge [12].

The goals at the root of the agent hierarchy consist of the agent’s obligations and prohibitions. As a case in point, a goal might be to obligate the agent to provide a Web service. Goals consist of plans to execute in order satisfy the goal; for example, an obligation to provide a Web service might require a plan to start httpd on port 8080. Agent behaviors are governed by rules, which might specify that an ActiveX control is not permitted into the Web service.

Rules operate on the steps that are part of a plan; thus, if an agent receives an http request containing an ActiveX control, the rule may require steps to discard the request. The choreography of agent actions may be decomposed into three

levels: high-level, intermediate, and low-level. Perhaps an agent has multiple goals, and each goal has multiple plans. For example, the goal “maintain current version levels of applications,” may have a high-level plan entitled: “Version Updates.” For this high-level plan, there are intermediate plans that perform a sequence of steps, tasks, and log data updates, such as, “Automatic Update AcroRd32.exe” may require a network connection to be opened and a file download from the Adobe website over a TCP/IP socket.

Low-level plans perform the system tasks and log data updates, for example, open 127.0.0.1:2076, and record the conversation with Adobe. In this way, beginning with an initial set of plans, execution may proceed through paths from one plan to another by the agent, and this allows it to limit the scope of its response to address more localized problems (e.g. provide an http connection for agent X, but not for agent Y). Also, if a goal can be achieved in different ways (for example, manual or automatic update), the three levels of plans allow for this localized ability.

Damages that may occur vary according to the types of activities that an agent attempts. In data sharing for example, agents need to utilize at least two different forms of resources: storage in which each agent has to set aside some storage space for files that may be needed by other agents even though these files may not be useful to the agent itself, and bandwidth, where each agent must devote some of its bandwidth for messaging and communications requested files by other agents. Damage in this specific sense is assessed according to the excess of security policy-defined thresholds.

To illustrate this case scenario, security agents interrogate their configurations and vulnerabilities ontologies against requests for services according to available resources, access and rights, reputations, and “suitable behaviors.” According to [13], running initially through applications and generating profiles would create baseline configurations, but this is not always possible. It is more likely in many configurations that agents learn by adding benign behaviors into the agent’s ontology. For example, a request may be made for Web service on port 8080. When the request is serviced, the agent checks the ontology for the Web service to determine what actions are permissible. As long as the requestor behaves properly, no danger is detected, and permissions are granted for using resources according to defined QoS (e.g. memory utilization, network utilization, CPU utilization) and configuration (e.g. file checksums, etc.) and plan parameters. In order for the agent to initiate a self-healing process after servicing a request that causes damage, for instance, if an agent detects violations to QoS mission parameters or a violation of normal behavior, it calculates a vector for the violation and determines damage severity.

If severe damage is determined after a servicing a request, the damage is flagged as malignant and filtered through a damage controller to determine what part of the agent is malignant and what actions to take, along with

gathering data about the agent whose request caused the damage and adding it to the prohibitions and give the malevolent agent a low reputation rating. On the other hand, if a stranger makes a request, or if a requestor with a high reputation makes a novel request, then danger is determined and the request is proactively monitored. That is, in cases of agent or behavior novelty, the request is quarantined and monitored to determine what resources it may utilize and to what extent, according to the obligation QoS parameters, or available resources in lieu of obligations. If a threshold is likely to be exceeded or a resource violation is likely to occur, the agent may notify the requestor, for example, to cease or reroute the request. If the requestor complies, a high reputation is given. On the other hand, if the request represents a known threat, the request is flagged as malignant non-self, and is filtered through a damage controller to determine what actions to take, such as to deny the request, and issue a low reputation for the requestor.

## 5 Conclusions

Based on the evolution where systems are melding telephony with computing more practical solutions are needed compared to the conventional forms of security and fortifications. A way this can be facilitated is by direct agency, enabling the flexibility for agent goal-directed autonomous behavior. However, given the nature and limits of the topology, proxy cooperation with other agents is essential [9]. Each agent carries out its own set of goals according to its plans, and makes requests of other agents, which are fulfilled so long as there remains cooperation and “good” behavior.

A plan as we indicated consists of the sequence of steps associated with the goal, and rules that govern its responses to requests, events, and exceptions. An agent may have multiple plans available for a given goal, and multiple goals to accomplish, which are dynamic and negotiable [12]. Collective agency develops in response to agent reactions to events and other agent requests, and agents provide services based on its available resources and the agreements the agent forms socially in the mobile network such as a MANET. However, the combined sociability and the ad hoc nature of the MANET create an insecure environment. It must be possible for an agent to have the ability to detect severe impending danger so that some preventative measures can be taken before damage occurs. Malevolent behavior is collected, and reported to other agents by reputation, but some conservatism is built into the immunology by treating strangers cautiously while at the same time, not denying them the opportunity to prove themselves as trustworthy.

On some level, danger needs to be monitored, but not necessarily disrupted, unless it presents a severe and imminent threat potential for lethal damage. However, dangerous behaviors by undetected illegitimate agents (or colluders) may involve eavesdropping or interception of communications and may not escalate to disrupting the

mission parameters and might not be detected, and thus rely on the typical cryptographic approaches to this problem –which may not be supported by all MANET agents.

Danger provokes monitoring –and because in an ad hoc environment, novel behaviors must be tolerated until it is learned whether the actions are benign or malignant but does not necessarily preclude them –depending on the assessment of the severity and imminent threat assessment. Damage on the other hand, cannot be tolerated (at least for long). Being able to rapidly assess the QoS and timely actions by an agent is critical problem to solve [22]. These remain critical research and development challenges to solve in widely-disbursed commercial applications.

## 6 References

- [1] Iqbal, A., & Maarof, M. A. (2005). Danger theory and intelligent data processing. *World Academy of Science, Engineering and Technology*, 3, 110-113.
- [2] Workman, M., Ford, R., & Allen, W. (2008). A structuration agency approach to security policy enforcement in mobile ad hoc networks. *Information Security Journal*, 17, 267-277.
- [3] Wang, A. H. & Yan, S. (1995). A stochastic model of damage propagation in database systems. *Conference on Security and Management, SAM'09 Las Vegas, NV, 1*, 3-9.
- [4] Toutonji, O., & Yoo, S. M. (1995). Realistic approach against worm attack on computer networks by emulating human immune system. *Conference on Security and Management, SAM'09 Las Vegas, NV, 1*, 251-257.
- [5] Ford, R. (2008). *BITSI*. Unpublished technical whitepaper, Melbourne, FL: Florida Institute of Technology.
- [6] Aickelin, U., & Cayzer, S. (2002). The danger theory and its application to artificial immune systems. *University of Kent at Canterbury*, 141-148.
- [7] Sterne, D., et al., (2005). A general cooperative intrusion detection architecture for MANETs. *Proceedings of the Third IEEE International Workshop on Information Assurance (IWIA), Washington DC*, 57 - 70.
- [8] Ford, R., Carvalho (2007). *Biologically Inspired Security*. Unpublished technical whitepaper, Melbourne, FL: Florida Institute of Technology.
- [9] Boella, G., et al., (2005). Admissible agreements among goal-directed agents. *Proceedings of the IEEE/WIC/ACM Conference on Intelligent Agent Technology (IAT'05)*, Paris.
- [10] Bandura, A. (2001). Social cognitive theory: An agentic perspective. *Annual Review of Psychology*, 52, 1-26.
- [11] Feldman, M., & Chuang, J. (2005). Overcoming free-riding behavior in peer-to-peer systems. *ACM SIGcomm Exchanges*, 5, 41-50.
- [12] Dastani, M., van Riemsdijk, M. B., Dignum, F., & Meyer, J. J. (2004). *A programming language for cognitive agents goal directed 3APL*. Lecture Notes in Computer Science (pp. 1611-3349). Heidelberg: Springer.
- [13] Provos, N. (2002). Improving host security with system call policies. *Proceedings of the 11th USENIX Security Symposium*, 2, 207–225.
- [14] Moreira, E. & Martimiano, L, Brandao, A., & Bernardes, M. (2008). Ontologies for information security management and governance. *Information Management & Computer Security*, 16, 150-165.
- [15], [16], [17] Chomsky, N. (1979). Human language and other semiotic systems. *Semiotica*, 25, 31-44 - parts 1, 2, 3.
- [18] Sun, O., & Garcia-Molian, H. (2004). *SLIC : A Selfish link-based incentive mechanism for unstructured peer to peer networks*. Proceedings of the 34<sup>th</sup> International Conference on Distributed Computing Systems. Los Alamos, CA: IEEE Computer Society.
- [19] See [11], Feldman & Chuang, presentation of manuscript.
- [20 and [21] Buchegger, S & Le Boudec, J. Y. (2005) Self policing mobile ad hoc networks by reputation systems. *IEEE Communications Magazine, July*, 101-107, and presentation.
- [22] Antoniadis, P., Courcoubetis, C., & Mason, R. (2004). Comparing economic incentives in peer-to-peer networks. *The International Journal of Computer and Telecommunications Networking*, 46, 133 - 146

# A Robust Trust Model for Named-Data Networks

Vahab Pournaghshband and Karthikeyan Natarajan  
Computer Science Department  
University of California, Los Angeles

**Abstract**—Any future Internet architecture must offer improved protection and resilience over today's network, which is subject to pervasive and persistent attacks. A recently emerging architecture, Named-Data Network (NDN), treats content as the primitive entity. This leads to decoupling location from identity, security and access, and retrieving content by name. NDN security is based on the establishment of a trustworthy routing mesh, relying on signed routing messages and an appropriate trust model. Signature verification of NDN content merely indicates that it was signed with a particular key. Making this information useful to applications requires managing trust, allowing content consumers to determine acceptable signature keys in a given context.

In this paper, we propose a robust trust model for NDN to securely learn public keys of content publishers so that applications can determine what keys are trustworthy. In doing so, the user asks for publisher key recommendations from all entities in its community of trust, which consist of people the user personally knows, as in real world interactions. A local policy is then used to decide consistency of responses, and hence trustworthiness of the publisher's key. Also, we present a suitable key revocation approach for this model. We then provide a discussion on robustness of this model against various attacks.

**Keywords:** decentralized trust model, named-data network, web of trust

## 1. Introduction

In the current Internet, packets are routed based on IP addresses irrespective of the content the user is looking for. NDN advocates for a routing infrastructure where forwarding decisions are made on the content requested. NDN also caches data at the routers along the path through which the response (data) travels. An interest packet travels until it encounters a valid source of data, which could either be a router's cache, or the publisher of the data.

From a security standpoint, there are two concerns that are to be addressed in such architecture:

1. Did the user receive the message as sent by the publisher?
2. Is the publisher really the person whom he claims to be?

Here we are trying to establish the integrity of the data and the authenticity of the data source. A user depends on

the trust model provided by the Internet infrastructure to determine these two. We will present a trust model that is decentralized, providing the user with necessary information to make a decision to establish trust on an entity.

### 1.1 NDN Basics

In NDN, when an interest is sent out for a particular name, we receive the data back from the publisher or from a node that had cached the data. The response has the following components, which are of specific interest to the discussion here: Signed info and Signature (Figure 1).

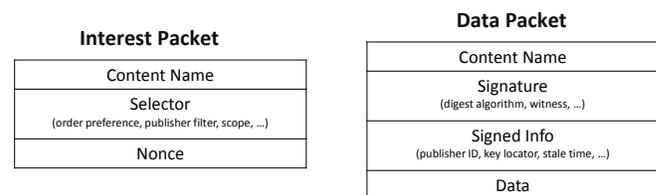


Figure 1: NDN Packet Types

The content publisher signs the hash of the content and the name of the content. Hence, given the publisher's key, the integrity of content received is verified by comparing the given signature.

A more challenging problem, however, is to verify the authenticity of the publisher's public key. Every trust model would provide means of retrieving a published public key. Here, the "Signed info" component of the data consists of the keylocator and the publisher's public-key name, to facilitate key verification problem.

By querying the publisher using the information provided in the signed info part of the message one can fetch the publisher's public key. However, this is susceptible to man-in-the-middle attacks sitting between the user and the publisher, supplying fake keys to the user. Hence, there is a need for secure retrieval of public keys. The conventional PKI scheme trusts a third party to provide the user with the correct public key of the publisher. In most of the cases, the end user has no idea of who this trusted entity is. We address this problem by distributing trust among a set of people whom the user would trust.

### 1.2 Review of Current Approaches

This section provides a brief overview of the current approaches of establishing trust in the Internet. We will briefly

cover leap-of-faith, PGP, Certificate Authorities, DNSSEC, and SPKI/SDSI. We will also discuss the shortcomings of each of these approaches.

### 1.2.1 Leap-of-faith

This is the approach widely adopted by the secure shell application. When a client connect to an SSH server, the server sends its certificate to the client. The client, in order to communicate securely afterward, accepts this certificate to be the trusted key of the server. This certificate is cached for future access to the server. The application trust the certificate obtained, assuming that no Man-in-the-Middle attack was performed during the initial key request. Such a trust model, while simple, is clearly vulnerable to attacks especially in the case of key changes and hence is not being used for sensitive applications like online banking or e-commerce.

### 1.2.2 PGP

Pretty Good Privacy (PGP) is a computer program that provides cryptographic privacy and authentication. PGP is often used for signing, encrypting and decrypting e-mails to increase the security of e-mail communications. It uses the concept of a web of trust. To trust a certificate (i.e., the public key of an entity), the user requires someone he trusts to endorse the untrusted entity. The trusted person who endorses the certificate is likely to have used out-of-band means to get the certificate in question.

### 1.2.3 Certifying Authority

Certifying Authority (CA) is a trusted third party that issues certificates for publishers' keys. The certificate contains the public key and the identity of the publisher. The public key of the CA is publicly known and is normally hard-coded within all the nodes connected to the Internet. The CA computes the hash of the key as well as the publisher identity and encrypts it with its private key, which serves as the certificate of the publisher. When a client wants to communicate to a server, it can verify the authenticity of the server by comparing the hash of (key,name) pair obtained from the publisher to the hash signed by the CA. This is essentially trusting whatever the CA has signed. A typical certificate issued by a CA for some publisher would contain the publisher's public key, publisher's name, the key's expiration date, and the signature of this information signed by the CA's private key. The expiration date is used to identify the certificate's lifespan.

### 1.2.4 DNSSEC

This model is used to secure the DNS service running on top of the IP Network. It provides authenticated response to the lookup queries. All DNS answers are signed by the authority server that is responsible for a particular domain.

Consider A.B.com : A's key is signed by B. B's key is signed by com and com's key is signed by the root. The root's key is hard-coded in all browsers. They follow this chain of keys to establish trust in a DNS response. All domains are signed by their respective parent domains. The key of the root is installed in your system. Key revocation and rollover has been also a problem for this model.

### 1.2.5 SPKI/SDSI

Simple Public Key Infrastructure (SPKI) is designed to be an alternative to the X.509 standard for digital certificates. SPKI views authority as being associated with principals, which are identified by public keys. It allows binding authorizations to those public keys and delegation of authorization from one key to another.

## 2. Design

### 2.1 Principles and Assumptions

There are numerous characteristics of a robust trust model that any design at the Internet level should have such as scalability, easy enrollment, decentralized authority, and resiliency against attacks such as Denial-of-Service (DoS) and Man-in-the-Middle (MitM) attacks. In designing our trust model for NDN, besides taking these characteristics into consideration, we primarily concentrated on preserving the liberty of choosing who to trust by avoiding complete trust on third parties which are trusted not at the user's will. In addition, flexibility in local policies is also desirable, in the sense that each person defines security individually which would give them locality of control. This is important since not all users care at the same level and not all contents need the same level of attention. And lastly, which is provided by the nature of NDN, the notion of trust should be contextual, i.e. narrowly determined in the context of particular content and the purpose for which it will be used. In our design, we assume that attacks are either localized to a particular network scope or of limited duration, since a larger attack is more easily detected and further remedied. This is true mainly since most network level attacks on integrity and secrecy of information need to remain undetected to be successful.

### 2.2 Overview

We define the notion of community of trust to be a combination of our friends (whom we know) and a selective subset of publicly available network notaries. Notaries (or pseudo-friends), as defined in Wendlandt et al. [4], are servers which are solely designed to monitor and record history of public keys used by a network service. A notary frequently asks for the keys from particular servers and updates its database upon a key change. As its service, each notary responds to queries from clients who ask notaries about content publishers' public keys.

Upon request for a particular data, the user receive the data signed by its claimed publisher. If the user has the publisher key associated to the data cached, then the authenticity of data can be verified immediately using the cached key. If the cached publisher's key is suspected as invalid or not cached, the client issues a public key request for the publisher to all its friends and notaries in its community of trust. Upon receiving responses from them, as well as an offered key from the publisher itself, the user applies its local policy to judge the consistency of, possibly weighted, responses to make its trust decision on whether to accept the key or not. In the following sections we discuss this approach in details.

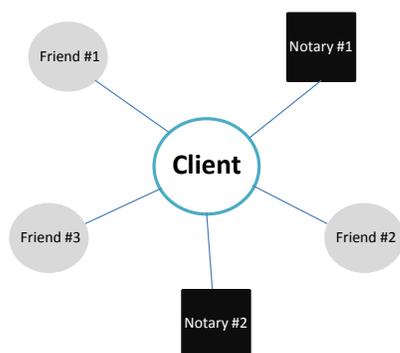


Figure 2: Notion of Community of Trust

### 2.3 Trust Bootstrapping

A provably secure, and yet its practicality being still in question, approach to bootstrap trust is the use of out-of-band mechanisms to learn the public keys of the user's friends in his community of trust. This could be less of the problem since the user is expected to have personal relationship with his friends and hence utilizing a large variety of such out-of-band means. To obtain, the notaries' public key, however, the user can request it from his friends, the same way to get any other publisher's key. Some other mechanisms such as security through publicity [5] can be applied for some well-known notaries.

### 2.4 Notion of Master Key

Every publisher maintains a master key which is used to sign only other keys under the publisher's domain. In fact master key is the only offered publisher key that invoke key recommendations from friends. Once that key is validated and cached, the publisher would play as the Certifying Authority (CA) for all its sub-domains by generating certificates and revocation notifications.

The master key needs to be highly secure and requires a longer life span. That is why it is only used to sign top level keys in the publisher's domain and not content, mainly to protect it against known-plaintext attacks by having very few samples publicly available. Besides, due to higher measures

of security considered in choosing master keys, the signature verification process is expected to be relatively expensive, hence not suitable for signing contents.

Note that even though the notion of master key shares similarities with the current CA mechanism, they are fundamentally different. One reason is that there is no third party involvement in the process, since the publisher manages certificates for its own domain. Also, it incurs no cost to generate or revoke certificates, unlike current approaches that involves third party companies that charge for such services.

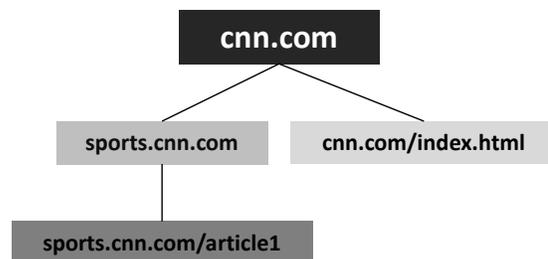


Figure 3: Master Key Signing its Sub-domains

### 2.5 Key Revocation

An important part of every public-key trust model is key revocation. The public-key will be well known to a large but unknown set of users. The revocation system must have the ability to replace the private-public key pair, distribute the new key, and notification to the users of the revocation should happen in a timely manner. Key revocations have two fundamental approaches, implicit and explicit. An implicit revocation is asserted by the ability to retrieve the certificate. Any certificate retrievable is guaranteed to be valid near the time of retrieval. Also common is expiration or Time-To-Live (TTL), which is the maximum time the certificate is valid before requiring a new certificate from the issuer. This is more commonly known as certification expiration. However, from the time the certificate is retrieved to the expiration date is a window of vulnerability if a key has been explicitly revoked. Explicit revocation is when the issuer states that certificates are no longer valid, or have been compromised (also indirectly which certificates have not been revoked). Current explicit revocation approaches come in two forms, online and offline. Some popular offline models include Certificate Revocation Lists (CRL) [13],  $\Delta$ -CRL [13], and Certificate Revocation Trees [14]. These models explicitly state that particular certificates have been revoked, and they can be obtained through the issuer or a third-party. On the other hand, some online models include Online Certificate Status Protocol (OCSP) [11], and Semi-Trusted Mediator (SEM) [15], which provide an online service used to obtain the revocation status. Similar to offline models, which provides the URI to request the location of the issuers CRL, online models also include the location of

this service within the issuer's certificate. The main drawback of offline models is that they require frequent downloads to keep the revocation list current. Online models overcome this limitation by checking the certificate status in real time.

Our revocation approach is inspired by OCSP [11]. Once a node requests a key through our trust model, and it has been accepted by our policy, it will cache the key/certificate until it implicitly expires or is explicitly revoked. When the node sends a key request interest message, the key response message contains an ordered list of Revocation Authorities (RA), providing an online service that states the status of the issuer's certificate (similar to the OCSP URI in the certificate [11]). The next time a node requests data from the same issuer with a cached key, it would only need to request the status of the certificate from the RAs. RAs would reply back with a signed *Valid*, *Invalid*, or *Unknown* response. A nonce would also be included in this reply to prevent replay attacks when a malicious user was to capture a valid response and replay it during another status request. Note that the RA themselves would also have a trusted certificate which will be obtained using our trust model. Any "invalid" responses to a certificate status results in the node clearing its certificate cache and falling back to the trust model to obtain a trusted key.

However, this model is still vulnerable to attack. If a DoS attack on the RA would prevent a response with the current status of the certificate, this model considers this timeout to be an *invalid* response. Further, if a malicious user does compromise an RA there are a few possible outcomes. Either the key is still valid but the compromised RA would respond with *invalid*, which would result in the node clearing its cache and returning to the start of the trust model. The other possibility and more threatening to the user is the key has been revoked, but the compromised RA response back with a *valid* response. The node maybe be fooled into accepting non-valid content from the issuer of the certificate. However, this alone would not be a meaningful attack. For a more successful attack, a node would have to first trust a key through our trust model, say for example `cnn.com`. If the publisher then revokes its compromised key, the RA's would be notified, and would respond with *invalid*. The malicious user must then compromise both the RA's for `cnn.com` and `cnn.com` itself. Then wait to intercept a request for both `cnn.com` data and replace it with its malicious content signed with the compromised key, then also intercept the certificate status request to any of the RA's and respond with a *valid* message. This makes an attack against our revocation model costly, specially if the RA list contains more than one RA in which all RA's responses must be forged.

## 2.6 Key-trust Policies

Once the user has key response messages from its trust community, it uses a key-trust policy to accept or reject an offered key based on the collected information.

Basically the user must make a decision which leads to a security vs. availability trade-off. The user can take the risk of accepting the offered key based on the responses from its friends or reject the key and discard the content received. Every user has a liberty of choosing a local policy most suitable for them. Here, we discuss Perspectives's policy model [4] as an example of a simple and yet effective local policy implementation. The notion of quorum [4] as a key-trust primitive is defined as followed:

**Definition:** For a set of  $n$  entities in a community of trust, a service  $S$ , and a threshold  $q$  ( $0 \leq q \leq n$ ) we say that a key  $K$  has a quorum at time  $t$  iff at least  $q$  of the  $n$  entities report that  $K$  is the key for  $S$  at time  $t$ .

As an example, consider our trust community depicted in Figure 2. Let's assume that the user has received the following responses from the community as illustrated in Table 1.

Table 1: An example of key recommendations from the community of trust

Notary#1	Friend#1	Notary#2	Friend#2	Friend#3
$K_A$	$K_A$	$K_C$	$K_B$	$K_A$

If the user's quorum is  $0.6n$ , then it accepts the offered key from `cnn.com`. Any higher quorum, however, would reject it. Notice that the security vs. availability trade-off is apparent in this case when a user can simply set the quorum to total number of friends,  $n$ , where this provides the strongest protection against accepting a false key, but it means that a single unavailable or misinformed friend could cause the user to reject a valid key.

Now that we have learned the details of the design, to better understand the model, let us turn to the example depicted in Figure 3. The scenario is that the user downloads an article from `cnn.com`. We assume that the user's cache does not contain `cnn.com`'s master key. Hence, upon receipt of the article, the user issues a series of key interest requests to his trust community and asks for `cnn.com`'s master key. The user asks `cnn.com` directly for its master key as well (offered key). If the friend (or notary) has the key in interest cached, which in that case he would prepare the signed key response message that includes the key. But if the friend does not have the key, he issues his own key interest request messages to his community of trust, accepts or rejects the key, and forwards his decision to the user. Once the key recommendation responses are received, the user's local key-trust policy decides whether to accept the key or not. If he accepts, the user caches the `cnn.com`'s master key, and will be able to validate any certificate generated by `cnn.com` or any of its sub-domains. Using this information,

the user can verify the authenticity of any article published by `cnn.com`.

To summarize, the content name in the key request interest message from user  $A$  to a friend  $B$  in  $A$ 's community of trust would be:

$$B : A, \sigma_{k_A}(P, \tilde{n})$$

where  $k_A$  is  $A$ 's private key,  $P$  is the publisher whose key is in question, and  $\tilde{n}$  is the nonce used for this interest message. And the data in the key response message from  $B$  would be:

$$B, \sigma_{k_B}((k_P, t, RA), \tilde{n})$$

where  $k_B$  is  $B$ 's private key, and  $(k_P, t, RA)$  is the publisher's key information (key, expiration date, and the RA ordered list).

### 3. Security Analysis

To better understand the robustness of our approach against attacks, we examine the following attack scenarios and analyze their effectiveness in details.

#### 3.1 Man-in-the-Middle Attack

In this scenario, the attacker compromises the link between the user and one of his trust community entities, in an attempt to make the user to believe a false key. However, to successfully launch this attack, the attacker needs to forge the signatures of a sufficient number of the user's friends in the key response messages. Note that the number of messages need to be forged depends on the user's local policy. Forging signatures from multiple friends is a costly attack on just a single user, specially given that the friends' keys are assumed to be obtained by the user through out-of-band means.

In a different scenario the attacker who has compromised the link from the user to his friends could drop the legitimate key response messages, in an attempt to make the user to accept the false key by majority rule. The way to prevent this is by checking the quorum by the percentage of queries sent and not but by the percentage of responses received. Considerable amount of lost responses is suspicious and relevant actions must be taken.

#### 3.2 Denial-Of-Service Attack

Attackers can perform a Distributed Denial-of-Service (DDoS) attack on a particular Revocation Authority, overwhelming it by generating many queries to it. This will lead to the RA not being responsive to legitimate validation queries by users. In this case, the user will not know whether the key of interest is still valid or not. This form of attack is not potentially helpful if not coupled with key compromise of a well-known publisher associated to the RA. But still, this could be mitigated by having ordered list of RA's to ask for instead of only one for particular content-sensitive publishers.

#### 3.3 Replay Attack

In this case, the attacker stores a user's friend's response for a particular key for later replay. This is particularly effective when an emergency rollover happens, making the key no longer valid. By replaying the same invalidated key, the user will receive false key information. To remedy this, a nonce is used in key request interest messages which needs to be included in response messages. Same technique can be used to protect the user against replay attacks against RA responses.

#### 3.4 Compromised Notary

There could be a scenario that notary gets compromised and turns malicious. In that case it could potentially send incorrect information to the user. However, not following the complete trust principle immunizes the user from harms of a single compromised notary.

#### 3.5 Compromised Friend

A compromised friend, similar to a compromised notary, could send false key information. As discussed in the previous section this is generally not an effective attack.

A compromised friend can also perform a DoS attack on the user by abusing the user's resources. To achieve this, the compromised friend sends overwhelming number of key request interest messages to the user. However, this can be mitigated by limiting the rate of queries sent by a particular friend.

#### 3.6 Compromised Revocation Authority

A compromised revocation authority lies about the validity of a particular key. We examine both of possible scenarios. In the first scenario, the key is still valid and RA advertises that it's invalid. In this case, the user removes the key from the cache and generate key interest requests to his friends for the key. The user is expected to still believe in the correct key after consensus from his friends. In the other scenario, RA can be potentially harmful by advertising a prematurely revoked key as valid. The attacker, in this case, will only benefit from such false information, if the key for the publisher is also compromised by the same attacker. This is believed to be a hard task since the attacker must learn the keys for both the publisher and the associated RA at the same time and remain undetected throughout the attack.

## 4. Evaluation

While there are no standard means for evaluating various trust model implementations [10], there is a set of desirable characteristics that any Internet-scale system should have to succeed. These characteristics are distributed authority, independent policy, scalability, and easy enrollment. Our proposed trust model has these characteristics since there is no single or multiple globally central authorities in this trust model which also leads to scalability. Also in our

model, every user has the liberty of who to trust and what local policy to implement. This model also promotes easy enrollment since any user at any time can join and gradually expands its community of trust.

## 5. Conclusion and Future Work

In this paper, we introduced and presented the details of a trust model for NDN that gives more freedom to the user to make his own trust decisions. We then analyzed various attack scenarios and discussed how this design would thwart each attack.

This system could be further improved by a well defined reputation system that assists the user in deciding the degree of trust on a friend or a notary. It would then required well-defined metrics, such as correct responses ratio and responsiveness. This reputation system should be able to detect friend's misbehavior over time, and remove him from the user's friend list. Also, privacy issues involved in this design should be further investigated. As an example, one concern involved in this trust model, potential privacy issues by sharing what content the user is interested in with his friends.

## References

- [1] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L. 2009. "Networking named content". In *Proceedings of the 5th international Conference on Emerging Networking Experiments and Technologies* Rome, Italy, December 01 - 04, 2009.
- [2] D. K. Smetters and V. Jacobson. "Securing network content", October 2009. PARC Technical Report.
- [3] Osterweil, E., Massey, D., and Zhang, L. 2009. "Managing Trusted Keys in Internet-Scale Systems". In *Proceedings of the 2009 Ninth Annual international Symposium on Applications and the internet (July 20 - 24, 2009)*. SAINT. IEEE Computer Society, Washington, DC, 153-156.
- [4] Wendlandt, D., Andersen, D. G., and Perrig, A. 2008. "Perspectives: improving SSH-style host authentication with multi-path probing". In *USENIX 2008 Annual Technical Conference on Annual Technical Conference (Boston, Massachusetts, June 22 - 27, 2008)*. USENIX Association, Berkeley, CA, 321-334.
- [5] Osterweil, E., Massey, D., Tsendjav, B., Zhang, B., and Zhang, L. 2006. "Security through publicity". In *Proceedings of the 1st USENIX Workshop on Hot Topics in Security (Vancouver, B.C., Canada)*. USENIX Association, Berkeley, CA, 3-3.
- [6] Blaze, M., Feigenbaum, J., and Lacy, J. 1996. "Decentralized Trust Management". In *Proceedings of the 1996 IEEE Symposium on Security and Privacy (May 06 - 08, 1996)*. SP. IEEE Computer Society, Washington, DC, 164.
- [7] C. M. Ellison, B. Frantz, B. Lampson, R. Rivest, B. M. Thomas, and T. Ylonen. "SPKI Certificate Theory", September 1999. RFC2693.
- [8] R. L. Rivest and B. Lampson. "SDSI - A Simple Distributed Security Infrastructure". Technical report, MIT, 1996.
- [9] A. Lenstra and E. Verheul. "Selecting cryptographic key sizes". *Journal of Cryptology*, 14(4):255-293, 2001.
- [10] Wojcik M, Venter HS, Eloff JHP: 2006. "Trust Model Evaluation Criteria: A Detailed Analysis of Trust Evaluation", In *Proceedings of the ISSA 2006 from Insight to Foresight Conference, Information Security South Africa*, pp 1-9.
- [11] Myers, M., R. Ankney, A. Malpani, S. Galperin and C. Adams, "Online Certificate Status Protocol - OCSP", RFC 2560, June 1999.
- [12] <http://www.ccnx.org/>
- [13] Housley, R., Polk, W., Ford, W. and D. Solo. "Internet X.509 Public Key Infrastructure: Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.
- [14] Paul C. Kocher. "On certificate revocation and validation". In *Financial Cryptography*, pages 172-177, 1998.
- [15] Dan Boneh, Xuhua Ding, and Gene Tsudik. 2004. "Fine-grained control of security capabilities". *ACM Trans. Internet Technol.* 4, 1 (February 2004), 60-82.

## Practical IDS alert correlation in the face of dynamic threats

Sathya Chandran Sundaramurthy, Loai Zomlot, Xinming Ou  
 Kansas State University, Manhattan, Kansas, USA  
 {sathya, lzomlot, xou}@ksu.edu

### Abstract

A significant challenge in applying IDS alert correlation in today's dynamic threat environment is the labor and expertise needed in constructing the correlation model, or the knowledge base, for the correlation process. New IDS signatures capturing emerging threats are generated on a daily basis, and the attack scenarios each captured activity may be involved in are also multitude. Thus it becomes hard to build and maintain IDS alert correlation models based on a set of known scenarios. Learning IDS correlation models face the same challenge caused by the dynamism of cyber threats, compounded by the inherent difficulty in applying learning algorithms in an adversarial environment. We propose a new method for conducting alert correlation based on a simple and direct semantic model for IDS alerts. The correlation model is separate from the semantic model and can be constructed on various granularities. The semantic model only maps an alert to its potential meanings, without any reference to what types of attack scenarios the activity may be involved in. We show that such a correlation model can effectively capture attack scenarios from data sets that are not used at all in the model construction process, illustrating the power of such correlation methods in detecting novel, new attack scenarios. We rigorously evaluate our prototype on a number of publicly available data sets and a production system, and the result shows that our correlation engine can correctly capture almost all the attack scenarios in the data sets.

### I. INTRODUCTION

IDS alert correlation has been studied for more than a decade and a number of correlation methodologies have been proposed. Although research has made significant progress in creating various correlation models, what one finds in practical use still remains rudimentary. Our conversation with system administrators and security analysts indicate that there is a significant gap between the desired capability of IDS alert/event correlation technologies and what the current commercial tools can provide. The question naturally arises that why more than ten years' research into IDS alert correlation has not found wide-spread use in practice.

Several attempts in applying IDS alert correlation have suffered from many of the following limitations. Most of the approaches are based on constructing a knowledge base of known attack scenarios and thus lack the ability to detect emerging and new ones. Another problem is the difficulty in updating the knowledge base. If there is a same attack with slight modification in the sequence of events, the knowledge base may not be able to detect it. A more important concern with past works on alert correlation is the lack of rigorous evaluation on a number of data sets. For an IDS to be a practical tool it has to perform consistently over a variety of data sets.

In this paper we propose a simple and practical approach to IDS alert correlation that addresses the above challenges. Our contributions are:

- Our approach is *generic* in that it is not customized to detect pre-modeled scenarios but has the ability to detect previously unseen attack scenarios. It is also *flexible* in the sense that it can handle data from a variety of sensors like NIDS, HIDS and other relevant information for intrusion analysis. Adding a new sensor is easily done by adding a new semantic mapping for the information the sensor reports. The correlation graphs can be generated with any desired level of granularity, depending on how detailed the system administrator wants to know about specific attack scenarios.
- The correlation tool we develop can report attacks, if any, in *real time*. Our correlation tool handles network traffic *continuously*, generating attack graphs in .svg file format deployed as a web page on a web-server. The correlation is staged in two phases wherein we do the time-consuming activities like alert grouping and summarization in the first phase and the reasoning engine that takes those processed alerts, generating attack scenarios constitutes the second phase.
- We have done rigorous evaluation of the efficiency of our tool on a number of data sets that range over a wide period of time. We call it rigorous because we used the same semantic model consistently over all the data sets and the result shows that the same model works perfectly on all of them.
- The attack scenarios produced from our correlation tool can be used as input for various prioritization methods.

### II. RELATED WORK

Alert correlation in intrusion detection systems has been a topic researched for almost ten years [1, 3, 5, 6, 8, 13]. Ning, et al. [7] proposed an approach using pre and post-conditions. The concept of hyper-alerts are introduced, which consists of the attack activity, the pre and post-conditions corresponding to that attack. The mappings of raw alerts to one of these hyper-alerts are pre-generated and stored in a knowledge base. Two hyper-alerts are correlated if the post-condition of one hyper-alert contributes to the pre-condition of another one. Cuppens, et al. [3] propose a correlation model called CRIM which provides clustering, merging and correlating of alerts from multiple IDSes. The alerts are clustered based on their similarity and merged, where each group of merged alerts represents a single attack. The correlation module takes these merged alerts and constructs a number of possible attack scenarios. CRIM specifies a number of attack modules using the LAMBDA language where each module has a pre-condition that must be true for the attack to take place, the post-condition that may result if the attack

succeeds, the attack activity itself, and other information. To correlate the modules and construct the attack scenarios the authors propose two methods, direct and indirect correlation. In the explicit correlation method two modules are correlated if successful execution of one module contributes to the initiation of another. In the indirect correlation approach ontological rules are used to correlate modules that aren't directly related but through a series of events, with one module being the initial event and the other the last event. Cheung, et al. [2] proposed a model called Correlated Attack Modeling Language (CAML) that aims at developing attack patterns. The main idea is to construct a set of modules that describe specific attacks with pre-conditions to be satisfied for the attack to occur, the attack activity itself and the post-condition that may result if the attack succeeds. The post condition of one attack module may satisfy the precondition of some other attack modules in which case both these attack modules will be linked.

Most of the above previous works adopt a pre- and post-condition correlation module. A potential drawback of ascribing a pre- and postcondition to an IDS alert is that the model itself may already have assumed some specific attack patterns. The attack modules knowledge base must be frequently updated to include newer attack patterns, which has become impractical in today's rapidly changing threat environment.

Ren, et al. [11] propose the design of an online correlation system for real-time intrusion analysis. There are two components: an off-line Bayesian-based knowledge base construction tool and an online correlation and attack graph constructor. The offline component maintains tables that specify the frequency of occurrence of possible hyper-alert types and also the correlation between different hyper-alert pairs. This correlation is dynamically updated depending upon the network traffic observed over a past time window. The online component, on receiving alerts and grouping them into different hyper-alerts, uses the knowledge base from the offline component to construct the attack graph. This approach is practical since it moves to offline the work that involves maximum processing, and it is dynamic, ie., able to adjust the causality between hyper-alerts depending on the network traffic. However, the automatic knowledge base construction can only learn and detect the type of attacks that have occurred in the past and on the network where it is trained. We propose a different approach where a generic knowledge base captures an attacker's intentions and constraints, instead of specifics of attacks.

### III. CORRELATION MODEL

The biggest problem with IDS is the large volume of false alarms. IDS alert correlation can potentially help in finding attack traces in all these alerts. Our correlation model is built within the context of the SnIPS tool suite [4]. A key feature of SnIPS is using qualitative uncertainty tags and a *proof strengthening* techniques to handle the uncertainty challenge in intrusion analysis [9]. The new correlation model described in this paper would allow for more sophisticated mathematical theories to be applied to handle uncertainty, as compared to the empirically developed proof strengthening approach.

Figure 1 shows the overall architecture of the SnIPS system. Our correlation model is based on a direct semantic model for

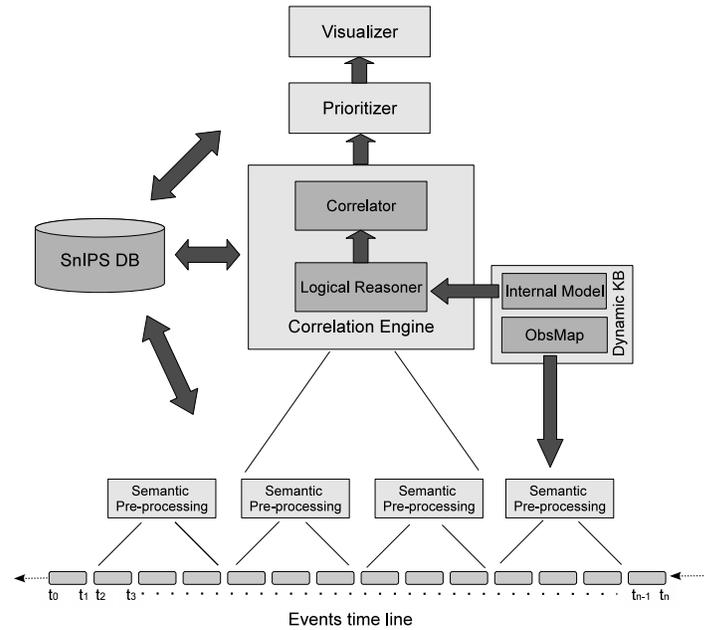


Figure 1. SnIPS System

IDS alerts, which maps an alert to its potential meanings. The semantic pre-processing layer applies this semantic model to translate raw alerts into high-level “summarized alerts” which are grouping of alerts with similar properties (source/destination address and alert type). The correlation engine then applies a two-stage process to build up an alert correlation graph. The correlation system is capable of handling dynamic knowledge base such as black-listed IP addresses which change frequently. It provides online real-time response, running continuously on streams of input events.

#### A. Semantic pre-processing

The pre-processing step is performed to translate and reduce the amount of information entering the reasoning engine. This process consists of the following parts.

1) *Translation*: SnIPS takes Snort alerts and other relevant events as input. Then it maps the observations (events) to their semantics. This process uses a set of mapping rules called *obsMap*.

**Definition 1.** Observations  $\xrightarrow{mode}$  Internal Conditions

Definition 1 shows the formal mapping rule between observations (e.g. alerts) and internal conditions (semantics). The *mode* is used as a tag indicating the strength of the belief (e.g. *unlikely, possible, likely, or certain*). The assignment of the mode is done by interpreting the natural language description of the snort rule (IDS signature).

**Example 1.** *Observation mapping*:

$obs(portScan(H1,H2)) \xrightarrow{p} int(probeOtherMachine(H1,H2))$

Example 1 shows an *obsMap* rule that maps a *portscan* alert to probing activity with the *p* (possible) mode.

2) *Summarization*: The summarization step is performed to reduce the amount of information entering the reasoning engine. We apply a data abstraction technique by grouping a set of similar “internal conditions” into a single “summarized” internal

$$\left. \begin{array}{l} \text{int}(\text{probe}(\text{ext}_1, H), c, T_1) \\ \text{int}(\text{probe}(\text{ext}_2, H), c, T_2) \\ \dots \\ \text{int}(\text{probe}(\text{ext}_n, H), c, T_n) \end{array} \right\} \text{int}(\text{probe}(\text{external}, H), c, \text{range}(T_1, T_n))$$

Figure 2. Summarization

condition. The summarization is done on both the time stamps and IP addresses. For timestamps, if a set of internal conditions differ only by timestamp we merge them into a single summarized internal condition with a time range between the earliest and latest timestamp in the set. We also abstract over external IP address. We begin by selecting a set of internal conditions that differ only in the external source or destination IP addresses, and give a special variable, “external” as an abstraction of the external IP addresses. We do not summarize on internal IP addresses as this knowledge may be useful in the reasoning process. We maintain the mapping between the summarized internal condition and the raw internal conditions/observations in the SnIPS database, which helps us identify the low-level facts belonging to the summarized predicates. The summarized tuples are then given to the reasoning engine. The output of this module will be stored in the SnIPS database. Figure 2 illustrates the summarization process.

3) *Black List IP Processor*: SnIPS can digest any emerging information about the threat landscape and use them in the reasoning process. All that is needed is to provide an obsMap rule for the information. One example of such dynamic information is black listed IPs. A machine can be put into a blacklist if it has been found to be involved in malicious activities (bot activities, ssh brute-force log in attempts, etc.). Such a list can be used in two ways. The first method is to map a blacklisted IP to the predicate *compromised*, with the mode assigned by the IP’s age in the list. That is, the IP address will be mapped to a higher mode if it is more recently added to the list. Over time the confidence will decrease, due to the fact that the machine could have been cleaned up. The second method is to create a snort rule that will be triggered whenever there is a communication between any local host and the black-listed IP. This alert will be mapped using obsMap rules stored at the SnIPS dynamic knowledge base (Figure 1).

**Example 2.** *H2 is a black-listed IP:*

$$\text{obs}(\text{anyCommunication}(H1, H2)) \xrightarrow{c} \text{int}(\text{compromised}(H1))$$

The advantage of the second method is that it can capture all communication with a black-listed IP even if it would not trigger an alert otherwise.

## B. Correlation Engine

The goal of this step is to build the scenario picture of attacks and consists of two stages: the *Reasoner* and the *Correlator*.

1) *Reasoner*: The goal of this reasoning process is to find all the possible semantic links among the summarized facts. It uses an *Internal Model* (see Figure 1). Definition 2 gives the formal format for reasoning rules in the internal model (called *internal rules* thereafter). The rule derives one internal condition from another, with two qualifiers: *direction of inference*, and *mode*. The direction tag has two values either *backward* or *forward*. The mode tag has been discussed before.

**Definition 2.** *Condition 1*  $\rightarrow$  *Condition 2*

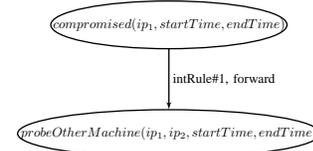


Figure 3. A proofStep example

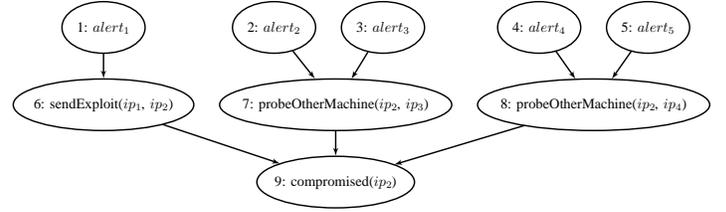


Figure 4. Correlation graph example

Example 3 illustrates one internal rule. If we know that machine  $H_1$  is compromised, then it may perform malicious probing to another machine  $H_2$ . Conversely, if we know that a machine  $H_1$  is performing malicious probing against another machine, we can also know that machine  $H_1$  is compromised. So each internal rule can be used either in the forward direction, or the backward direction, in the reasoning process.

**Example 3.** *Internal rule:*

$$\text{int}(\text{compromised}(H1)) \xrightarrow{f} \text{int}(\text{probeOtherMachine}(H1, H2))$$

The output of this stage is a collection of individual “proof steps” (*proofStep*). Figure 3 gives an example of a *proofStep*. Each node is associated with a fact like *compromised*( $H_1$ ), and a time range (*startTime, endTime*), indicating when the fact becomes true. The direction of inference (forward or backward) is also indicated in the proof step. The time range of the conclusion can be calculated based on the time range of the antecedent and the direction of the inference. All the proofSteps will be stored in the SnIPS database.

2) *Correlator*: The correlator module collects all the small pieces of evidence in the form of proofSteps into a possible scenario. The input of this engine is a list of *proofSteps* from the *reasoner*, and the output is a set of correlation graphs. Each graph is an illustration of attack scenarios gathered from all the pieces of evidence.

Figure 4 is an example correlation graph, which can be viewed from top to down. “*compromised*”, “*probeOtherMachine*”, and “*sendExploit*” are predicates used to describe various attack hypotheses. *alert<sub>1</sub>* is mapped to the fact that host  $ip_1$  sent an exploit to  $ip_2$ ; both *alert<sub>2</sub>* and *alert<sub>3</sub>* are mapped to the fact that  $ip_2$  did malicious probing to  $ip_3$ , and so on. The rationale behind this correlation graph is that after  $ip_1$  sent an exploit to  $ip_2$ ,  $ip_2$  may be compromised (node 9). Once the attacker has compromised  $ip_2$ , he can send malicious probing to any other machine. Thus these alerts are all potentially correlated in the same underlying attack sequence. Section IV explains the algorithm that computes these correlation graphs.

## C. Prioritizer

The prioritizer can further refine the result of correlation by assigning each node in the correlation graph a belief value

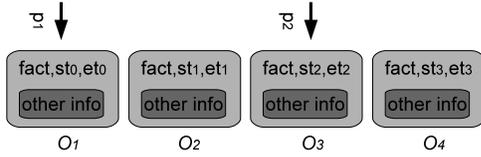


Figure 5. Correlation of different time ranged overlapping facts objects

based on an extended version of *Dempster-Shafer* evidential reasoning theory [12]. The belief values can be used to rank the correlation graph segments by the belief values of the nodes within the segments, the higher the belief value, the more likely the correlation represents a true attack. This will help the system admin to spot the most important correlation scenario. The calculation uses the mode values assigned to the observations and translates them into numeric basic probability assignment on the interpretations of the observations. Then an extended Dempster Belief Combination method is applied to calculate the belief value of each node in the graph. Detailed explanation of this module falls beyond the scope of this paper.

#### D. Visualizer

The final step is to introduce the output to the system admin in an easy to understand and manipulate manner. We use the *GraphViz* tool to display the correlation graphs in the *.svg* format, so that the user can use an interactive web interface to further analyze portions of a correlation graph. For example, the user can examine the raw alerts behind a summarized alert, the IDS signatures that trigger them, the payload, and other relevant information.

#### E. Dynamic Knowledge Base

The dynamic knowledge base is used in a number of steps described above. It includes both the *obsMap* relations used in the semantic pre-processing stage and the internal model used in the logical reasoning phase. The dynamism of this model comes from the fact that it can be automatically or manually updated based on the emerging threats. For example, the black-listed IP addresses change every hour. The simple *obsMap* relations allow for quick update of the *obsMap* relations for this piece of information.

#### F. Implementation

We use the Prolog system XSB [10] to perform the semantic mapping and logical reasoning of the input alerts. The Correlator and the Prioritizer are implemented in Java. The Visualizer is implemented using a collection of web programming languages such as PHP.

### IV. CORRELATION ALGORITHM

The correlation algorithm takes the collection of *proofSteps* output from the *Reasoner* and builds a set of correlation scenarios in the form of graph segments. The Correlator follows the following steps:

- **Step 1:** Translate the input *proofSteps* into a form that can be handled by the engine. We will call this translated object  $O_i$  (Figure 5). An object contains a number of fields, including the fact associated with it, the start time, and the end time.

- **Step 2:** All translated objects  $O_i$  will be classified based on the facts associated with them. For example, all objects with the fact *compromised(h)* will be in one group and so on. Figure 5 gives an example of one group. We can assume that each fact in the figure has the form *compromised(h)*. Each group of objects will be sorted ascendingly by the end time ( $et_i$ ), and then by the start time ( $st_j$ ).
- **Step 3:** Each group will then be correlated using time overlapping between objects. Figure 5 illustrates the correlation process. There are two sliding pointers to track the correlation process. The first pointer  $p_1$  will start at the first object  $O_1$ . The second pointer  $p_2$  will move to the second object. If the time range of  $O_2$  overlaps with  $O_1$  then the intersection of the two time ranges will be taken and stored in a variable *intTimeRange*. Pointer  $p_2$  then moves to  $O_3$  and takes its time range to intersect with *intTimeRange*. The process stops when *intTimeRange* becomes empty, meaning we can no longer correlate the facts represented by the objects. A new graph node will be created for all the objects that have a non-empty time-range intersection, which will have *fact, intTimeRange* as its fields. Then  $p_1$  will move forward until the time-range intersection for objects between  $p_1$  and  $p_2$  becomes non-empty.

Figure 1 and 2 show the pseudo code for this algorithm. Line 9 in Algorithm 2 includes constructing the edges of the graph, which utilizes the “other info” field in each object to connect the merged nodes with one another. Details of this part of the algorithm is omitted due to space limitation.

---

#### Algorithm 1 Correlation engine

---

```

1: function CORR(ProofStepsSet)
2:   ObjectsSet  $\leftarrow$  Translate all proofSteps in
   ProofStepsSet
3:   for each Object(O) in ObjectSet do
4:     ObjectsGroupList  $\leftarrow$  group by fact of Object O.
5:   end for
6:   for each ObjectGroupList do
7:     sort ascendingly by the endTime of the
   TimeRange.
8:     Graph  $\leftarrow$  CREATEGRAPH(ObjectGroupList)
9:   end for
10:  return Graph
11: end function

```

---

### V. EXPERIMENTATION

We have tested our correlation model on a number of publicly available datasets and on our departmental network as well. For the datasets the tcpdump of the network traffic is obtained and the correlation is done offline. **The construction of the reasoning model is done completely separately from the evaluation and without any knowledge about the specifics of the data sets.** This testing is to ensure that our correlation model works fine and is able to identify different types of attacks. The evaluation on the departmental network analyzes data from various sources like Snort IDS, black-list logs from computer clusters, and so on, and produces real-time attack scenario graphs in Scalable

**Algorithm 2** Create graph

---

```

1: function CREATEGRAPH(ObjectGroupList)
2:   for each Object( $O_i$ ) in ObjectGroupList do
3:     intTimeRange  $\leftarrow$  find the intersection of
       timeRange with the next  $O_i$ 
4:     if intTimeRange is Empty then
5:       NodesHash  $\leftarrow$  create new node with
       intTimeRange and  $O_i$ 's fact and use  $O_i$  as the key
       for the hash table.
6:     end if
7:   end for
8:   for each Node in NodeHash do
9:     Graph  $\leftarrow$  build the edges from the related  $O_i$ 
10:  end for
11:  return Graph
12: end function

```

---

Vector Graphics (svg) format which can be viewed on a web-browser. The generated svg file is hosted on a web-server so that the System Administrator can view the current security state of the system as the tool updates the file.

The Snort alert collection and correlation were carried out on a Ubuntu server running a Linux kernel version 2.6.32 with 16GB of RAM on an eight-core Intel Xeon processor of CPU speed 3.16GHz. So far we have not encountered any performance bottleneck in our algorithm.

We have tested our correlation system on the following data sets

- Lincoln Lab DARPA intrusion detection evaluation data set
- Honeynet Project
- Treasure Hunt

#### A. Lincoln Lab DARPA intrusion detection evaluation data set

1) *Lincoln Lab Scenario (DDOS) 2.0.2 Data Set*: In this attack scenario, the goal of the attacker was to break into a network on the internet through a remote buffer-overflow exploit, install software required to launch DDOS attack on the hosts inside the internal network. The attacker first breaks into the DNS server for the internal network through the remote *sadmind* buffer-overflow exploit and installs the *mstream* DDOS master software. He then uses the HINFO records in the captured machine to probe for machines within the internal network. Once he found the vulnerable hosts, he broke into them and installed the server software one of them. The vulnerability exploited is the well known *sadmind* vulnerability in the Solaris system. The attacker then telnets to the remotely compromised machine, starts the DDOS master and launches attack against other systems on the internal network.

We collected the tcpdump data from the Lincoln Lab website and ran Snort on it. The alerts from Snort were logged into a MySQL database. The correlation engine read the alerts from the database, did clustering and merging of the alerts, constructed an attack scenario based on the knowledge base and generated an svg file hosted on a web-server. The graph we obtained is shown in Figure 6. In this graph and the subsequent graphs the nodes of *box* shape represent Snort alerts and *oval* shaped nodes represent hypotheses. The first graph in the figure shows the

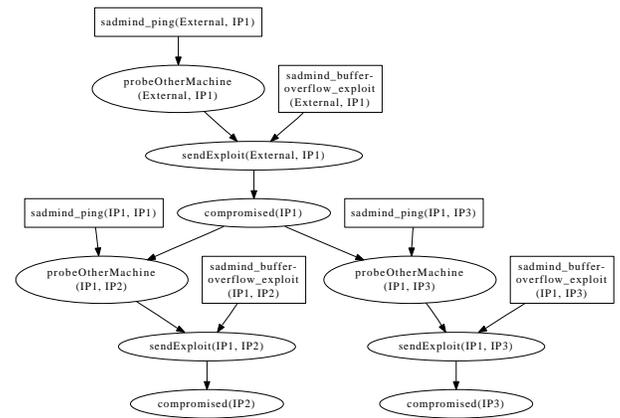


Figure 6. Lincoln Lab Scenario (DDOS) 2.0.2 Result

graph corresponding to the probing of the attacker for services that have the *sadmind* vulnerability. The alerts pertaining to this activity are grouped as a single group of snort alerts and are correlated to an abstract event that says *an internal machine, in this case the DNS resolver, is getting probed by some external IP*. The System Administrator can click the snort alerts block to see the alerts, their payload and a detailed description that says how this particular type of attack can occur. All these features are implemented as hyper-links in svg. The second graph segment shows how the DNS resolver of IP address IP1 is compromised along with the Snort alerts that support this hypothesis. There are also probes emanating from this compromised machine to internal machines of IP addresses IP2 and IP3. This captured the scenario described in the truth file that after capturing the DNS resolver the attacker started probing for other machines inside the internal network for more machines that run services with *sadmind* vulnerability. All the Snort alerts that supported the probe are grouped as an abstract predicate *probeOtherMachine*. The third and fourth segments capture the fact that the internal hosts are compromised through *sadmind* buffer-overflow exploit. This abstract representation depicts the scenario very clearly so that it is easy to interpret there are malicious activities going on. Further investigation for the specifics of the activity can be done by clicking the alerts.

2) *Lincoln Lab 1998 Intrusion Detection Data Set*: The 1998 intrusion detection data set from Lincoln Lab has both test and training data. Only the training data's truth file was publicly available. So we used the training data, which was seven weeks long in evaluating the performance of our correlation engine. In the first week's data there was a lot of background traffic and it was meant to test whether the IDS was working fine. It just had two attacks per day. Attacks were added gradually from the second week onwards. The traffic from second to the seventh week had 100 instances of 25 attack types.

We compared our results with the truth file published on the website (<http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/attacks.html>). Each day had specific types of attacks targeted at specific machines. The attack types included teardrop, land attack, ipsweeping, portsweeping etc., to name a few. Our correlation engine was able to correctly identify the machines that were under attack along with their attack type for each day of the data. The best part about this is that, we were able to identify the different types of attacks even without encoding the specific attack patterns. Our knowledge

base, which is an abstract model, successfully captured all the specific attack types and visualizing them as a svg file makes decision making an easy process for the user. Figure 7 shows the graph for the machine under land attack, captured by our correlation engine. Seeing this graph one can easily conclude that the machine is under land attack where one sends a packet to a machine with the same IP address and port number that causes the machine to lock itself. There are two sets of snort alerts, each identifying two different types of attacks. One group identified packets with the same source and destination IPs being sent and another group identifies smurf attack. The correlation engine identified these groups of alerts to be part of the same attack (the tcpdump has background traffic too) and built the scenario graph. There are two arrows pointing towards the node labelled “compromised(IP1)”. One of them is because we have a sendExploit, for which an internal model rule says there can be a possible compromise. Applying the internal rule in the backward direction we can say that if we have a compromised host then there might have been an exploit sent to it.

This graph is just one instance of the attack type captured by our correlation model. We captured 90 of the 100 attack instances. The false negative is due to the fact that we used the latest Snort signatures that do not contain attack rules to capture attacks that were prevalent in 1998. Nevertheless, we tried to obtain the older Snort rules but we couldn't obtain signatures old enough to capture those remaining 10 attacks.

### B. Data Set from the Honeynet Project

This data set is from a forensics challenge organized by The Honeynet Project, an international non-profit research organization in security. The data set we obtained is from the event Scan 34 (<http://old.honeynet.org/scans/scan34/>). The challenge was to analyze the various log files posted on the website and to figure out what exactly happened in the honeynet. The honeynet had 3 systems named *bridge*, *bastion* and *combo*. The *bridge* machine performed routing and filtering, *bastion* was the IDS running Snort and *combo* was the victim machine that was assigned 11.11.79.67 as its IP address with many other virtual IP addresses in the vicinity of its physical address. We obtained the Snort log file and converted it into an observation file readable by our correlation engine.

We were able to confirm that the honeynet was compromised which conforms to the ground truth published in the website (<http://old.honeynet.org/scans/scan34/sols/sotm34-anton.html>). Some of the noise correlations include MySQL worm attack against the bridge which was not part of the honeypot. This was due to the fact that the Snort IDS was not configured to be context aware and so it generated alerts for packets that were indeed malicious but weren't targeted towards the honeypot. Figure 8 is a correlation graph for one of the attacked machines. First we have exploits being sent from an external machine to a machine inside the honeypot and the internal machine gets compromised. There are Snort alerts to support the fact that the machine is compromised. The internal machine then starts probing for vulnerable services on other machines, which is a typical attacker action. All the individual attack steps are indeed supported by some Snort alerts but they form an attack scenario

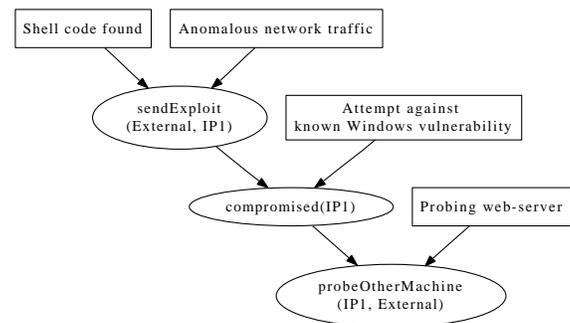


Figure 8. Honeynet Result

only when they are correlated and that's what exactly SnIPS does.

### C. Data Set from the Treasure Hunt event - UCSB

Treasure Hunt [14] is an event organized as part of the graduate-level security course at the University of California at Santa Barbara. The class was divided into two teams: Alpha and Omega and the goal was to compete against each other in breaking into a payroll system and performing a money transaction. To avoid interference with each other there were two identical subnets provided separately for both the teams. Each team has to perform a multi-stage attack and they had a number of tasks to do, each within a time period of 30 minutes. The first team to finish the task gets the higher points. The following is a general description of the two identical subnets. There was a webserver in a DMZ zone accessible directly outside the Local Area Network. There was a file server, a MySQL server and a transaction server. One must first compromise the web server in order to access the other servers. The task is to compromise the web server and then change the entries in a specific table in the MySQL database and then exploit the transaction service vulnerability and schedule a paycheck transfer.

We obtained the tcpdump data from the Treasure Hunt web site and ran Snort on it. Next we ran SnIPS on the alerts generated by Snort. The SnIPS system generated correlation graphs. We were not able to find the truth file for this activity but we can certainly say that every packet was an attack packet as there was no legitimate background traffic. We were able to identify the multi-stage attack in the packet capture of both Alpha and Omega teams. Figure 9 shows the correlation graph generated for the Alpha team by SnIPS. The correlation graph corresponds to the attack activities during the entire event. Figure 9(a) shows how the web server got compromised and started probing for file server, MySQL server and the transaction server. These probes are malicious and are meant for finding out vulnerable services and exploiting them. Figure 9(b) is a correlation graph for the event that the file server got compromised and Figure 9(c) for MySQL server probing for the file server. Within Figure 9(c) we can see that the web server gets compromised and then probes for the other servers which is exactly the requirement of that assignment.

## VI. CONCLUSION AND FUTURE WORK

In this paper we presented a technology to correlate intrusion detection alerts. The strength of this technology comes from its flexibility to handle the dynamism of the emerging new threats.

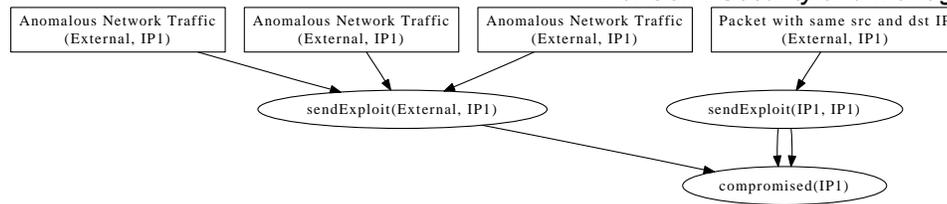


Figure 7. Lincoln Lab 1998 Result

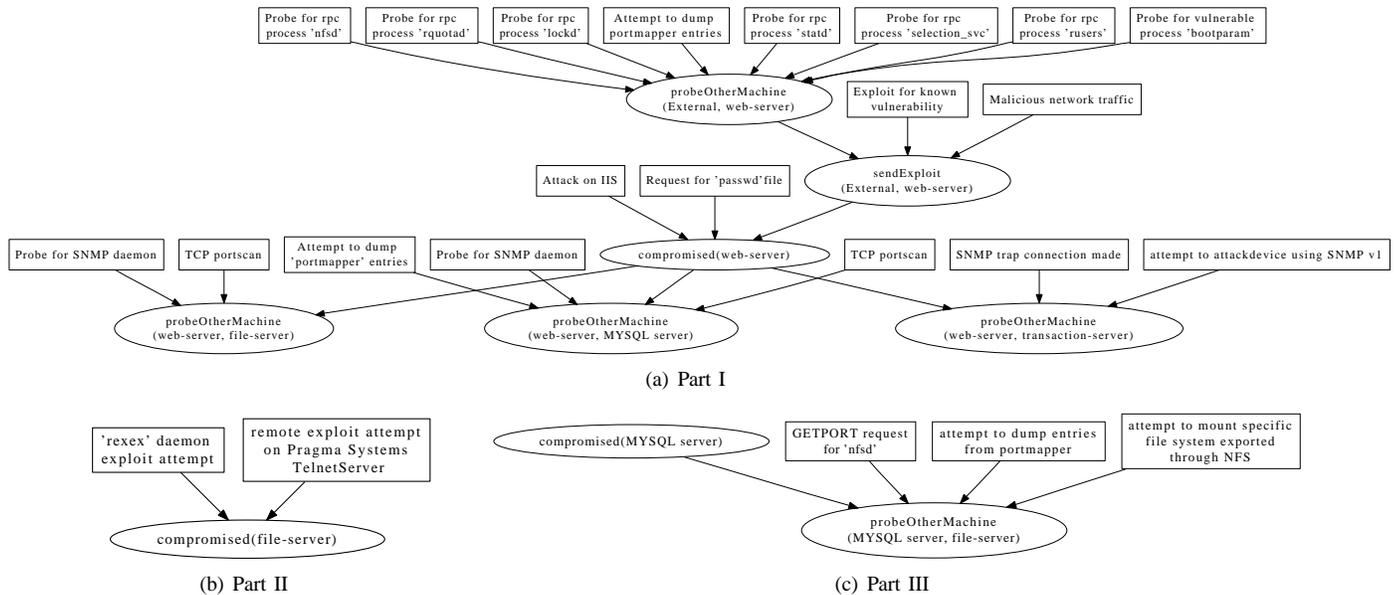


Figure 9. Treasure Hunt Result

This tool is a real-time engine to help the system administrator to spot attack scenarios on the fly. We have performed rigorous evaluation of the correlation model, and the results indicate that such a correlation model can effectively capture a variety of attack scenarios from a number of data sets.

The correlation technology forms a solid base to build other analysis theories, making the output more precise and useful. For example the prioritizing engine currently uses an extended version of Dempster-Shafer theory. Incorporating more information sources into the system can also make the attack scenarios more accurate, and the correlation model allows for easy extension due to its straightforward semantics. We already have the ability to handle a dynamic black-list IP addresses. Our plan is to widen the window of our system by getting information from multiple sources. This can be done by adding system and server logs, and IDS systems other than Snort could all provide valuable intrusion information. Once such information can be consumed by an inference system and a more accurate correlation graph will be produced.

## VII. ACKNOWLEDGMENT

This material is based upon work supported by U.S. National Science Foundation under grant no. 1038366 and 1018703, AFOSR under Award No. FA9550-09-1-0138, and HP Labs Innovation Research Program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, AFOSR, or Hewlett-Packard Development Company, L.P.

## REFERENCES

- [1] Steven Cheung, Ulf Lindqvist, and Martin W Fong. Modeling multistep cyber attacks for scenario recognition. In *DARPA Information Survivability Conference and Exposition (DISCEX III)*, pages 284–292, Washington, D.C., 2003.
- [2] Steven Cheung, Ulf Lindqvist, and Martin W Fong. An online adaptive approach to alert correlation. In *DARPA Information Survivability Conference and Exposition (DISCEX III)*, 2003.
- [3] Frédéric Cuppens and Alexandre Miège. Alert correlation in a cooperative intrusion detection framework. In *IEEE Symposium on Security and Privacy*, 2002.
- [4] Argus Lab. Snort intrusion analysis using proof strengthening (SnIPS). <http://people.cis.ksu.edu/~xou/argus/software/snips/>.
- [5] Benjamin Morin, Hervé, and Mireille Ducassé. M2D2: A formal data model for IDS alert correlation. In *5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002)*, pages 115–137, 2002.
- [6] Peng Ning, Yun Cui, Douglas Reeves, and Dingbang Xu. Tools and techniques for analyzing intrusion alerts. *ACM Transactions on Information and System Security*, 7(2):273–318, May 2004.
- [7] Peng Ning, Yun Cui, and Douglas S. Reeves. Constructing attack scenarios through correlation of intrusion alerts. In *Proceedings of the 9th ACM Conference on Computer & Communications Security (CCS 2002)*, pages 245–254, 2002.
- [8] Steven Noel, Eric Robertson, and Sushil Jajodia. Correlating Intrusion Events and Building Attack Scenarios Through Attack Graph Distances. In *20th Annual Computer Security Applications Conference (ACSAC 2004)*, pages 350–359, 2004.
- [9] Xinming Ou, S. Raj Rajagopalan, and Sakthiyumaraja Sakthivelmurugan. An empirical approach to modeling uncertainty in intrusion analysis. In *Annual Computer Security Applications Conference (ACSAC)*, Dec 2009.
- [10] Prasad Rao, Konstantinos F. Sagonas, Terrance Swift, David S. Warren, and Juliana Freire. XSB: A system for efficiently computing well-founded semantics. In *Proceedings of the 4th International Conference on Logic Programming and Non-Monotonic Reasoning (LPNMR'97)*, pages 2–17, Dagstuhl, Germany, July 1997. Springer Verlag.
- [11] H. Ren, N. Stakhanova, and A. Ghorbani. An online adaptive approach to alert correlation. In *The Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA)*, 2010.
- [12] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
- [13] Fredrik Valeur, Giovanni Vigna, Christopher Kruegel, and Richard A. Kemmerer. A Comprehensive Approach to Intrusion Detection Alert Correlation. *IEEE Transactions on Dependable and Secure Computing*, 1(3):146–169, 2004.
- [14] G. Vigna. Teaching Network Security Through Live Exercises. In C. Irvine and H. Armstrong, editors, *Proceedings of the Third Annual World Conference on Information Security Education (WISE 3)*, pages 3–18, Monterey, CA, June 2003. Kluwer Academic Publishers.

# Twitter on Drugs: Pharmaceutical Spam in Tweets

Chandra Shekar, Kathy J. Liszka, and Chien-Chung Chan

Department of Computer Science, University of Akron,  
Akron, Oh 44325-4003, USA  
{liszka, chan}@uakron.edu

**Abstract**—Twitter presents a new forum for spammers to facilitate illegal pharmaceutical scams. We present a classification scheme using decision strategy and data mining techniques taking into account the unbalanced nature of the data set. Four classifiers are used to identify pharmaceutical spam tweets. Classifiers J48 and Random Tree (RT) are generated by Weka tools, and classifiers DL(J48) and DL(RT) are based on the combination of J48 and RT with the decision matrix. The classifiers were tested using manually labeled data sets collected at different time spans. Experimental results suggest that the combination of RT with the decision matrix provides a stable performance improvement over using stand-alone tree-based classifiers.

**Keywords**-data mining; text mining; spam; pharmaceuticals; social networking; Twitter; microblogging

## 1 INTRODUCTION

The growth of social media is phenomenal, and so is its impact on our daily lives. Even those who do not directly engage in microblogging technology are affected by it. It has changed our lives and the way we communicate in ways we never imagined. We focus this research on an insanely popular social networking tool called Twitter [1]. Created by a small, ten person startup company in San Francisco, the microblogging service launched in July 2006. Their intent was to create a way for people to stay continually connected in a simple, convenient way. The obvious application is to keep in touch with busy friends, evoking content such as “Drinking double caramel mocha latte at Starbucks” and “Monday afternoon blues.”

Other uses quickly became obvious and suddenly Twitter took on different dimensions. People looked to Twitter for communication and coordination after Hurricane Katrina hit and later, while waiting for Hurricane Gustav. Similarly Twitter was used to coordinate hostage situations during the terrorist attacks in Mumbai, in a completely ad hoc fashion, proving very effective. During the recent riots in Cairo, Twitter was used to report events to the rest of the world [2]. Then came the ripple effect as the world watched Egypt pull

Internet services from its population. What started out as a casual tool built on instant messaging, to keep in touch, has become a political tool. Twitter has been used in Iran and Moldova as one tool to organize anti-government protests [3]. Politically or humanitarially motivated, Twitter can, and has been used for proactive disaster information dissemination and relief.

With all the attention in the media about microblogging, yet another application has emerged. Spammers have pounced on this new opportunity for a fast and dishonest dollar, disseminating pornography, fake lotteries, fake inheritances, counterfeit watches, free software, and illegal or ineffective pharmaceuticals. Phishing sites and malware threats are now a fixed part of the Twitter-verse. This is a shame because of the untapped and as yet, undiscovered potential for Twitter and other similar social networking platforms. While it hasn't reached the level of invasiveness that it has on email, the annoyance level, as well as the dangers are still present. The potential for phishing, and thus malware infection and identity theft, is very real.

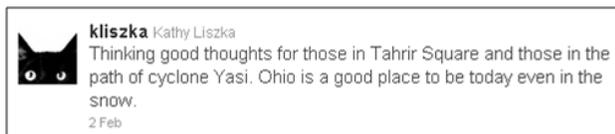
In this paper, we focus on pharmaceutical spam; those tweets related to sale of advertisement of pharmaceuticals such as Viagra, Levitra, and Xanax. According to Symantec [4], pharmaceutical spam currently accounts for 65% of all traditional email spam sent. That statistic alone gives motivation to study pharmaceutical spam in the context of microblogging. What we have found is that illegally pushing prescription drugs, or scamming people with sugar pills instead of legitimate drugs, fortunately has not caught on as rampantly in Twitter. It does, however exist and pose a threat to people naïve enough to believe that they can purchase their medications cheaper through these scammers. Anecdotally, during a visit to the Twitter website on a cold February afternoon, a search query of “Viagra” produced, on average, one new tweet spam every 30 seconds. And that's only for one drug!

We present a classification scheme using decision strategy and data mining techniques to specifically identify pharmaceutical spam taking into account the unbalanced nature of the data set. Then a broader decision set is used, classifying a post as strongly identified as pharmaceutical spam, yes, maybe, and no. We compare these techniques using older corpus of microblog data and currently collected data. This paper is organized as follows. Section II explains Twitter usage and facets of Twitter that spammers take advantage of to spread their

messages. Section III gives background on prior work with Twitter spam. Section IV describes the data collection, organization and preprocessing steps taken. In section V, we discuss the data mining techniques used in our experiments. Results and conclusions are presented in section VI.

## 2 TWITTER

Twitter is a real time information network that currently has over 175 million registered users creating approximately 95 million tweets per day [5]. They provide free access to their web site and tools to post a quick thought in 140 characters or less. These brief little messages, averaging 10 words per post, are called *tweets*. They are constrained in length by the SMS protocol. Tweets can be sent and received from mobile communication devices as well as web browsers running on any platform. Figure 1 shows an example of a user tweet and a typical pharmaceutical spam tweet.



(a) Normal user tweet.



(b) Pharmaceutical spam tweet.

Figure 1. Example tweets.

A popular spamming technique in any venue is the use of malformed words for the product so that filters are fooled, if only temporarily. Intentionally misspelled words (Vaigra), adding symbols in words (Vi@gra), and diluting "spam-iness" of a message by adding "good" words along with spam words, are just a few techniques used in both microblogs and regular email. Twitter has several features that are unique to the site that spammers exploit. These features offer new opportunities not present in email systems.

When it comes to Twitter, the major concept to grasp is the pathway of communication. In the email world, a spammer must find a way to spread thousands or more messages from a single or limited point of origin. The message goes from the spammer, or a bot-infected machine, to user inboxes across the Internet. Thus, spam comes to the user. In the social networking model, users come to the spam. Following are some of the more popular and nefarious techniques used to entrap victims.

- *Hashtags* –A hashtag is a symbol, #, used before relevant keywords in a tweet. The purpose is to categorize tweets related to the word so that they will be retrieved better in a Twitter search. Once a hashtag becomes very popular it tends to become a trending topic. Spammers attach their message or advertisement to one of the trending topics via hashtags. This is an attempt to lure users into reading their posts, led there under the assumption that the tweet actually relates to the popular topic marked in the message. Figure 2 shows an example of a spam tweet that uses the hashtag **#iranelection** to market videos.



Figure 2. Spam tweet using a hashtag.

- *Mass Following*- A major thrust of Twitter is having a conversation in some shared social context. Participants can follow other users or they can follow a topic of interest. Twitter allows you to follow other people's tweets and for others to follow you. However, the social network created by these links is not bidirectional. Someone may follow you without you following them back, unlike making friends on Facebook. Twitter spammers target the network by following other users and attempting to get them to follow back. They create fake profiles and start following as many people as is feasible without being identified as a spammer. They keep spam messages as their recent tweets and use provocative icons such that when the followed users check the followers they see the spam content.
- *Tweetjacking* - Tweets can be reused through a process called *retweeting*. This is a convenient way to share information without retyping it. It is accomplished by inserting the tag:

RT @username

The username indicates the original author of the tweet. Similar to this feature are replies and mentions. Placing @username at the beginning of a tweet followed by a message is called a reply. Placing it anywhere else in a tweet is called a mention. Twitter collects these and sends them to the user so they know what is being discussed. Spammers take advantage of these to get attention without having to be followed. It also tends to give the spammer credibility, albeit misleadingly. In tweetjacking, spammers reply to or retweet the message replacing originals hyperlinks in a message with hyperlinks that lead to porn sites, phishing, or malware traps.

- *Helpful tips* - A favorite technique of spammers is to post tips on health, beauty, weight loss etc which are actually links to fraud websites. Self promotional posts invite people to join communities, organizations and get free dining offers and other gifts. The user only needs to follow the link.

Not all promotional messages or hyperlinks are spam. Many of these contain genuine information but sites like Twitter and Facebook are growing so fast, it is difficult to determine which message or post is a spam and which is not without actually following the link.

Because the point of spam advertising and phishing is to sell a product or lure people to a site for nefarious purposes, a URL is a key element of the tweet. Indeed, it is common to include URLs in legitimate tweets as well. This causes problems where the URL contains a significant number of characters since tweets are limited to 140. Several URL “shortening” services are available for free online [6, 7]. An obfuscated URL makes it impossible for a user to determine the authenticity of the site before clicking on it.

Figure 3 shows two tweets using bit.ly to obfuscate a URL. Note that these ads for Viagra are posted by the same user (Abriannella) and are luring potential customers to two different web sites yet the posts are made only six minutes apart.



<http://bit.ly/hlmzuk>  
expands to  
<http://www.mahalo.com/buy-cheap-viagra/>  
(a)



<http://bit.ly/fRKhY8>  
expands to  
<http://meds-store.org/>  
(b)

Figure 3. URL shortening.

The URL in Figure 3(a) was advertised in a tweet by user Dominiqueela on February 12, 2011. An attempt to access that user several days later reveals the account has been suspended by Twitter. Yet on February 26, user Abriannella is up and running, advertising for the same site. Thus, Twitter works diligently to fight these

fraudulent activities, but clearly, the battle is far from over.

### 3 RELATED WORK

Spam filtering and identification for email is certainly not new, but research into microblogs as a target for spam is still in its relative infancy.

There are several recent works that seek to identify spammers rather than tweets containing spam. Benevenuto, et al., [8] use machine learning techniques to identify spammers based on number of followers, number of followees, and other social interactions, such as age of the account, average tweets per day and so forth. They also conduct some preliminary work with SVMs to identify spammy tweets based on spam words and the presence of a hash tag.

Yard, et al., [9] study a small collection of spammers and their behavior. They study the number of tweets and number of followers. The structure of their small social network, how the followers are connected, is a main theme of this research.

Wang [10] also uses social graphs, defining reputation as the ratio of friends to followers. Spammers are identified by sending a larger than normal number of duplicate or remarkably similar tweets. Behavioral content features studied are trending topics, replies, and mentions. They also search tweets for the presence of a URL. A number of classification algorithms are compared with the naïve Bayesian classifier outperforming k-nearest neighbor, neural networks, and support vector machines. They conclude that analysis based on content features proves tricky and will continue to become more difficult as legitimate businesses start leveraging the popularity of Twitter to promote themselves. Moh and Murmann [11] introduce trust metrics in addition to adding a finer-grained list of attributes identifying spammers

Abuse of automated agents using the Twitter API is discussed in Mowbray [12]. One technique allows spammers to generate followees automatically in anticipation that in some cases the social link will be reciprocated.

Blocky [13] is a website that provides a free service to block spammers, but their methodology is a simple human voting system on whether a user is sending spam or not. If the user is identified as a spammer, he/she is added to a blacklist. This is a very effective technique in the small, but in the large, it is impractical. No human-based system can hope to combat a problem that encompasses 95 million tweets per day.

In Liszka et al., [14] apply text mining techniques to preprocess Twitter data to be used by data mining tools to generate classifiers for spam tweet detection. A simple method was applied for labeling spam training tweets based on 65 pharmaceutical discriminating words selected

manually. Results show that the J48 decision tree classifier may be used as an effective tool for detecting pharmaceutical spam.

#### 4 PROBLEM FORMULATION

In this research, we consider the following problem. Given a collection of tweets from the Twitter data stream, how do we derive a classifier for identifying new tweets as pharmaceutical spam? In order to apply data mining tools to generate a classifier, we need to determine a list of features to represent tweets and assign a pharmaceutical spam label to each tweet.

Two sources are used to supply the data for our experiments. First, we downloaded a publicly available dataset for our sample space, provided for research purposes under Creative Commons license from Choudhury [15]. This data set contains more than 10.5 million tweets collected from over 200,000 users in the time period from 2006 through 2009. Second, we used the Twitter API to download another set of 0.5 million tweets during the time period of four weeks in the months of November 2010 - December 2010. Each set was kept separate for testing and training to see if there is a change in the nature of tweets.

Microblogs, as the name suggests, are about short length messages. It is common for people to type abbreviated form of words, for example, *how* becomes *hw*, *are* becomes *r* and *fine* becomes *f9*. This is common lingo in chats and e-mail but is more prominent in microblogs due to the message length limitation. Tweets are difficult to process for the following reasons:

- Words cannot be found in a dictionary yet they convey meaning to the reader. For example, LOL is short for “laugh out loud”.
- Emoticons are used to express feelings. One can say “I am happy” or alternately use ☺, both mean the same but with emoticons conveying it in fewer characters.
- Many tweets are not written in English.
- *Stop words* are words such as *the*, *have*, *then*, *is*, *was*, and *being*. These are necessary for sentence formation but really do not convey any message as an individual word unlike *appreciate*, *thoughtful*, and *significant*. These are adjectives and verbs which are more meaningful than propositions.
- Use of symbols and control characters makes data more noisy and difficult to process.

We clean and preprocess the tweets with the following steps:

- We remove stop words because they do not convey any useful information in the context of pharmaceutical spam. We created our own lists based on reading through many tweets. This step reduces the size of data and removes noisy words.

- Emoticons are identified and removed.
- Hyperlinks are identified and replaced with the token *URL*. This reduces the length of message and makes it easier to interpret. We’ve kept the originals for further research.
- We then tokenize each message.
- Control characters and punctuation symbols are removed.

Finally, we tackle the problem of generating an effective work list to classify tweets. Generating a good word list is a challenging feat if the end result is to be effective. We started with a list of common terms related to this type of spam derived from several sources [11, 12, 13]. Then we added words from our email spam and finally, we manually analyzed over 5000 tweets, looking for the common terms used to peddle prescription drugs.

From observation, we decided to work with two sets of words that we label primary and secondary. We identify 86 primary words including, for example, {viagra, xanax, pharma, prescription, pill, medication}. Additionally, there are 45 secondary words identified including {generic, refill, shipping, wholesale}. This is the basis for our decision logic. We hypothesize that while we can get good results working with just primary words, additional intelligence including the presence of one or more secondary words, and in particular with a URL, will help us differentiate true pharmaceutical spam from tweets that are only relaying a joke. For example:

Non-pharma: I can mingle with the stars and have a party on mars i am a prisoner locked up behind **xanax** bars

Pharma: Order Online Buy low cost **Xanax** (Alprazolam) medication You can buy **Xanax** online at very... Buy Alprazolam -->>http://bit.ly/grCsej

#### 5 EXPERIMENTAL RESULTS

Our training sets were created by encoding each tweet with the set of 131 keywords mentioned in the previous section. A training tweet is labeled as spam if it contains at least one keyword from the set. We used J48 and Random Tree (RT) learning algorithms from Weka to generate spam classifiers. In addition, a decision logic introduced in Liszka et al. [14] is used to improve the performance of J48 and RT classifiers. Two training sets were randomly selected from the collected tweets described in Section IV. Since pharmaceutical spam tweets are in a rare class, under-sampling of non-spam tweets is used to create balanced training sets: one consists of 48,133 tweets from the collection of 10.5 millions, and the other consists of 1,079 tweets from the 0.5 million collection. Two independent testing sets of 4,956 and 4,973 tweets are randomly selected from the two collections, and they are labeled manually.

Performance evaluation of the classifiers are shown in Table 1 and 2 where D1 refers to training set created from the 10.5 millions collection and D2 refers to the one created from 0.5 million collection. DL(J48) denotes the classification based Decision Logic plus J48, and DL(RT) denotes the combination of Decision Logic plus Random Tree. The basic idea of our decision logic is to include the frequency of occurrences of keywords in determining if a tweet is spam or not. The detailed description can be found in [14]. The measures are based on standard definitions from the literature as mentioned in Liszka et al. [14]. Table 1 shows the results evaluated by using the manually labeled testing set of 4,956 tweets drawn from the 10.5 million tweets, and Table 2 shows results using 4,973 tweets drawn from the 0.5 million tweets.

Table 1. Performance of pharma spam classifiers evaluated by 4,956 tweets.

Measures	J48		RT		DL(J48) and DL(RT)	
	D1	D2	D1	D2	D1	D2
accuracy	0.96	0.96	0.87	0.89	0.97	0.97
recall	0.42	0.22	0.93	0.84	0.65	0.65
precision	0.39	0.34	0.22	0.23	0.55	0.55
F-measure = $\frac{2*TP}{2*TP+FP+FN}$	0.40	0.27	0.34	0.36	0.59	0.59
sensitivity = tp	0.42	0.22	0.93	0.84	0.65	0.65
specificity = 1-fp	0.98	0.98	0.87	0.89	0.98	0.98
sensitivity*specificity	0.43	0.22	0.81	0.75	0.63	0.63
# of nodes in resulting tree	69	19	363	119	-	-
# of primary spam words selected	31	8	49	29	-	-
# of secondary spam words selected	3	1	23	14		

Table 2. Performance of pharma spam classifiers evaluated by 4,973 tweets.

Measures	J48		RT		DL(J48) and DL(RT)	
	D1	D2	D1	D2	D1	D2
accuracy	1.00	0.99	0.99	0.99	1.00	1.00
recall	0.23	0.15	0.85	0.85	0.69	0.69
precision	0.19	0.11	0.16	0.19	0.39	0.39
F-measure = $\frac{2*TP}{2*TP+FP+FN}$	0.21	0.13	0.27	0.31	0.50	0.50
sensitivity = tp	0.23	0.15	0.85	0.85	0.69	0.69
specificity = 1-fp	1.00	1.00	0.99	0.99	1.00	1.00
sensitivity*specificity	0.23	0.15	0.84	0.84	0.69	0.69

# of nodes in resulting tree	69	19	363	119	-	-
# of primary spam words selected	31	8	49	29	-	-
# of secondary spam words selected	3	1	23	14		

Table 3. Performance of pharma spam classifiers evaluated by 4,956 tweets using primary keywords only as shown in [14].

Measures	J48		RT		DL(J48) and DL(RT)	
	D1	D2	D1	D2	D1	D2
accuracy	0.97	0.97	0.97	0.97	0.97	0.97
recall	0.40	0.21	0.40	0.40	0.65	0.65
precision	0.62	0.54	0.61	0.62	0.55	0.55
F-measure = $\frac{2*TP}{2*TP+FP+FN}$	0.49	0.31	0.49	0.49	0.59	0.59
sensitivity = tp	0.40	0.21	0.40	0.40	0.65	0.65
specificity = 1-fp	0.99	0.99	0.99	0.99	0.98	0.98
sensitivity*specificity	0.40	0.21	0.40	0.40	0.63	0.63
# of nodes in resulting tree	57	17	97	73	-	-
# of primary spam words selected	28	8	48	36	-	-
# of secondary spam words selected	-	-	-	-	-	-

## 6 DISCUSSION AND CONCLUSIONS

Since spam tweet detection is a problem of learning from imbalanced classes, it is well-known that accuracy of classification is not a good indicator of evaluating a classifier's performance. Many agree that recall, precision, and F-measures are better indicators. From Table 1 and Table 2, it is clear that our decision logic provides improvement over J48 and RT. In our experiments, we have observed that the decision trees generated by both J48 and RT are degraded into linear lists. There were no combinations of more than one keyword in the trees leading into a leaf node of positive class, i.e., spam tweets. Further studies are required to determine if it is caused by the attribute-oriented greedy strategy used by typical decision tree learning algorithms. One of our future works is to observe if this pattern also appears in rule-based learning algorithms, which are local learners based on attribute-value pairs.

In our experiments, we have evaluated the classifiers by cross validation using testing sets collected one year apart. Table 1 is the results where testing set is collected at the same time as the D1 training set, and Table 2 shows the results where testing set is collected at the same time

as the D2 training set. It is interesting to note that the patterns captured by J48 fluctuate more with respect to time comparing to the Random Tree (RT) patterns. Our experiments show that the RT patterns seem to have a more stable performance with respect to time. In addition, they have a much higher overall recall rate; however, they have lower precision rates. It also indicates that J48 tends to under fit the data in our studies.

The use of secondary keywords for spam tweets detection seems to have a negative impact for J48 and RT classifiers. Table 3 shows the results taken from previous work presented in [14], where only primary keywords were used. The complexity of trees increases in both J48 and RT learning when comparing Table 1 to Table 3.

It seems that a reliable tool for spam detection in general, and pharmaceutical spam in particular, is crucial in applying Twitter's search feature. Using Random Tree + Decision Logic is a good tool to study pharmaceutical spam, because RT has high recall rate, which means it can be used to select training tweets from a data set without losing too many spam tweets. However, RT suffers from low precision, which can be remedied by the proposed Decision Logic in pharmaceutical spam tweet prediction.

## 7 REFERENCES

- [1] Twitter, <http://twitter.com>, last accessed February 2011.
- [2] Egyptians Were Unplugged, and Uncowed, The New York Times, February 2011.  
<http://www.nytimes.com/2011/02/21/business/media/21link.html?src=busln>.
- [3] Twitter, The New York Times, April 15, 2010.  
<http://topics.nytimes.com/top/news/business/companies/twitter/index.html?inline=nyt-org>.
- [4] Pharmacy Spam: Pharmaceutical Websites Fall into Two Distinct Operations, March 1, 2010.  
<http://www.symantec.com/connect/blogs/pharmacy-spam-pharmaceutical-websites-fall-two-distinct-operations>.
- [5] About Twitter, <http://twitter.com/about>, last accessed February 26, 2011.
- [6] bit.ly, <http://bit.ly/>, last accessed February 2011.
- [7] TinyURL, <http://tinyurl.com>, last accessed July 2010.
- [8] Benevenuto, F., Magno, G., Rodrigues, T., and Almeida, V., "Detecting Spammers on Twitter," Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, CEAS July 2010.
- [9] Yardi, S., Romerao, D., Schoenebeck, and G., Boyd, D., "Detecting Spam in a Twitter Network", First Monday, 15(1), 2010.
- [10] Wang, A., "Don't follow me: Twitter spam detection," Proceedings of 5th International Conference on Security and Cryptography (SECRYPT), July 2010, Athens, Greece.
- [11] Moh, T. S., and Murmann, A., "Can you judge a man by his friends? - Enhancing spammer detection on the Twitter microblogging platform using friends and followers," Information Systems, Technology and Management, Communications in Computer and Information Science, 2010, Volume 54, no. 4, 210-220, DOI: 10.1007/978-3-642-12035-0\_21.
- [12] Mowbray, M., "The twittering machine," Journal of Applied Statistics, 17(2):211-217.
- [13] Blocky, <http://blocky.elliottkember.com/>, last access July 2010.
- [14] Liszka, K., Chan, C., Shekar, C., and Wakade, S. (2010). "Mining pharmaceutical spam from Twitter," International Conference on Intelligent Systems and Applications (ISDA 2010), November 2010, pp. 813-817.
- [15] Choudhury, M. D., Lin, Y.-R., Sundaram, H., Candan, K. S., Xie, L. and Kelliher, A., "How Does the Sampling Strategy Impact the Discovery of Information Diffusion in Social Media?," Proc. of the 4th Int'l AAI Conference on Weblogs and Social Media, George Washington University, Washington, DC, May 23-26, 2010.

# Designing Information Security Policy for Establishing Trust within Health Care Environments

Sarah M. North, Ed. D.  
Information Technology Department  
Computing and Software Engineering  
Southern Polytechnic State University  
Marietta, GA 30060 sarah@spsu.edu

Max M. North, Ph.D.  
Management Information Systems  
Engineering Technology and Management  
Southern Polytechnic State University  
Marietta, GA 30060 max@spsu.edu

## ABSTRACT

While there are several laws and policies concerning the security of patient information in hospital environments, there is not much exploration of design and guidelines for information security to deal with this important matter. The main objective of this paper is to explore a possible design for an information security policy that protects patient information and provides a network system policy to identify the level of authorized access to patients' health records. Furthermore, this design will discuss the relevant laws and statutes applicable to the medical organization and how they are able to handle patients' records. In addition, the policy will elaborate on the high level cost involved with this design, with general ideas for an implementation plan that will complement a typical hospital Information Security Policy.

## Keywords

Information Security Policy, Health Care, Computer Security Awareness, Computer Ethics Awareness, Management Information Systems

## 1. INTRODUCTION

Designing information technology security policy for hospitals involves assessing network and patient information, risk of unauthorized disclosure, and modification or changes of personal information. Since personal information is kept in both electronic and paper format, which can be shared in a number of ways such as an e-mail or verbal conversation, it often can be vital for the progress and quality of hospital care [1]. The strategy aims to provide a robust design with flexible security policy that is trusted by maximizing the accuracy and validity of patient information [9]. Furthermore, the design acknowledges the emerging challenges of ACME hospital patient information, validating numbers on all patient records, authorization level of access to patient data, and identifying the existing guidelines on management of data. In addition, handling information quality assurances requirements is direct and provides a basic, fundamental statement on responsibilities on all data collection, management and mentoring activities within the principles and "trust" code.

An information technology security policy is the most important and critical element of a hospital security program. It identifies the rules and procedures which all personnel accessing the computer system must follow in order to ensure the comprehensive confidentiality, integrity, and availability of patient data kept private based on the policy guidelines. In

general, a Hospital Information Technology Security Policy includes the following actions [10]:

- Communicate clear, concise, and realistic information;
- Define the scope, responsibilities, and applicability of the policy;
- Identify the level of authorization access for doctors, nurses, and management;
- Provide sufficient guidelines for design and develop new procedures;
- Create a balance to enhance productivity and patient record protection; and
- Handle patient data and identify how incidents will be handled.

## Access of Patient Information

Computer technology has definitely raised the standards of health care by creating an information pathway that causes patient information to be readily available in order to procure the best care. Even though information is passed so easily through electronic transference, many organizations have overlooked a very vital issue for discussion and action: true security. Not only do doctors and nurses have access to patient information, but many non-caregivers, including researchers, medical transcribers, insurance companies, and their employees, must view and utilize this information as well. This information has no problem traveling among many sets of eyes to view, scan, fax, print, or send patient information wherever they choose [1, 3, 6].

## Planning Information Security

Security is essential to any public business organization, but is even more important in hospitals, which hold the key to so much vital and sensitive information, such as personal information, medical records, expensive medical equipment and an abundance of drug and distribution information. Hospitals desire to create a calm, welcoming environment, which often leaves very tight security measures lacking. It is imperative to patient and employee privacy that hospitals grow and expand their thinking to create a well balanced environment that includes the most effective agencies available.

## Types of Data and Information

Hospitals are responsible for a large variety of personal information on doctors, employees, volunteers, donors, vendors, partners, and, of course, patients. The data that requires protection includes names, social security numbers, diagnoses, treatments, health insurance plans, and much more. In order to

maintain the integrity of this information, a legitimate security plan, and policy/procedure must be put into action. The information security management system should be based on three levels of security: confidentiality, integrity, and availability [4, 7, 11]. Each level of security is unique to the overall protection of pertinent sensitive data.

### Cost and Implementation

The initial backlash from HIPAA is sometimes overshadowed by the cost of compliance for many health organizations. Although steep, the overall cost for a complete makeover is manageable with a concentrated effort throughout the organization. It is not an effort to be completed within a short amount of time; it requires much organization, planning, and will. In order for such an implementation to take place, the organization must be fully aware of the assessed goals and all its financial allowances [5].

## 2. THE DESIGN ASPECTS OF INFORMATION SECURITY POLICY

**Information Security Policy for ACME Hospital Scope.** The ACME hospital (ACME is a fictitious name used here for illustration) policy applies to the security of the electronic data managed and owned by the ACME hospital and all trusted employees, including the staff and contractors working on behalf of the trust. It applies to all information technology staff and their related activities, and includes the following:

- Handling of information that govern management by specifying the principles of data collections of the “trust” information processes;
- Confidentiality and data protection assurance which confirm trust policies;
- Clinical information assurances allocate the responsibility for update and review of data quality; mandate the use of validated unique identifiers on all patient records;
- Compliance with the department of Health Data Standards and all associated legal obligations, including secondary use assurances;
- Establishment of the provision of Standing Financial Instructions in regard to information technology security in financial systems, including network protection.

**Aims.** The security policy aims to ensure that the strategy is intended to achieve the following:

- Define the ACME Hospital organizational structures that manage and monitor employees’ activities and to improve the quality of patient information care;
- Outline main responsibility and accountability of all employees with appropriate process evaluation and feedback;
- Control adequate producers and management practices over the services that doctors and nurses provide to patients;
- Mandate all employees to use relevant IT and legal requirements;
- Confirm the trust policy and procedures related to data quality and information collection from patients.

- Provide awareness training of information security and quality issues with clinical and nursing staff in order to meet clinical needs;
- Maintain the confidentiality, integrity and availability to all the employees by authorizing access to information that is associated assets when required;
- Detect and resolve any problem that breaches the security policy; and
- Verify that the security policy complies with Freedom of Information legislation and public visibility of information quality management.

**Principles.** The principles of information quality policy move toward the delivery and support of patient care, including internal and external hospital boundaries and their responsibilities for everyone involved. In order to ensure the information standards and policy are thorough and effective, we need to overview these principles:

- Effective delivery of patient-centered services at the heart of the ACME hospital care record, with accuracy and accountability;
- Obtaining information and delivering it to the right person, at the right time, and at the right level of detail to care for and inspire confidence in patients;
- Efficient service, in a systematic way, for delivery, performance management, and planning of future service needed for each patient [9].

**Legal Compliance.** ACME hospital trust is bound by the following legislation that affects the management and control of information needed to care for patients’ rights:

- a) Freedom of Information Act 2000
- b) Regulation of Investigatory Powers Act 2000
- c) Data Protection Act 1998 - Chapter 29 (DPA)
- d) Human Right of Act 1998
- e) Computer Misuse Act 1990
- f) Access to Health Record 1990
- g) Copyright, Design and Patents Act 1988
- h) Health Insurance Portability and Accountability Act 1996 (HIPAA)
- i) Health and Safety at Work Using Computer Technology Act 1974 (Cambridge University Hospitals NHS Foundation Trust, [2])

### Security Principles

Information technologies at ACME hospital are essential tools for the delivery of safe, high-quality patient care and efficient organization of doctors and nurses. Therefore, it is imperative to have a stable information security system that sends patient information across the network. The security level is created with the following policy:

- **Confidentiality:** Authorized users only have access to a certain level of patient information; all other access will be denied.
- **Integrity:** Safeguard the process of information to make sure that data is accrued, to assure the completeness of needed data, and to verify the authentication of the

information and the system that is operating according to their specifications;

- **Availability:** Provide authorized user access to information needed with appropriate time given to care for patient.

## Security Levels

ACME Hospital is led by security managers whose responsibilities for implementation and enforcement of the Information Security Policy are as follows:

- Monitor, report, and evaluate the state of all information security levels within the organization;
- Plan, update, and manage the improvement of security policy throughout the organization;
- Develop and reinforce detailed producers and standards to maintain security;
- Identify and ensure all employees' responsibilities and accountability for the information's security throughout the computer system network;
- Monitor for actual information breach and internal and external access to network;
- Provide advisory security team on information security services to provide feedback and govern the process;
- Ensure that all access to information must be unique ID and password with secure authorization, following the method of approval by the system manager;
- Ensure that no sharing or generic accounts can be used; these can pose a significant security risk for unauthorized access to the network system.

**Security Incidents.** All employees must follow the ACME hospital information security policy and procedures; any breach of this policy could result in disciplinary action, which may result in dismissal from employment. The IT security manager is responsible for comprehensive investigation of all incidents and provides evidence to report to upper level management.

**Chief Information Officer (CIO).** The CIO is responsible for overseeing the security policy and standards that are implemented to ensure that

- development of any new system is performed following the ACME hospital security policy and standards;
- when new systems are purchased or installed, these systems pass through assessment and technical review before registration to follow the terms of the Data Protection Act.

**System Manger.** System managers are individual managers who are responsible for all local producers who support the Information Security Policy. Their responsibilities are as follows:

- Provide training for all the staff and identify their responsibilities
- Ensure staff may only access the system that they are have been authorized
- Ensure that all the software and database system owners are registered as policy requires and that all the hardware equipment is risk-assessed before use;
- Make sure that downloaded software, screen saver programs, games, etc are not installed;

- Keep viruses and other malicious software out of the system.

## Physical Security Control

The two types of physical security controls that used in organizations security policies that would be applicable to hospital security controls are reliable power, using newer technology such as a UPS (*uninterruptable power supply*), and *Human guards* around a hospital or any building with sensitive information. The physical security controls are measures taken to safeguard an information system from attacks against its confidentiality, integrity, and availability (giac.org, 2010). Having physical security controls is very important not only to a hospital, but to any business. In order to defend employees and information against unwanted outside users, it is imperative to have these physical controls in place.

## Risk Management

All systems and IT infrastructure must be regularly checked for risk assessment, the results of which should be reviewed with the system manager. The review should include the following:

- Identify the potential threats to the *confidentiality, integrity or availability* of the system or patients' data and records;
- Identify any *impact* if an attack occurs and have a formal procedure for recovery in order to implement counter action;
- Identify all assets, which consist of all computer systems and equipment;
- Identify cost-effective countermeasures to reduce damage and impact.

## Health Records

ACME hospital employees are committed to protect patient information, including—but not limited to—patients' health records. Based on the Access to Health Records Process, which is based on the DPA (Data Protection Act) 1998 (Live patient request) and the Access to Health Records Act 1990 (Deceased patient request), all the records must be used in the "trust" and control for the use and/or disposed which must be kept in the archives of the hospital.

## Audit

The information security policy implementation must be subject to periodic review, assessment, and evaluation, both internally and externally, using qualified auditors to proceed with the given recommendations by sharing all the feedback with the upper level of management.

**Access Protection.** All information systems, manual files, and computer system networks must contain information on how to access the system and must follow the policy and guidelines with appropriate authorized access. It is the responsibility of the level manager to ensure that all new staff properly follow proper standards and IT security policy. It is imperative to provide appropriate *training* to all staff before granting authorized access to the computer system.

**Training.** The employees are responsible for participating in hands-on training in order to be authorized to access the system.

The training is mandatory for all staff to prepare them for security duties.

**Disaster Recovery.** The entire recovery plan must be identified in detail and must be performed if the system fails in any way. These plans will include:

- Recovery procedures for emergency and immediate actions in the event of an incident;
- Full documentation and system configuration for computers;
- Clear procedures to return to normal full services after the incident occurs.

**Cost.** The cost related to information security policy is often difficult to measure, because it involves theft of property information or financial fraud in addition to system attacks. The cost may involve malicious code, unauthorized access, and cyber attacks, which are all problems continuing to reflect dramatic growth each year [8]. Most recently, many organizations, including hospitals, are using Cost-Benefit Analysis (CBA) techniques to come up with better estimations for establishing and managing security policy by using these applications, along with support of National Institutes of Health (NIH). These useful methods and guidelines, documented for preparing security policy for hospitals using CBAs, are required by the U.S. Federal government to support the IT environment and their management decisions [8].

### 3. ACKNOWLEDGMENT

This effort was supported by the Department of Defense (DOD)/National Security Agency (NSA) Grant. The content of this work does not reflect the position or policy of the DOD/NSA and no official endorsement should be inferred. Southern Polytechnic State University has been designated a Center of Excellence in Information Security Assurance (CAE/ISA) by the Committee on

National Security Systems (CNSS) and the National Security Agency (NSA).

### 4. REFERENCES

- [1] Ahima.Org. (2002). Laws and Regulations Governing the Disclosure of Health Information. Retrieved August 14, 2010, from <http://library.ahima.org/xpedio/groups/public/documents/ahima/bok1>
- [2] Cambridge University Hospitals NHS Foundation Trust, (2008). *Policy - Information Technology* . Retrieved from the web on August 13, 2010, [http://www.cuh.org.uk/resources/pdf/cuh/profile/publications/selected\\_policies/](http://www.cuh.org.uk/resources/pdf/cuh/profile/publications/selected_policies/)
- [3] Ferreira, A. (2007). *Who Should Access Electronic Patient Records*. Retrieved August 14, 2010, From <http://www.dcc.fc.up.pt/~lfa/healthinf08-ac.pdf>
- [4] GIAC, . (2010). *Security Control Types and Operational Security*. Retrieved from the web on August 6th, 2010, from <http://giac.org/>
- [5] Lorie.Cranor.Org. (2005). Retrieved August 14, 2010, from <http://lorrie.cranor.org/courses/fa05/mpimenterichaa.pdf>
- [6] Nationalacademics.org. (1997). HIPAA. Retrieved August14, 2010, from <http://nationalacademics.org/>
- [7] Pentadyne,. (2010). Flywheel Technologies Retrieved from the web on August 15<sup>th</sup>, 2010, from [www.pentadyne.com](http://www.pentadyne.com)
- [8] Mercuri , Rebecca. (2003). Security watch analyzing security costs. *Communication of ACM*, 46(6), 15-18.
- [9] NHS Choices, (2010). Cambridge University Hospitals, *Information security*. Retrieved from the web [http://www.cuh.org.uk/addenbrookes/contact/contact\\_add.html](http://www.cuh.org.uk/addenbrookes/contact/contact_add.html)
- [10] SANS Institute, (2002). *The Basics of an IT Security Policy*. Retrieved from the web on August 13, 2010, [http://www.giac.org/certified\\_professionals/practicals/gsec/1863](http://www.giac.org/certified_professionals/practicals/gsec/1863)
- [11] University of Miami. (2008). *Confidentiality, Integrity, and Availability*. Retrieved August 14, 2010, from <http://it.med.miami.edu/x904.xml>

# Using Ciphertext Policy Attribute Based Encryption for Verifiable Secret Sharing

Nishant Doshi<sup>1</sup>, Devesh Jinwala<sup>2</sup>

<sup>1,2</sup> Computer Engineering Department, S V National Institute of Technology, Surat, India  
{<sup>1</sup>doshinikki2004@gmail.com, <sup>2</sup>dcjinwala@acm.org}

**Abstract** - Threshold secret sharing schemes are used to divide a given secret by a dealer in parts such that no less than the threshold number of shareholders can reconstruct the secret. However, these schemes are susceptible to the malicious behavior of a shareholder or a dealer. To prevent such attacks, it is necessary to make a provision for verification of the integrity of the shares distributed by the dealer. Such verification would ensure fair reconstruction of the secret. In this paper, we present a novel approach for verifiable secret sharing wherein the dealer and the shareholders are not assumed to be honest. Our proposed scheme uses attribute based encryption (ABE) to provide verifiability and for the semantically correct reconstruction of the secret. We call the new protocol as AB-VSS (Attribute Based Verifiable Secret Sharing).

**Keywords:** Attribute, Attribute based cryptography, Network Security, Verifiable secret sharing.

## 1 Introduction

In modern cryptography, the security of a cipher is heavily dependent on the secrecy of the cryptographic key used by the cipher. Hence, the key is required to be carefully guarded - needs to be stored *super-securely*. Obviously, one of the most secure ways to do so is to keep the key in a single *well-guarded* location. However, once the "well-guarded" location is compromised, the system fails completely. Hence, the other extreme is to distribute the secret at multiple locations. However, such a de-centralized approach increases the vulnerability to failure and makes the task of the potential attackers a bit easier. Additionally, in real world, the stakeholders and the key distributor may not trust each other. Secret sharing then, appears to be a good solution to deal with such problems. In secret sharing, a secret is distributed and shared across a number of shareholders with the caveat that, no less a designated number of shareholders would be able to reconstruct the secret. Secret sharing as such is a bit of misnomer. In secret sharing, the shares of a secret are distributed among a set of participants, and not the entire secret, to deal with the mutual mistrust. Hence, the scheme is better termed as *threshold* secret sharing.

Adi Shamir [1] and G. Blakley [2] in 1979 independently introduced the concept of the threshold secret sharing. As per these proposals, a dealer  $D$  who holds a secret  $s$  would distribute it amongst  $n$  shareholders in such a way that a quorum of less than  $t$  shareholders cannot regenerate the secret. That is, any combination of at least  $t$  shareholders is

required to regenerate the same secret correctly. An interesting *real-world* example to illustrate this scenario was given in the *Time Magazine* as per which, the erstwhile USSR used a *two-out-of-three* access control mechanism to control their nuclear weapons in the early 1980s. The three parties, viz. the President, the Defense Minister and the Defense Ministry, were involved to execute this scheme.

Shamir's threshold secret sharing scheme [1] has been extensively studied in the literature. The Shamir's threshold secret sharing scheme is *information theoretic secure* but it does not provide any security against cheating; as it assumes that the dealer and shareholders are honest. However, in real world one may encounter the dealers and the shareholders in an otherwise. A misbehaving dealer can distribute inconsistent shares to the participants or misbehaving shareholders can submit fake shares, during reconstruction. To prevent such malicious behavior of cheaters, we need a *Verifiable Secret Sharing*(VSS) scheme. The VSS was first proposed in 1985 by Benny Chor et al [3]. In their scheme, the validity of shares distributed by a dealer is verified by shareholders without being revealed any information about the secret. The initial VSSs were *interactive* verifiable secret sharing schemes that it required interaction amongst the dealer and the shareholders to verify the validity of shares [4]. This scheme used homomorphism and probability encryption function. However, as we observe this scheme only verifies the share provided by dealer to shareholders and does not verify the shares at secret reconstruction time. The interaction required itself imposes enormous amount of extra overhead on the dealer, as a single dealer may have to deal with a large number of shareholders. Later, *non-interactive* verifiable secret sharing schemes were proposed to remove the extra overhead on the dealer [5][6][7].

The non-interactive VSS proposed by Paul Feldman in [5] relies on the share proving *its own* validity. The one proposed in [6] tries to verify the reconstructed secret by *maximally matching* the secret. This scheme works in the same way as [1] when a threshold numbers of parts are given to reconstruct secret. The scheme proposed in [7] suggests iterating the process of secret sharing  $m$  times with one secret as  $S$  and others as dummy secrets - so with each shareholder there are  $m$  shares. This approach increases storage requirements, communication and computation cost. The schemes in [8] [9] are based on the use of a hashing function. The flaws in these schemes are already discussed in [10].

Thus, as per our observations, these schemes assume that the dealer is honest and the shareholders accept their shares without any verification. The shareholders simply

cannot identify cheaters in the system. The existing approaches for verifiable secret sharing either verify the shares, distributed by a dealer or submitted by shareholders for secret reconstruction, or verify the reconstructed secret but not both.

In order to verify shares, a dealer either transfers some additional information like check vectors [11] or certificate vectors or it uses different encryption mechanisms. If the VSSs do not use the check vectors or certificate vectors, the security of such schemes depend on the intractability of a number theoretic problem in one way or another. If the scheme uses check vector or certificate vectors, then it increases an extra overhead on a dealer to compute and distribute that extra information among a large number of participants.

In this paper, we use and extend the verifiable secret sharing approach to not only verify the validity of shares distributed by a dealer but to verify the shares submitted by shareholders for secret reconstruction, and to verify the reconstructed secret. We use the notions of the Attribute Based Encryption to deal with the limitations of the existing schemes – at the same time offering *user verification, secret distribution and secret regeneration* using valid threshold secret parts.

In the scheme proposed in [12] the problem of cheater detection is discussed when there are  $t - 1$  cheaters in  $n=2t-1$  shareholders. However, this scheme is vulnerable to attacks. In the scheme proposed in [13], an Elliptic Curve Cryptography based approach is used for VSS. However, this scheme requires the dealer to hide the secret in a secure place. Hence, if the dealer is compromised the secret is also lost forever. As compared in our approach anyone having threshold shares can regenerate the secret. In the scheme proposed in [14], the Chinese Remainder Theorem (CRT) is used for devising secret sharing. However, a malicious shareholder can change its own share and submit a fake share and help reconstruct a fake secret – rendering the scheme useless.

Thus, as compared our scheme that employs the notions of the Attribute Based Encryption is free from all these attacks. In fact, as per our modest belief, ours is the first attempt at using the Attribute Based Encryption for the purpose of secret sharing.

## 1.1 Attribute Based Cryptography (ABC)

In this section, we review the state of the art in ABC and discuss the justification of the scheme used in our approach.

The ABC has actually been motivated from the Identity Based Encryption, which in turn was motivated by overcoming the limitations of the certificate management in the traditional Public Key Cryptography. The basic focus in ABC is on using some of the publicly known attributes of a user as his public key. In the traditional IBE systems, the identity of a user is specified using either the *name, the email ID, or the network address* – a string of characters. This makes it cumbersome to establish the necessary correlation

between a user's identity (in his private key) and the same associated in the ciphertext that he intends to decrypt. This is so, because even slight mismatch would render the match as a failure. Hence, in a variant of the traditional IBE, the identity is specified in the form of descriptive attributes. In the first of such scheme proposed as Fuzzy Identity Based Encryption (FIBE) in [15], a user with identity  $W$  could decrypt the ciphertext meant for a user with identity  $W'$ , if and only if  $|W - W'| > d$ , where  $d$  is some threshold value defined initially.

In [16], the authors propose more expressive ABE schemes in the form of two different systems viz. Key Policy Attribute Based Encryption (KP-ABE). In KP-ABE, a ciphertext is associated with a defined set of attributes and user's secret key is associated with a defined policy containing those attributes. Hence, the secret key could be used successfully only if the attribute access structure policy defined in the key matches with the attributes in the ciphertext. As compared, to the same the authors in [17] propose a fully functional Ciphertext Policy Attribute Based Encryption (CP-ABE) in which a user's secret key is associated with a defined set of attributes and the ciphertext is associated with a defined policy. One of the limitations of CP-ABE schemes is that the length of ciphertext is dependent on the number of attributes. That is, with  $s$  being the number of attributes involved in the policy, the ciphertext length is  $O(s^3)$ .

In [18], the authors propose another CP-ABE which had positive or negative attributes. But the decryption policies in this are limited to AND gate only. In [19][20], the authors first overcome the limitation due to the ciphertext length and propose a constant size ciphertext

Motivated from these efforts, in our scheme we use the approach proposed in [17]. For large number of shares we can use the concept of [19][20]. [21] had used time specific encryption in which they use time as attribute and time limit condition in policy so user can decrypt ciphertext if they have valid attributes at right time.

In VSS we can add time attribute if we want that the secret must be regenerated at a specific time only. After that time passes the secret becomes invalid. For example during war we can generate secret key to fire missile and add the specific time limit so after the war is over the secret to fire missile will become invalid itself. And if we want that at the time of secret generation or verification user must be at a particular location then we can consider an extra attribute 'location' in our proposed scheme. If same dealer has more than one set of  $n$  shareholders and if two shareholders from different sets will exchange their secret key which is based on hash value of share, then the given attack is not possible in our approach because if shareholder exchange key then the new key cannot pass the policy.

*Organization of the paper:* The rest of the paper is organized as follows. The second section will explain preliminaries which we are used throughout the paper. In the third section our proposed approach for verifiable secret sharing will be introduced and we will analyze it in the fourth section as well show a snapshot using the CPABE toolkit. The last section concludes the paper followed by the references.

## 2 Preliminaries

### 2.1 Notations

Most cryptographic protocols require randomness, for example generating random secret key. We use  $x \in_R A$  to represent the operation of selecting an element  $x$  randomly and uniformly from an element set  $A$ . We use  $\Phi$  to denote the NULL output. This paper deals with the computational security setting where security is defined based on the string length. For  $\ell \in \mathbb{N}$  where  $\mathbb{N}$  is the set of natural numbers,  $1^\ell$  denotes the strings of length  $\ell$ . If  $x$  is a string then  $|x|$  denotes its length, e.g.  $|1^\ell| = \ell$ .

### 2.2 Secret sharing

Divide some secret  $S$  into  $n$  parts  $S_1, S_2, \dots, S_n$  and distribute them among a set of  $n$  shareholders in such a way that for any threshold value  $t$ , the knowledge of any  $t$  or more  $S_i, 1 \leq i \leq n$  parts computes  $S$  easily but the knowledge of any  $t-1$  or fewer  $S_i$  parts leaves  $S$  completely undetermined. Such a scheme is called  $(t, n)$  threshold secret sharing scheme [1].

### 2.3 CP-ABE construction [7]

The CP-ABE toolkit consists of the following four algorithms as follows.

1. **Setup**: It will take implicit security parameter and output the public parameter PK and a master key MK.
2. **KeyGen**(MK, S) : The key generation algorithm run by CA, takes as input the master key of CA and the set of attributes for user, then generate the secret key SK.
3. **Encrypt** (PK, M, A): The encryption algorithm takes as input the message M, public parameter PK and access structure A over the universe of attributes. Generate the output CT such that only those users who had valid set of attributes which satisfy the access policy can only able to decrypt. Assume that the CT implicitly contains access structure A.
4. **Decrypt**(PK,CT,SK) : The decrypt algorithm run by user takes input the public parameter, the ciphertext CT contains access structure A and the secret key SK contain of user attribute set S. if S satisfies the access tree then algorithm decrypt the CT and gives M otherwise gives " $\Phi$ ".

## 3 Proposed approach for VSS

### 3.1 Share Generation and Distribution Phase

Input: Secret  $S \in \text{GF}(p)$  and a public hash function H

Output: Shares of the secret S,  $S_i$  Where  $i = 1, 2, 3, \dots, n$

1. Dealer D chooses a large prime  $p > \max(S, n)$
2. Then it selects  $t-1$  random independent coefficients,  $a_1, a_2, \dots, a_{t-1}$  where  $0 \leq a_i \leq p-1$
3. Select the random polynomial and set  $a_0 = S$   
 $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1}$ .

4. Compute the share of the secret for each shareholder and distribute the pair  $(i, S_i, SK_i)$  to each shareholder. We assume that every user  $i$  has only one attribute 'value =  $H(S_i)$ ' where  $S_i = f(i) 1 \leq i \leq n$ .  
 $SK_i = \text{KeyGen}(MK, A)$  where MK=master key of dealer  
 $A$ =attribute set for  $i^{\text{th}}$  user
5. Dealer makes policy for access tree structure as follow  
 policy=Encrypt(PK,M,T) where PK=public key of dealer,  
 M=Message and T=Tree structure  
 Here policy makes on condition  
 "value =  $H(S_1)$  OR value =  $H(S_2)$  OR value =  $H(S_3)$  OR ... OR value =  $H(S_n)$ "
6. Dealer broadcasts policy and  $t$  in public file.
7. Each  $i^{\text{th}}$  shareholder verifies their share by Decrypt (policy,  $SK_i$ ). If message M successfully decrypted then user accepts their share.
8. User  $i$  verify its  $SK_i$  anytime by sending  $S_i$  to dealer. Dealer compute  $SK_i$  based on  $H(S_i)$ . No required to store any information of share secret on dealer side other than hash function.
9. If all the shareholders find their shares correct, then only the dealing phase is completed successfully. Dealer discards  $S, a_1, a_2, \dots, a_{t-1}$  and policy.
10. Otherwise, it is up to the honest shareholders to decide whether it is the Dealer or the accuser that misbehaves.

### 3.2 Share Reconstruction Phase

Input: Shares  $S_i$  where  $i \in \{1, 2, 3, \dots, n\}$  and  $\geq t$ , a public hash function H and policy.

Output: Secret S.

1. Dealer verifies each share by generating hash code for each share and make SK and apply it to policy, accepts if it pass the policy otherwise add in cheater set.
2. Dealer verifies that each  $S_i$  is unique and deletes the repetition of same share.
3. If  $t$  or more than  $t$  shares are available then the dealer computes an interpolated polynomial  $f(x)$  at  $t$  or more points  $(i_1, S_1), (i_2, S_2), \dots, (i_t, S_t)$ .  
 Here, if we assume that we have  $j, j > t$  shares than we make two sets  $set_1 = \{S_1, S_2, \dots, S_t\}$  and  $set_2 = \{S_{t+1}, S_{t+2}, \dots, S_j\}$ . Then generate initial secret  $S$  from  $set_1$  and store. Replace each  $S_i, S_i \in set_2$  with  $S_1$  in  $set_1$  and generate secret  $S$  and compare with previously stored secret S.

If at any point secret match fails then dealer must added forged share in policy, otherwise return S as secret.

## 4 Analysis

In our algorithm, we extend the Shamir's original threshold secret sharing scheme [1] to verify the shares and the secret. For a threshold value  $t$ , we choose a random polynomial of degree  $t-1$  where the coefficients are also chosen randomly in  $\text{GF}(p)$  of prime order  $p$ . We set the secret as a constant term of the polynomial. Now we can use the polynomial to generate the shares of a secret and distribute it among a set of shareholders. Up to this point our scheme works the same as

Shamir's scheme [1]. Thereafter, we generate a hash function based on the part of secret for each part. We also make a policy using OR threshold gate, which requires any one condition in the given policy to be true in order to successfully decrypt the message. If the combiner (other than TA) wants to generate the secret then after receiving the parts, it can send each part to a dealer for generating the secret key based on the

hash value and check if it satisfies the policy. If it is so, then the secret is allowed to be reconstructed, otherwise not.

We show a typical snapshot of the execution of our scheme using the CP-ABE toolkit [22]. We assume that dealer D has a secret  $S=30$ . The dealer divides  $S$  into 5 parts and gives each shareholder  $S_i$ , the hash of the part of the secret. In the snapshot shown in Fig 1, the hash values of the five parts are 31, 28,43,83,61 respectively.

```

$ cpabe-setup
$ ls
master_key pub_key
$ cpabe-keygen -o s1_priv_key pub_key master_key 'value=31'
$ cpabe-keygen -o s2_priv_key pub_key master_key 'value=28'
$ cpabe-keygen -o s3_priv_key pub_key master_key 'value=43'
$ cpabe-keygen -o s4_priv_key pub_key master_key 'value=83'
$ cpabe-keygen -o s5_priv_key pub_key master_key 'value=61'
$ cpabe-keygen -o s6_priv_key pub_key master_key 'value=1'
$ ls
master_key pub_key s1_priv_key s2_priv_key s3_priv_key s4_priv_key s5_priv_key s6_priv_key
$ gedit message.txt
Congratulations, You had right part of share
$ cpabe-enc pub_key message.txt
1 of (value = 31, value = 28, value = 43, value = 83, value = 61 )

// NOTE: Now file message.txt will become message.txt.cpabe which is our public file.

// User 1 tries to decrypt the message
$ ls
S1_priv_key message.txt.cpabe pub_key
$ cpabe-dec pub_key s1_priv_key message.txt.cpabe
$ ls
S1_priv_key message.txt pub_key

//Now, a user 6 which does not have the right part try to decrypt the policy tries doing so
$ ls
S6_priv_key message.txt.cpabe pub_key
$ cpabe-dec pub_key s1_priv_key message.txt.cpabe
Cannot Decrypt, attributes in key do not satisfy policy

```

Figure 1 Snapshot of execution of the proposed scheme in the CPABE toolkit

## 5 Conclusions and future work

In this paper we propose an innovative approach for VSS using the ABE called AB-VSS. Our approach is resilient against attacks which are prevalent against the existing schemes for VSS. Currently we are using only one attribute per user for designing the scheme. In a setup that demands higher security, we can extend the existing scheme for other attributes like *location*, *time* etc. Such a scheme would employ  $t$  number of attributes for the policy. If the policy is satisfied, then the secret may be given to the shareholders, otherwise not.

## 6 References

- [1] Shamir, A. "How to share a secret." In: *Communication of the ACM*, Volume 22, Issue 11, pp. 612-613, (1979).
- [2] Blakley, G.R. "Safeguarding cryptographic keys." In: *Proceedings of the AFIPS1979 NCC*. Volume 48, pp. 313-317, (1979).
- [3] Chor, B., Goldwasser, S., Micali, S., Awerbuch, B. "Verifiable secret sharing and achieving simultaneity in the presence of faults." In: *SFCS '85: Proceedings of the*

- 26th Annual Symposium on Foundations of Computer Science, pp. 383-395, 1985.
- [4] Cohen Benaloh, J. "Secret sharing homomorphisms: Keeping Shares of a Secret." In: *CRYPTO-86: Proceedings on Advances in Cryptology*, pp 251-260, 1987.
- [5] Feldman, P. "A practical scheme for non-interactive verifiable secret sharing." In: *SFCS '87: Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, pp.427-438, 1987.
- [6] Harn, L., Lin, C. "Detection and identification of cheaters in (t, n) secret sharing scheme." In: *Des. Codes Cryptography*, Volume 52, Issue 1, pp. 15-24, 2009.
- [7] Tompa, M., Woll, H. "How to share a secret with cheaters." In: *Journal of Cryptology*, Volume 1, Issue 2, pp. 133-138, 1988.
- [8] Cao, Z., Markowitch, O. "Two optimum secret sharing schemes revisited." In: *FITME '08: Proceedings of the 2008 International Seminar on Future Information Technology and Management Engineering*, pp. 157-160, 2008.
- [9] Obana, S., Araki, T. "Almost optimum secret sharing schemes secure against cheating for arbitrary secret distribution." In: *Advances in Cryptology ASIACRYPT 2006*. Pp. 364-379, 2006.
- [10] Araki, Toshinori and Obana, Satoshi. "Flaws in Some Secret Sharing Schemes against Cheating." In: *LNCS 4586*, pp. 122-132, 2007.
- [11] Rabin, T., Ben-Or, M. "Verifiable secret sharing and multiparty protocols with honest majority." In: *STOC '89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pp. 73-85, 1989.
- [12] Ghodosi, Hossein. "Comments on Harn-Lin's cheating detection scheme." In: *Designs, Codes and Cryptography*, Springer, 2010.
- [13] Basu, Atanu and Sengupta, Indranil. "Verifiable (t, n) Threshold Secret Sharing Scheme Using ECC Based Signcryption." In: *Information Systems, Technology and Management Communications in Computer and Information Science*, pp.133-144, Volume 54, Issue 3, 2010.
- [14] T. Araki and S. Obana. "Flaws in some secret sharing schemes against cheating." In: *Proceedings of the ACISP 2007*, LNCS 4586, pp. 122-132. Springer-Verlag, 2007.
- [15] Sahai A, Waters B. "Fuzzy identity-based encryption." In: *Proceeding of EUROCRYPT 2005*, pp. 457-473, Springer, 2005.
- [16] Goyal V, Pandey O, Sahai A, et al. "Attribute based encryption for fine-grained access control of encrypted data." In: *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89-98, ACM, New York, 2006.
- [17] Bethencourt J, Sahai A, Waters B. "Ciphertext-policy attribute-based encryption." In: *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P 2007)*, pp. 321-334, IEEE, 2007.
- [18] Cheung L, Newport C. "Provably secure ciphertext policy ABE." In: *Proceedings of the 14th ACM conference on Computer and Communications Security*, pp. 456-465, ACM, New York, 2007.
- [19] Zhibin Z., and Dijiang H. On Efficient Ciphertext-Policy Attribute Based Encryption and Broadcast Encryption. [Online]. Available: <http://eprint.iacr.org/2010/395.pdf>.
- [20] Emura, K., Miyaji, A., Nomura, A., Omote, K., Soshi, M. "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length." In: *Bao, F., Li, H., Wang, G. (eds.) Proceedings of the ISPEC 2009. LNCS 5451*, pp. 13-23. Springer, Heidelberg 2009.
- [21] Paterson, K., Quaglia, E. "Time-specific encryption." In: *J. Garay (ed.) Proceedings of Seventh Conference on Security and Cryptography for Networks*, 2010.
- [22] The CP-ABE toolkit. [Online]. Available: <http://acsc.cs.utexas.edu/cpabe/cpabe toolkit>.

# A Trust Model for Routing in MANETs : A Cognitive Agents based Approach

B. Sathish Babu<sup>1</sup>, and Pallapa Venkataram<sup>2</sup>

<sup>1</sup>Dept. of Computer Science and Engg.,  
Siddaganga Institute of Technology, Tumkur-572 103, India  
E-mail: bsb@sit.ac.in

<sup>2</sup>Protocol Engineering and Technology Unit, Dept. of ECE.,  
Indian Institute of Science, Bangalore-560 012, India  
E-mail:pallapa@ece.iisc.ernet.in

**Abstract**—*Mobile ad hoc networks (MANETs) is one of the successful wireless network paradigms which offers unrestricted mobility without depending on any underlying infrastructure. MANETs have become an exciting and important technology in recent years because of the rapid proliferation of variety of wireless devices, and increased use of ad hoc networks in various applications. Like any other networks, MANETs are also prone to variety of attacks majorly in routing side, most of the proposed secured routing solutions based on cryptography and authentication methods have greater overhead, which results in latency problems and resource crunch problems, especially in energy side. The successful working of these mechanisms also depends on secured key management involving a trusted third authority, which is generally difficult to implement in MANET environment due to volatile topology. Designing a secured routing algorithm for MANETs which incorporates the notion of trust without maintaining any trusted third entity is an interesting research problem in recent years. This paper propose a new trust model based on cognitive reasoning, which associates the notion of trust with all the member nodes of MANETs using a novel Behaviors-Observations-Beliefs (BOB) model. These trust values are used for detection and prevention of malicious and dishonest nodes while routing the data. The proposed trust model works with the DTM-DSR protocol, which involves computation of direct trust between any two nodes using cognitive knowledge. We have taken care of trust fading over time, rewards, and penalties while computing the trustworthiness of a node and also route. A simulator is developed for testing the proposed algorithm, the results of experiments shows incorporation of cognitive reasoning for computation of trust in routing effectively detects intrusions in MANET environment, and generates more reliable routes for secured routing of data.*

**Keywords:** MANETs; routing; trust; DTM-DSR; security; Cognitive agents; BOB model

## 1. Introduction

As an important concept in network security, trust is interpreted as a set of relations among nodes/entities participates in network activities. Trust relations are mainly based on previous behaviors of nodes/entities. The concept of trust is the same as within real life, where we trust people who have been helpful and acted trustworthy towards us in the past. In the case of ad hoc networking, nodes that operate the protocols correctly are considered as trusted nodes. The purpose of developing a notion of trust within an ad hoc network is to provide a heuristic for security. Allowing faulty or malicious nodes to be detected and removed from the network, with minimal overhead and restriction to the network.

There are three definitions of trust as follows [1]:

- 1) Trust is the subjective probability of one entity expecting that another entity performs a given action on which its welfare depends. The first entity is called trustor, while the other is called trustee.
- 2) Direct trust refers to an entity's belief in another entity's trustworthiness within a certain direct interaction to a certain direct experience.
- 3) Recommendation trust refers to one entity which may also believe that another entity is trustworthy due to the recommendations of other entities with respect to their evaluation results.

Trust management in distributed and resource-constraint networks, such as MANETs and sensor networks, is much more difficult but more crucial than in traditional hierarchical architectures, such as the Internet and infrastructure based wireless LANs. Generally, these types of distributed networks have neither pre-established infrastructure, nor centralized control servers or trusted third parties (TTPs).

The trust information or evidence used to evaluate trustworthiness is provided by peers, i.e. the nodes that form the network. The dynamically changing topology and con-

nectivity of MANETs establish trust management more as a dynamic systems problem. Furthermore, resources (power, bandwidth, computation etc.) are normally limited because of the wireless and ad hoc environment, so the trust evaluation procedure should only rely on local information. Therefore, the essential and unique properties of trust management in this paradigm of wireless networking, as opposed to traditional centralized approaches, are: uncertainty and incompleteness of trust evidence, locality of trust information exchange, distributed computation, and so on. We are addressing this issue by storing the evidence of trust calculation in the form of beliefs generated in beliefs database stored with every mobile node.

### 1.1 Agents

Agents are the autonomous programs which sense the environment, acts upon the environment, and use its knowledge to achieve their goal(s) [2]. An agent program can assist people or programs and some occasions acts on their behalf. Agents possess the mandatory properties such as: *reactiveness*: agents sense changes in the environment and acts according to those changes, *autonomy*: agents have control over their own actions, *goal-orientation*: agents are proactive, and *temporal continuity*: agents are continuously executing software. A typology of agents refers to the study of classification of agents based on some of the key attributes, exhibited by agent programs [3]. The agents are classified into *Static and Mobile agents, Deliberative and Reactive agents, and Smart or intelligent agents*.

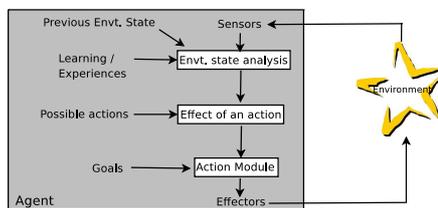


Fig. 1: Intelligent agent structure

An ideal rational/intelligent agent should do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the precept sequence and whatever built-in knowledge the agent has. An agent should possess explicit utility functions to make rational decisions. Goals, enable the agent to pick an action right away if it satisfies the intended goal. The Fig. 1, gives the structure of an intelligent agent, showing how the current precept is combined with the old internal state, and experiences of an agent to generate the updated description of the current state. To do this intelligent agents use both learning and knowledge. Here the actions are not simple precept sequence based, an intelligent agent should able to

reason out the suitable action from set of possible actions, either reactively/pro-actively [4].

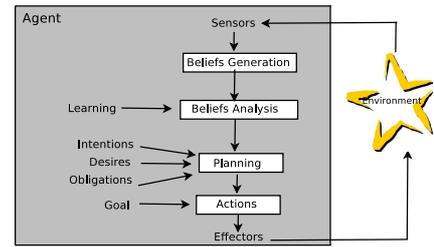


Fig. 2: Cognitive agent structure

Cognitive agents enable the construction of applications with: *context sensitive behavior, adaptive reasoning, ability to monitor and respond to situation in real time, and implements human cognitive architecture for knowledge organization*. A cognitive act performed by cognitive agents consists of three general actions: 1. Perceiving information in the environment; 2. Reasoning about those perceptions using existing knowledge; and 3. Acting to make a reasoned change to the external or internal environment. For an application perspective, CAs empower a user by combining the speed, efficiency and accuracy of the computer with the decision making capacity, experience and expertise of human experts. The agent represents its beliefs, intentions, and desires in modular data structures and performs explicit manipulations on those structures to carry out means-ends reasoning or plan recognition (refer Fig. 2).

### 1.2 DTM-DSR

The proposed model uses, the DTM-DSR (Dynamic Trust Mechanism - Dynamic Source Routing) protocol which is an extension of the DSR protocol [5]. In DTM-DSR, i) every node maintains trust table; ii) the route request message (RREQ), includes  $T_{low}$  and *BlackList*, where,  $T_{low}$  denotes the node's lower trust level on its neighbor, *BlackList* denotes distrusted node list; and iii)  $T_{route}$  field in the route reply message (RREP), denote the accumulated route trust.

#### 1.2.1 Route Discovery

During the process of route discovery, when a node *A* chooses another node *B* to forward a packet, *A* may suffer some attacks from *B*, such as black hole attack, wormhole attack, etc. Thus, a reliable relationship between *A* and *B* should be established. A trusted route represents a route that only involves trustworthy nodes, sending packets by the trusted route will decrease the probability of malicious attacks and improve the survivability of MANETs. The trustworthiness of a route is evaluated by trust values of nodes along the route, denoted by  $T_{route}$ . The route discovery includes three processes: i) RREQ delivery; ii) RREP

delivery; and iii) Route selection, which are briefly discussed as follows.

#### RREQ delivery

When the source node  $S$  needs to send data to the destination node  $D$ , it first checks whether there is a feasible path found between  $S$  and  $D$ . If so,  $S$  sends the data to  $D$ ; otherwise,  $S$  will start a route discovery. First,  $S$  appends its  $ID$  into the route record, and checks whether the trust on its neighbor nodes is lower than  $T_{low}$ . If so,  $S$  appends the  $ID$  of neighbor nodes into *BlackList*. Then,  $S$  broadcasts the RREQ packets with  $T_{low}$  and *BlackList*, and sets a timer window  $t_s$ . When any intermediate node receives a RREQ packet, it processes the request according to the following steps:

- 1) If the requested  $ID$  for this RREQ packet is found in the node's list of recently seen requests, then it discards the RREQ packet and does not process it further.
- 2) if the target of the request matches the node's own address, then the route record in the packet contains the route by which the RREQ reached this node from the source node of the RREQ packet. Intermediate node returns a copy of this route in a RREP packet to the source node.
- 3) Otherwise, it appends its own address to the route record in the RREQ packet, and checks whether the trust on its neighbor nodes is lower than  $T_{low}$ . If it is, it appends the  $ID$  of the neighbor nodes into *BlackList*.
- 4) Re-broadcast the request to the neighbor nodes.

#### RREP delivery

When the destination node receives the first RREQ packet, it sets a timer window  $t_d$ . If  $t_d$  expires, it discards the follow-up RREQ packet. Otherwise, it checks whether the *BlackList* is empty. If not, it discards the RREQ packet; otherwise, it sets  $T_{route} = 1$ , and then unicasts the RREP packet with  $T_{route}$  to the intermediate node. After receiving a RREP packet, the intermediate node computes  $T_{route}$ , and updates the value of  $T_{route}$ , then it forwards the RREP packet with  $T_{route}$ .

#### Route selection

When  $S$  receives the RREP packet, if the timer window  $t_s$  does not expire, it needs to update the  $T_{route}$  value of this message. Otherwise,  $S$  discards follow-up RREP packets and picks a path with the largest  $T_{route}$  with less number of hops.

#### 1.2.2 Route Maintenance

After each successful route discovery takes place,  $S$  can deliver its data to  $D$  through a route. However, the route may break at any time instance due to the mobility of nodes, or attacks. In order to maintain a stable, reliable and secure network connection, route maintenance is necessary to ensure the system survivability. Route maintenance is performed when all routes fail or when the timer window  $t_r$  for routing expires.

### 1.3 Proposed Trust Model

The trust model proposed in this paper is based on the DTM-DSR protocol. Every mobile node is assumed to host platform for executing agents built with cognitive intelligence. The Cognitive Agents (CAs) on a mobile node use the BOB-model to compute the trustworthiness of its neighboring nodes. The trust is computed based on the beliefs generated by observing neighboring nodes behaviors while forwarding the data. These trust values are used in the DTM-DSR protocol to route the packet from a given source to a given destination.

### 1.4 Rest of the paper

The rest of the paper is organized as follows; section 2 lists some of the related work, section 3 discuss the proposed trust model, section 4 provides simulation setup and results, and section 5 concludes the paper.

## 2. Related work

Existing works that are related to trust based security can be studied under two dimensions. First, the trust evaluation models in MANETs and the Second is the trust/reputations based routing protocols in MANETs. Many of the existing trust-based routing protocols are the extensions of the popular routing protocols, such as DSR and AODV.

### 2.1 Trust Evaluation Model

These models includes methods for evaluating the trust based on various parameters. Entropy based trust models are employed in ad hoc networks for secure ad hoc routing and malicious node detection [6]. It is based on a distributed scheme to acquire, maintain, and update trust records associated with the behaviors of nodes forwarding packets and the behaviors of making recommendations about other nodes. But it is not a generic mathematical model and can not prevent the false recommendations. A semiring-based trust model [7] interpret the trust as a relation among entities that participates in various protocols. This work is focusing on the evaluation of trust evidence in ad hoc networks, because of the dynamic nature of ad hoc networks, trust evidence may be uncertain and incomplete. Using the theory of semirings, it shows how two nodes can establish an indirect trust relation without previous direct interaction. The model has more dynamic adaptability, but its convergence is slow and cannot be adopted in large scale networks. To solve the vulnerabilities with existing trust management frameworks, a robust and attack-resistant framework called the objective trust management framework (OTMF) based on a modified Bayesian approach is proposed [8]. The theoretical basis for OTMF is a modified Bayesian approach by which different weights are put on different information related to observations of behaviors according to their occurrence time and providers.

A reputation based system is an extension to source routing protocols for detecting and punishing selfish nodes in MANETs [9]. In a mobile ad hoc network, node cooperation in packet forwarding is required in order for the network to function properly. However, some selfish nodes might intend not to forward packets in order to save resources for their own use. To discourage such behavior, a reputation-based system is proposed, to detect selfish nodes and respond to them by showing that being cooperative will benefit them more than being selfish. In this paper, besides cooperative nodes and selfish nodes, a new type of node called a suspicious node is introduced. These suspicious nodes will be further investigated and if they tend to behave selfishly, some actions are taken against them. A trust model based on Bayesian theory is proposed in [10]. The model assesses subjective trust of nodes through the Bayesian method, which makes it easy to obtain the subjective trust value of one node on another, but it cannot detect dishonest recommendations. A fuzzy trust recommendation framework [11], and the recommendation algorithm is based on collaborative filtering in MANETs has been proposed. It considers recommendation trust, but does not consider other factors, such as the time aging and the certainty nature of trust.

## 2.2 Trust/Reputation-based Routing Protocols

These algorithms makes use of the existing routing protocols, and during the routing the trust or the reputation values are included. A dependable routing by incorporating trust and reputation in the DSR protocol is proposed [12]. The mechanism makes use of Route Reply packets to propagate the trust information of nodes in the network. These trust values are used to construct trusted routes that pass through benevolent nodes and circumvent malicious nodes. But it does not consider how to prevent dishonest recommendation in the trust model. The cooperative on-demand secure route (COSR) protocol is used to defend against the main passive route attacks [13]. COSR measures node-reputation (NR) and route-reputation (RR) by contribution, Capability of Forwarding (CoF), and recommendation to detect malicious nodes. Watchdog and Pathrater techniques are proposed in [14]. The Watchdog promiscuously listens to the transmission of the next node in the path for detecting misbehavior's. The Pathrater keeps the ratings for other nodes and performs route selection by choosing routes that do not contain selfish nodes. However, the watchdog mechanism needs to maintain the state information regarding the monitored nodes and the transmitted packets, which would add a great deal of memory overhead. The extension to the above collaborative reputation (CORE) mechanism [15], uses the watchdog mechanism to observe neighbors, and aims to detect and isolate selfish nodes.

## 3. Proposed BoB-based Trust Model for MANET routing

Most statistical methods assume that the behavior of a system is stationary, so the ratings can be based on all observations back to the beginning of time. But often the system's behavior changes with time, and our main interest is to identify and isolate intrusion nodes by keeping the theory of rewarding the positive behaviors and punishing the negative behaviors intact. Fig. 3 shows the deployment of Cognitive Agents (CAs) over every mobile node belongs to MANET under consideration. The CA present on a node is responsible for computing the trust over its neighboring nodes, group of CA's collaboratively participate in establishing the trusted route from a given source  $S$  to a given destination  $D$ . We assume these CAs are secured enough and tamper-resistant from any host-based attacks.

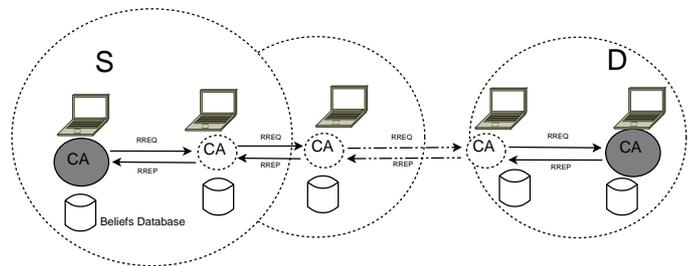


Fig. 3: The Proposed BOB-based Trust Model for MANET

### 3.1 The Behaviors-Observations-Beliefs (BOB) model

The BOB model is a cognitive theory based model proposed in our earlier paper [16], to generate beliefs on a given mobile node, by observing various behaviors exhibited by the node during execution of routing protocol. The BOB model is developed by giving emphasis on using the minimum computation and minimum code size, by keeping the resource restrictiveness of the mobile devices and infrastructure. The knowledge organisation using cognitive factors, helps in selecting the rational approach for deciding the trustworthiness of a node or a route. The rational approach implements systematic, step by step method in which data obtained through various observations is used for making long-term decisions. It also reduces the solution search space by consolidating the node behaviors into an abstract form called beliefs, as a result the decision making time reduces considerably.

#### Behaviors

The behaviors refer to the actions or reactions of a node while executing routing protocols. The behaviors are modeled using a set of behavior parameters. In general the probability  $P_{Bh_i}$  of generating a  $i^{th}$  behavior  $Bh_i$  is computed using the behavior parameters set,  $BP_i$ .

$$P_{Bh_i} = \frac{\sum_{k \in BP_i} W_{bp_k} * V_{bp_k}}{\sum_{k \in BP_i} W_{bp_k} * \max(V_{bp_k})} : \sum_{k \in BP_i} W_{bp_k} = 1 \quad (1)$$

Where  $W_{bp_k}$ ,  $V_{bp_k}$ , and  $\max(V_{bp_k})$  are the weightage given to each behavior parameter in the set  $BP_i$ , current value generated for the behavior parameter  $bp_k$ , and the maximum value the behavior parameter  $bp_k$  can take respectively. If the value of  $V_{bp_k}$ , tends more and more towards  $\max(V_{bp_k})$ , the probability of generation of the behavior  $Bh_i$  increases.

### Observations

In the system an observation is the summarization of various behaviors exhibited by a node during protocol execution. The probability of generating observation,  $Ob_i$ , i.e.,  $P_{Ob_i}$ , is computed using the union of occurrence of defined set of behaviors which leads to that observation. Let  $BH_{Ob_i}$  is the set of disjoint behaviors considered for  $i^{th}$  observation  $Ob_i$ .

$$P_{Ob_i} = P(Bh_a^i \cup Bh_c^i \cup Bh_k^i \cup \dots \cup Bh_m^i) \quad (2)$$

Where  $Bh_a^i, Bh_c^i, Bh_k^i, \dots, Bh_m^i \in BH_{Ob_i}$ , where  $a \leq j \leq m$ .

### Beliefs

A belief represents an opinion with certain confidence about a node. These beliefs are stored in a beliefs database, and periodically updated as and when the new beliefs on the event occurs. The probability of occurrence of a belief,  $P_{Bf_i}$ , is the union of those observations which will generate that particular belief. Let  $O_{Bf_i}$  is the observations set for belief,  $Bf_i$ .

$$P_{Bf_i} = P(Ob_c^i \cup Ob_f^i \cup Ob_l^i \cup \dots \cup Ob_n^i) \quad (3)$$

Where  $Ob_c^i, Ob_f^i, Ob_l^i, \dots, Ob_n^i \in O_{Bf_i}$ , where  $c \leq j \leq n$ .

## 3.2 Working of CA

The CA comprised of constructs used to implement the BOB model, the constructs are the logical structures used for periodic collection and analysis of behavior parameters of a mobile node, the BOB model uses four constructs namely: *Behaviors identifier*, *Observations generator*, *Beliefs formulator*, and *Beliefs analyser* as shown in the Fig. 4.

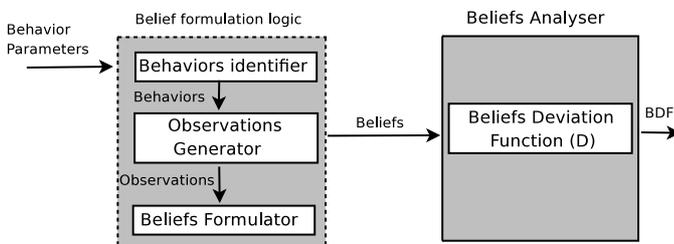


Fig. 4: The BOB model constructs built into CA

The *Behaviors identifier* construct periodically captures behavior parameters related to a mobile node. A set of behavior parameters participate in triggering one or more behaviors. A threshold-based triggering function ( $F$ ) is implemented to identify each behavior. The  $F$  accepts a set of behavior parameters, and computes the triggering value, if the value is greater than threshold then that behavior is successfully identified. The *Observations generator* construct generates one or more observations on identified behaviors. The summarization function generates an observation by enumerating number of favorable behaviors to generate an observation. If the number of favorable behaviors are less than the expected value, then that observation is not generated, otherwise an observation is generated. We propose to keep the percentage of favorable behaviors as least as 40% to generate an observation, so that the accuracy of the system is increased. The *Beliefs formulator* construct deduce belief(s) from one or more generated observations. Suitable logical relations are established between observations to construct predicates to deduce various belief(s). The *Beliefs analyser* construct analyse the newly formed beliefs, say  $Bl^{new}$ , to compute the Belief Deviation Factor (BDF) with respect to established beliefs, say  $Bl^{old}$ . The deviation function  $D$  finds the relative deviation between two beliefs, satisfies the distance property, i.e., increased distance between two beliefs produce higher deviations and vice versa.

## 3.3 Trust Modeling using CAs

The trust modeling using CAs in our scheme is explained in following steps.

**Step 1:** The behavior analysis is carried out by the CA on a mobile node over the actions a neighboring mobile node(s) takes over the data they receives for routing. Some of the malicious behaviors of a node includes; 1. data dumping, 2. energy draining, 3. suspicious pairing, 4. data delaying, 5. data reading, 6. data fabrication, etc. In our scheme we have modeled all these malicious behaviors using set of behavior parameters, which includes; time for forwarding, hard disk operations, energy level, next hop address, size of data received/forwarded, next hop used, and so on. These set of behaviors are accumulated over a time to generate observations, such as; formation of wormhole, formation of blackhole, denial of service, intrusion attempts, modification attempts, etc. The related observations are deduced into beliefs on a node, example, genuine node, intruder, service hijacker, wormhole attacker, blackhole attacker, route cache poisoner, etc.

**Step 2:** Generated beliefs on the neighboring nodes in the current time period  $\Delta t$ , are compared with established beliefs from the beliefs database stored in a node in order to compute the Belief Deviation Factor (BDF). The CA calculates the deviation factor between the probability values of newly computed beliefs, i.e.,  $P_{Bl}^{new}$ , by comparing them with the established corresponding probability values of

beliefs from the *beliefs database*, i.e.,  $P_{Bl}^{old}$ .

$$DF(Bl^{new}, Bl^{old}) = |P_{Bl}^{new}, P_{Bl}^{old}| \quad (4)$$

Exponentially moving averages are used to accumulate deviation factors of beliefs generated during various time instances. The weights for each deviation decreases exponentially, giving much more importance to current deviation while still not discarding older deviations entirely. The smoothing factor  $\alpha$  is given as,

$$\alpha = \frac{2}{\text{Number of Routing Requests} + 1} \quad (5)$$

The BDF at time  $t$  is given by,

$$BDF_{Bl}^t = \alpha \times DF(Bl^{new}, Bl^{old}) + (1 - \alpha) \times BDF_{Bl}^{t-1} \quad (6)$$

**Step 3:** The BDF is then combined with Time-aging Factor (TF), Rewards Factor (RF), and Penalty Factor (PF), to calculate the direct trust of a node  $i$  over its neighbor node  $j$  in time  $\Delta t$ , which is given by  $T_{new}^d(i, j)$ .

if  $(T_{old}^d(i, j) > 0$  and  $BDF = 0)$  then

$$T_{new}^d(i, j) = 1 - TF \times T_{old}^d(i, j) \quad (7)$$

if  $((T_{old}^d(i, j) > 0$  and  $BDF \neq 0)$  then

$$T_{new}^d(i, j) = T_{old}^d(i, j) \times (1 - TF \times (RF \times N_1 - PF \times N_2)) + TF \times (RF \times N_1 - PF \times N_2) \quad (8)$$

Where:  $TF = \frac{\lambda e^{C_1 \Delta t} - 1}{\lambda e^{C_1 \Delta t} + 1}$ , represents the trust fades with time.  $RF = \frac{\lambda e^{C_2 \times BDF / \Delta t} - 1}{\lambda e^{C_2 \times BDF / \Delta t} + 1}$ , represents the positive impact of trust when the BDF is low during  $\Delta t$ .  $PF = \frac{\lambda e^{C_3 \times (1 - BDF) / \Delta t} - 1}{\lambda e^{C_3 \times (1 - BDF) / \Delta t} + 1}$ , represents the negative impact of trust when the BDF is high during  $\Delta t$ .  $\lambda, C_1, C_2$  and  $C_3$  are determined according to practical requirements.  $N_1 = \frac{\text{Number of Times}(BDF < \text{Lower Threshold})}{\text{Number of Times}(BDF < \text{Lower Threshold}) + 1}$  and  $N_2 = \frac{\text{Number of Times}(BDF > \text{Higher Threshold})}{\text{Number of Times}(BDF > \text{Higher Threshold}) + 1}$ .

## 4. Simulation and Results

Following assumptions are made in the simulated network: 1. Each node has the same transmission radius; and 2. Each node knows the IDs of its neighbor nodes by exchanging their control information. Some of the parameters used in simulation are mobility speed, amount of data to be routed in a CBR mode, node bandwidth, and message sending duration. When the simulation began randomly chosen nodes participate in the routing process with source and destination pair. In this process if any nodes trust values have reached the lower value, then those nodes are considered as malicious nodes. The detected malicious nodes are not allowed in further routing process until their trust values are increased. Fig. 5 to Fig. 8 shows snapshots of simulator developed.

Fig. 9 shows throughput plotted for various simulation scenarios, we can observe the throughput decreases as more

Fig. 5: The sample MANET topology

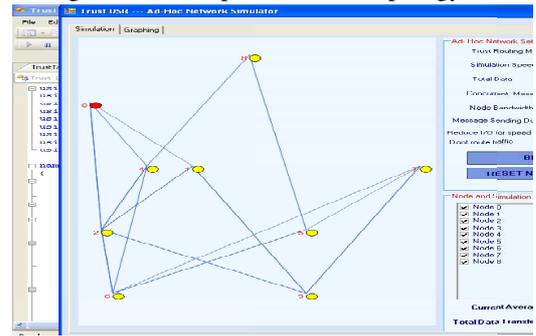
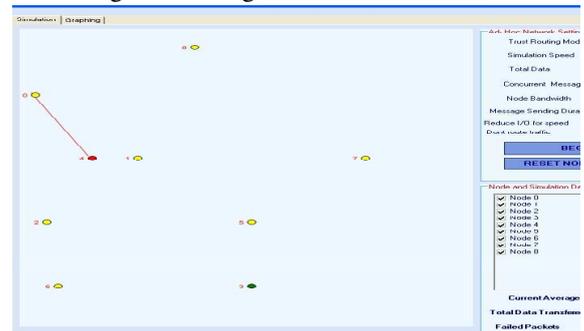


Fig. 6: Routing started from 0



and more nodes loose trust, and marked as intruders by the system. Fig. 10 shows variation of trust of four neighboring nodes of selected mobile node no 6. The result shows trust values remains constant for a node irrespective time, it slowly increases for two more nodes, and decreases for one node over a time.

## 5. Conclusions

The proposed trust-based routing using the BOB-model based on cognitive theory performs efficiently over the DTM-DSR and DSR, since the computation of trust is linked with belief generation and belief deviation. The delay incurred in computing the trust is very less compared to the DTM-DSR protocol, since the cognitive theory based knowledge is used. We could able to establish more reliable routes compared to previous two algorithms by isolating the intruder nodes from routing. We are conducting detailed performance analysis by subjecting the protocol into various routing conditions, and also incorporating trust calculated by recommendations by peers.

## References

- [1] V. Balakrishnan, V. Varadarajan, and U. Tupakula, "Trust Management in Mobile Ad Hoc Networks," Guide to Wireless Ad Hoc Networks, Springer, ISBN: 9781848003286, 2009.

Fig. 7: Node 1 and Node 3 trust calculation is on

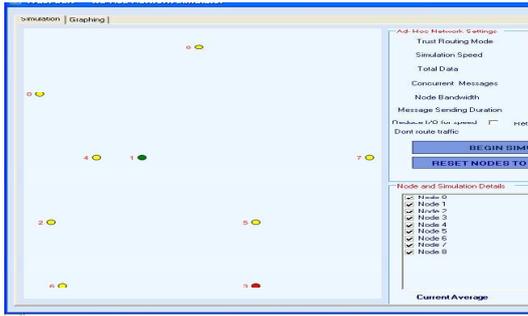


Fig. 9: Network Throughput



Fig. 8: Node 3 is detected as intruder

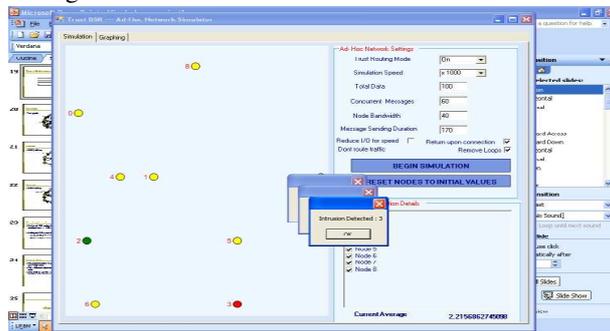
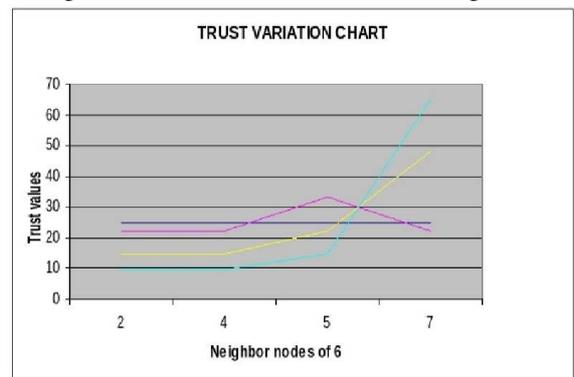


Fig. 10: Variation of trust values among nodes



[2] J. Bradshaw, "Software agents," AAAI press, California, 2000.  
 [3] H. S. Nwana, "Software Agents: An Overview," Knowledge Engineering Review, vol. 11, No 3, pp.1-40, 1996.  
 [4] N. R. Jennings and M. Wooldridge, "Applications of Intelligent Agents," 1998.  
 [5] S. Peng, W. Jia, G. Wang, J. Wu, and M. Guo, "Trusted Routing Based on Dynamic Trust Mechanism in Mobile Ad-Hoc Networks", IEICE TRANS. on Information and Systems.  
 [6] Y. Sun, W. Yu, Z. Han, and K. J. R. Liu, "Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks," IEEE Journal on Selected Areas in Communications, Vol.24, pp. 305-317, 2006.  
 [7] G. Theodorakopoulos and J. S. Baras, "On Trust Models and Trust Evaluation Metrics for Ad Hoc Networks," IEEE Journal on Selected Areas in Communications, Vol. 24, Issue 2, pp. 318-328, Feb.2006.  
 [8] J. Li, R. Li, J. Kato, "Future Trust Management Framework for Mobile Ad Hoc Networks," IEEE Communications Magazine, Vol. 46, Issue 4, pp.108-114, April 2008.  
 [9] T. Anantvalee and J. Wu, "Reputation-Based System for Encouraging the Cooperation of Nodes in Mobile Ad Hoc Networks," Proc. of IEEE International Conference on the Communications (ICC 2007), pp. 3383-3388, June 2007.  
 [10] S. Peng, W. Jia, and G. Wang, "Voting-Based Clustering Algorithm with Subjective Trust and Stability in Mobile Ad-Hoc Networks," Proc. of the IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2008), Vol. 2, pp. 3-9, Dec. 2008.  
 [11] J. Luo, et al., "Fuzzy Trust Recommendation Based on Collaborative Filtering for Mobile Ad-hoc Networks," Proc. of the 33rd IEEE Conference on Local Computer Networks (LCN 2008), pp. 305-311, Oct. 2008.  
 [12] A. A. Pirzada, A. Datta, and C. McDonald, "Incorporating trust and

reputation in the DSR protocol for dependable routing," Elsevier of Computer Communications, Vol. 29, pp. 2806-2821, 2006.  
 [13] F. Wang, Y. Mo, B. Huang, "COSR: Cooperative On-Demand Secure Route Protocol in MANET," International Symposium on Communications and Information Technologies (ISCIT 2006), pp. 890-893, Oct. 2006.  
 [14] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," Proc. of MobiCom, Boston, MA, pp. 255-265, August 2000.  
 [15] P. Michiardi and R. Molva, "Core: A Collaborative Reputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks," Communication and Multimedia Security Conference, September 2002.  
 [16] B.S. Babu and P. Venkataram, "Cognitive agents based authentication & privacy scheme for mobile transactions (CABAPS)," Computer Communications, vol. 31, pp. 4060-4071, 2008.

# An Approach for Automatic Selection of Relevance Features in Intrusion Detection Systems

Shan Suthaharan and Karthik Vinnakota

Department of Computer Science, University of North Carolina at Greensboro, Greensboro, NC 27412, USA

**Abstract** - In intrusion detection systems the environment data in general depends on many computer network related characteristics called features. However not all of the network features contribute to the discriminating properties of different types of intrusion attacks. Hence the selection of relevance features becomes an important requirement for the accurate and faster detection of intrusion. Several Rough Set Theory (RST) based feature selection approaches have been proposed and the effectiveness of these approaches has been tested using the KDD'99 intrusion detection dataset. However the RST implementation requires user interaction and hence the automatic analysis of intrusion detection datasets and the detection of intrusion types become more difficult. In this paper we propose a probability distribution based approach that extract appropriate information from the intrusion data and supplies that information to the RST implementation so that the relevance features can be selected automatically. The proposed automatic feature selection approach simplifies and automates the detection of intrusion attacks with added advantages of high accuracy and less computing time.

**Index Terms**—Intrusion detection, rough set theory, automatic feature selection, probability distribution

## 1. Introduction

Computer network security has been a topic of great significance due to the number of attacks and intrusions on the network. Several preventive measures, such as firewall and anti-virus software, have been deployed on computer networks to monitor and control intrusion attacks. One type of such preventive measure is the latest Intrusion Detection System (IDS). The IDS collects and analyses the information from various areas within a computer or a network of computers to identify possible security breaches which include intrusions (attacks from outside the organization) and misuse (attacks from within the organization). The IDS is expected to complement the firewall security management in an efficient way. The firewall and antivirus software protect organizations from the intrusion attacks, whereas the IDS sense the possibility of potential threats and notify the threats to the concerned security officials. The intrusion detection systems can be divided into two systems namely, host-based intrusion detection system and network-based intrusion detection system [1]. The host-based intrusion detection

monitors and detects intrusions at system level, whereas the network-based intrusion detection system analyses and detects intrusions in network level. This paper considers issues associated with the network-based intrusion detection system. The models used in intrusion detection systems can also be categorized as follows: Misuse Detection Model (MDM) and Anomaly Detection Model (ADM). The MDM analyses the system and network, and compares the activity against signatures of known attacks. The ADM assumes that the breach in the computer or network security can be detected by observing a deviation from the “normal” system and network behaviors. In this paper we consider the ADM model treating the intrusion as anomaly in the observations.

Intrusion detection systems in general deal with large datasets and known attacks that depend on many features. In order to improve the detection accuracy and to reduce the computational time, efficient data mining and machine learning techniques should be implemented on a subset to select relevance features. The data mining techniques allow us to select an appropriate subset of an intrusion detection dataset. The machine learning techniques are used to learn the regular patterns from a subset of a large dataset and then apply the knowledge to the large dataset. This machine learning idea has been used with the large intrusion detection dataset KDD '99 to find contributing features for intrusion detection. In practice 10% of the KDD'99 dataset is considered for training and the entire dataset for testing. The KDD'99 dataset is generated at Lincoln labs, where an environment was set up to acquire seven weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. The network was operated as a true Air Force environment except that the attacks on the network were intentional. The entire dataset and 10% dataset are available for research [2]. This dataset witnesses the network behavior depends on many features. However the research shows that not all of these features affect the type of network behavior such as normal, intrusion or misuse. Hence it is important to select relevance features for the accurate detection of intrusion type and reduce computational cost.

Several feature selection techniques have been proposed to serve this purpose. Feature selection is the technique for selecting a subset of relevant features for building an efficient learning model. Feature selection improves the performance

of learning models by enhancing generalization capability and speeding up the training and testing process. Feature selection in KDD '99 dataset is a process of selecting relevance features from the total number of 41 features with respect to the type of intrusion attack. Several other techniques like, information gain [3], wrapper based feature selection [4], recursive feature elimination and k-nearest neighbor [5], fuzzy association rule mining [6] were used in the selection of relevance features.

In this paper we study and examine the relevance of features in KDD '99 dataset with respect to an intrusion attack. This paper is organized as follows: section II presents the details about KDD'99 dataset, in section III RST approach is discussed; section IV describes the proposed approach, simulation results and discussion in Section V and lastly conclusion in section VI.

## 2.KDD'99 Dataset and Properties

Intrusion activities in the computer communication network have been the problem for last several decades. To address this problem KDD'99 intrusion detection dataset has been developed by Lincoln Labs [2]. Since then significant research in intrusion detection has been carried out using this dataset. The dataset is very large and hence a size-reduced dataset (10% KDD) has been created from the original 100% dataset and used in research as a training dataset.

The original KDD'99 dataset and the size-reduced dataset contain 743Mb and 75Mb of data respectively [2]. The 100% dataset provide data on normal network behavior and 39 common intrusion attacks, whereas the 10% dataset provide data on normal network behavior and 22 common intrusion attacks. In addition they present 41 features that contribute to these attacks. 10% KDD dataset has been used as a training dataset to work on IDS which deals with classification problem and feature selection. Although the KDD'99 datasets provide several attacks and several features not all of the features contribute to an attack. Therefore it is important to study the dataset and select relevance features that contribute to a particular attack. This will make the IDS systems more efficient by reducing the computational cost.

The research shows back attack is one of the important intrusions that significantly affect the performance of a TCP connection. In the 10% KDD dataset the Denial of Service attacks were a total of 391,458 records i.e. 79.24% [8] of the dataset. Back attack is one of the members of Denial service attacks. However in terms of selecting relevance features for the back attack conflicting results are obtained. For example, the research by Kayacik et al. [3] used Information Gain as a measure of contribution with 10% KDD dataset in selecting relevance features and they concluded that feature 10 and feature 13 contributing features for back attack. Similarly, Olusola et al. [8] used Degree of Dependency as a measure of contribution in selecting the relevance features and they concluded that features 5 and 6 are contributing features for back attack.

Although the size-reduced dataset reduces the computational time of selecting relevance features, these approaches still suffer from user interaction, accuracy and computational efficiency. Therefore to make the IDS more efficient we select a random sample from the dataset (either 10% or 100% datasets) and select relevance features for simplicity and automation of IDS system.

To address these problems we propose an intrusion detection system that is capable of automatically select relevance features. It will use KDD'99 dataset and RST to find the relevance features. The RST is discussed in section III and the proposed approach is discussed in section IV.

## 3.Rough Set Theory

The concept of Rough set theory (RST) was first described by a polish scientist Zdzisław I. Pawlak [9]. It was founded on the assumption that with every object of the universe of discourse some information (data, knowledge) is associated [10]. The RST is concerned with classification of incomplete or uncertain data and also helps in identification and evaluation of dependent data.

The basic definitions of RST as presented by Olusola *et. al.* [8] are as follows:

Information System (IS): It is defined as  $IS = (U, A, V, f)$  where  $U$  is the finite set of objects;  $U = \{u_1, u_2, \dots, u_n\}$ ,  $A$  is the finite set of attributes;  $A = \{a_1, a_2, \dots, a_m\}$ ,  $V$  is the value set of attributes  $A$ ;  $V_{a_1}, V_{a_2}, \dots, V_{a_m}$ ,  $f$  is a decision function;  $f(x, a) \in V_a$  for every  $a \in A$  and  $x \in U$  and  $A' = \{A \cup Q\}$ , where  $A$  is the set of condition attributes and  $Q = \{q_1, q_2, \dots, q_s\}$  is the set of decision attributes. The size  $s(Q)$  is much smaller than the size  $s(U)$ . Hence we have more than one object mapped to a single decision attribute.

In practice not all the objects are valid and not all the attributes contribute an object. Therefore, it is appropriate to select a subset of attributes and a subset of objects.

Let  $X = \{x_1, x_2, \dots, x_m\}$  be a subset of  $U$ , where  $m < n$ ,  $X_i$  could be empty or nonempty and the objects in  $X_i$  corresponds to  $q_j$  where  $q_j \in Q$ . Let  $B = \{b_1, b_2, \dots, b_l\}$  be a subset of  $A$  where  $l < m$ . The subset  $X$  can be classified into indiscernible subsets based on the similar attribute values  $V_{b_1}, V_{b_2}, \dots, V_{b_l}$  of  $B$ . We can now define the Indiscernible subset  $C_k(B)$  of set  $X$  as follows:

If  $x, y \in C_k(B)$ , then  $f(x, b) = f(y, b) \forall b \in B, i = 1, 2, \dots, N$  and  $\bigcup_{i=1}^N C_k(B) = X$ , such that  $C_{k1}(B) \cap C_{k2}(B) = \emptyset$  for  $k1 \neq k2$ .

With the indiscernible subset, we need to choose the most reliable indiscernible subsets, which can be achieved by defining lower approximation based on some predefined set of objects. Choosing such a predefined set of objects for the

user is a difficult task. In this paper we provide a scheme that automates this process by using statistical based approach. Therefore, the lower approximation can be defined as follows:

$Y_k(B, q_j) = \{C_k(B) \mid \forall x, \text{ if } x \in C_k(B), \text{ then } x \in Y\}$ , where  $k$  can have more than one value which ranges from 1 to  $N$  and  $j$  ranges from 1 to  $s$ .

The set of Lower approximations is given by:

$$Y(B, q_j) = \bigcup_{k=1}^N Y_k(B, q_j) \tag{1}$$

The Positive region of attributes  $B$  is defined by

$$POS_B(Y, q_j) = \{y \mid y \in Y(B, q_j)\} \tag{2}$$

The Degree of dependency of  $q_j$  on attributes  $B$  over the objects  $U$  is denoted by  $\gamma_B(Y, q_j)$  and is defined as follows:

$$\gamma_B(Y, q_j) = \frac{|POS_B(Y, q_j)|}{|U|} \tag{3}$$

The Degree of Dependency helps us in determining the relevance features i.e. higher the degree of dependency higher is the probability of that feature being relevant for an intrusion attack. The Degree of Dependency ranges from 0 to 1.

In traditional RST approaches the set  $Y$  is selected by the user. This selection of  $Y$  may result in conflicting values. In order to overcome these results, we introduce an approach which automates the process of selecting the set  $Y$  from  $U$ .

Therefore the effectiveness of the use of RST depends on the suitable selection of the subset  $Y$  of  $U$ , the subset  $B$  of  $A$  and the user defined set  $Y$  (we automate this). The proposed automated approach addresses this problem by selecting appropriate objects by eliminating the false positives using our previous technique (ellipsoid based approach), by selecting the features based on the findings from others and automating the RST by selecting the user defined set  $Y$  using a statistical approach.

### 4.Proposed Approach

In this paper we use the randomized and automated approach of selecting the relevance features from KDD dataset. This process is categorized as follows: -

#### 4.1New KDD dataset:

This new data set contains only some of the features out of the 41 different features and less number of attacks. This is done to reduce the computational time for feature selection. In our research we selected back intrusion attack to be compared with normal behavior and selected features 5, 6, 10, 13, 32 and 34. Previous research on KDD 99 dataset indicates that

feature 5 and feature 6 [8] are most relevance features of back attack and other research indicates that feature 10 and feature 13 [3] are most relevant features of back attack.

#### 4.2Random Selection of Objects:

The data contains a lot of redundant data and also outliers'. The redundancy is first removed from all the data. In order to eliminate the outliers (false positives and negatives) from the data we use a threshold. We use the sum of mean and standard deviation as the threshold and eliminate the outliers. The 3-dimensional graphs of the redundant data are plotted in Fig. 1 through to Fig. 4.

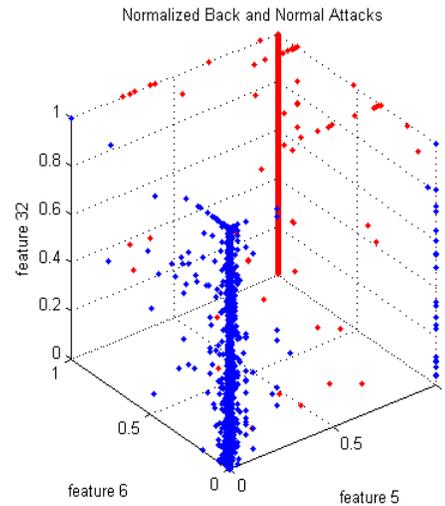


Fig 1: Relationships between the normalized back attack and normal with respect to features 5, 6 and 32.

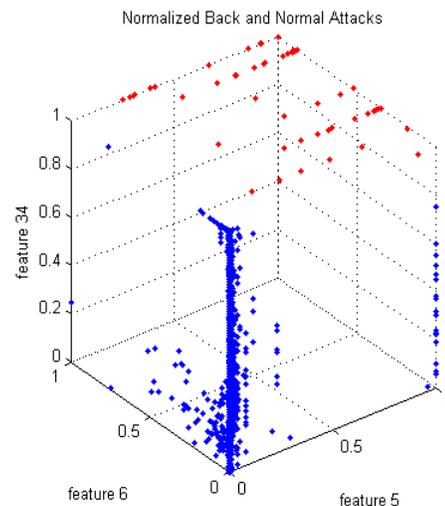
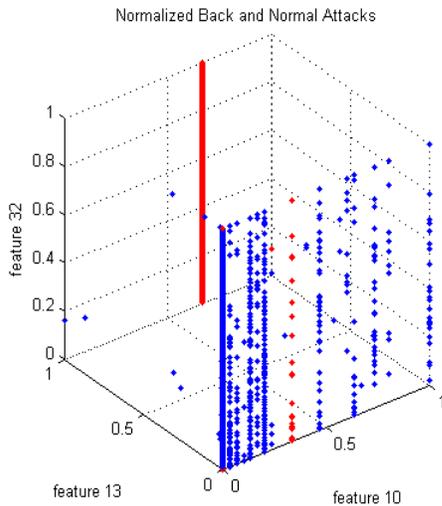
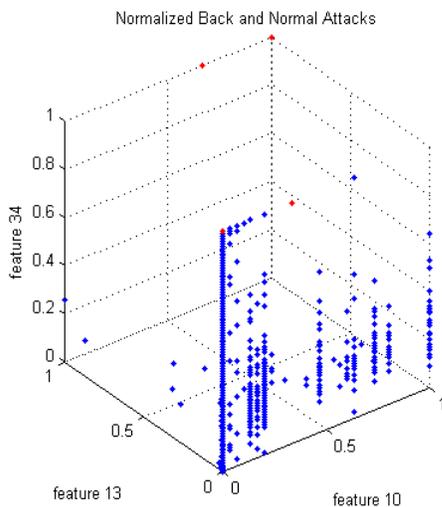


Fig 2: Relationships between the normalized back attack and normal with respect to features 5, 6 and 34.



**Fig 3:** Relationships between the normalized back attack and normal with respect to features 10, 13 and 32.



(d)

**Fig 4:** Relationships between the normalized back attack and normal with respect to features 10, 13 and 34.

Figs. 1 through to 4 show many outliers. With the use of threshold, we can eliminate these outliers so that normal and definite attacks are selected. This data, after eliminating outliers is made discrete with the help of ROSE2 rough set explorer [11].

### 4.3 Change in RST:

Significant changes have been made to the rough set theory and to tailor our needs. In the traditional RST, user had to calculate the indiscernible subsets and also select the set of object set Y, which are used in calculating the lower

approximations and also the degree of dependency. In this paper we use a program that finds all the indiscernible subsets for the given sample of data and also automates the selection of this object set Y. This automation is done based on the sum of mean and standard deviation. Lower approximations are calculated based on the object set Y and the indiscernible

subsets with the use of  $Y(B, q_j) = \bigcup_{k=1}^N Y_k(B, q_j)$ . Then we

calculate the degree of dependency for the selected attribute set B and object set Y. But, in this paper we have used the weighted approach of calculating the degree of dependency. The formula is as follows:

$$\gamma_B(Y, q_j) = \frac{\sum_{i=0}^n S(i) * DoDs(i)}{\sum_{i=0}^n S(i)} \tag{4}$$

It takes the effect of sample size into consideration in the calculation of the average degree of dependency.

**Table I:** Degree of dependencies for features 5, 6, 10, 13 and 32.

Sample Size Increase S(i)	Degree of Dependency 10, 13 and 32 DoDs(i)	Degree of Dependency 10 and 13 DoDs(i)	Degree of Dependency 5, 6 and 32 DoDs(i)	Degree of Dependency for 5 and 6 DoDs(i)
10	1.0000	1.0000	1.0000	1.0000
20	1.0000	1.0000	1.0000	1.0000
30	1.0000	0.9500	1.0000	1.0000
40	1.0000	1.0000	0.9125	1.0000
50	1.0000	0.9500	0.8600	0.9700
60	1.0000	0.9667	0.8750	1.0000
70	1.0000	0.9643	0.9500	1.0000
80	1.0000	0.9563	0.9938	1.0000
90	1.0000	0.9278	0.8778	1.0000
100	1.0000	0.9550	0.9650	1.0000
110	1.0000	0.9727	0.9091	1.0000
120	1.0000	0.9550	0.9292	1.0000
130	1.0000	0.9385	0.8808	1.0000
140	1.0000	0.9857	0.8929	0.9607
150	1.0000	0.9833	0.9267	0.9967
160	--	--	0.9156	1.0000
170	--	--	0.8971	1.0000

## 5. Simulation Results and Discussion

The results obtained are quite interesting. After the process of eliminating the redundant data and the outliers, the number of records of each feature was as follows:

- Feature 5, 6 and 32 has 171 back and 32699 normal.
- Feature 10, 13 and 32 has 153 back and 326 normal.

These numbers of records were taken into account for calculating the degree of dependency. The degrees of dependencies are presented in Table I. From our change in the RST in pervious section, we have a new formula for calculating the average degree of dependency. Using eq. (4), we calculate the degree of dependency for back attack, with respect to combination of features 10, 13 and 32; 10 and 13; 5, 6 and 32; and 5 and 6 as 1.0000, 0.9634, 0.9167 and 0.9951 respectively.

## 6. Conclusion

From our simulation with using only few features we can conclude that if 2 features are selected 5 and 6 are relevance features and if three features are selected 10, 13 and 32 are relevance features for back attack. Instead of using the 100% or 10% KDD datasets, we can consider random samples of the data to calculate the relevance features. This calculation gives acceptable accuracy with less computational time. Our future work will be to introduce new methods for eliminating the outliers and calculating the degree of dependency.

## References

- [1] A. S. Ashoor and S. Gore, "Importance of Intrusion Detection system (IDS)". International Journal of Scientific and Engineering Research, vol. 2, no. 1, pp.1-4, Jan-2011.
- [2] <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.
- [3] H. G. Kayacik, A. N. Zincir-Heywood and M. I. Heywood, "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets". Association of Computer Machinery, 2006.
- [4] Y. Li, J. Wang, Z. Tian, T. Lu and C. Young, "Building lightweight intrusion detection system using wrapper-based feature selection mechanisms". Computer and Security, vol. 28, pp.466-475, 2009.
- [5] K. T. Khaing, "Enhanced Features Ranking and Selection using Recursive Feature Elimination(RFE) and k-Nearest Neighbor Algorithms in Support Vector Machine for Intrusion Detection System". International journal of Network and Mobile Technologies, vol. 1, no. 1, pp.8-14, June-2010.
- [6] M. Sheikhan and M. S. Rad, "Misuse Detection Based on Feature Selection by Fuzzy Association Rule Mining". World Applied Sciences Journal, pp.32-40, 2010.
- [7] <http://kdd.ics.uci.edu/databases/kddcup99/task.html>.
- [8] A. A. Olusola, A. S. Oladele and D. O. Abosede, "Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features". Proceedings of the World Congress on Engineering and Computer Science, vol. 1, Oct-2010.
- [9] [http://en.wikipedia.org/wiki/Rough\\_set](http://en.wikipedia.org/wiki/Rough_set).
- [10] Z. Pawlak "Rough Set theory and its applications". Journal of Telecommunications and Information Technology, pp.7-10, March-2002.
- [11] <http://idss.cs.put.poznan.pl/site/rose.html>.

# Performance Evaluation of Machine Learning Methods for Intrusion Detection

Yasir Javed, Shafique Ahmad Chaudhry , Mohammed Habeeb Vulla  
{kianiyasir,hazrat.shafique,habeebvulla}@gmail.com

*Department of Computer Science, Al-Imam Muhammad bin Saud University, Saudi Arabia*

## Abstract

*An Intrusion detection system (IDS) is used to secure a computer system from the malicious activities over the network. There are a number of systems that have been developed for capturing the malicious traffic over the network but all tend to be heavy on the system resources usage. Therefore, there is a need to develop an IDS that is lightweight and can adapt to changes as it runs over the time on network. In this paper we used ID3, J.48 and linear regression techniques in order to predict the behavior of user and the traffic generated over the network by doing its classification. We provide a comparative study between ID3, J.48 and linear regression. The rules extracted during the study also provides basis for developing a light weight IDS that is adaptable to network dynamics. Results shows j.48 is better among all algorithms but in development of IDS combination of all techniques can be applied at multiple levels over the network.*

**Key Words:** Intrusion Detection System, ID3, J.48, Linear Regression, Machine Learning, Pruning, Classification

## 1. Introduction

Network systems are deployed in order to share information with others. Sometime information is intended for some special person or group of persons. This information if accessed by others can create potential threats to person, projects, states, inter (inside) and intra-relations (foreign relations). Thus utmost important task is to secure the information. The threats to security of network are both physical and software. For physical trespassing physical security can be implemented by elite force and installing highly sensitive sensors in valuable area. However, in providing security to information over network cannot be achieved either by hiring commando services or installing cameras and sensors as there is no physical interaction with trespasser. The trespassers are to be stopped without knowing what they are doing, what they plan to do, what will be next step that can be taken and what is their location. Software trespasses [2] can be in the two forms. One it can be in form of worm, Trojan horse or virus. Secondly it can be in form of Humans

attacking security in three shapes [2], Masquerader, Misfeasor and Clandestine User. Masquerader is the one which penetrates into the system and exploit legitimate user account although he is not authorized to do so. While Misfeasor is an authorized person who misuses his privileges or uses such resources or programs to whom he is not allowed. Clandestine User is one who takes over supervisory control and evade access or auditing control. There are only three measures to avoid intrusion [1][2][3]. One is prevention and other one is detection while next one is combination of both called hybrid. This means first detect whether intrusion is taking place or not then figure out how it happened, then plan and implement a solution for that, this will make security system powerful and adaptive in nature, thus up-to-date to fight new challenges. For intrusion detection, we need to have a classification system that will classify an interloper and authorized user. There are various IDS system proposed some [9][29][30][31] used machine learning classification techniques. Our focus is on exploiting machine learning techniques for development of light weight classification based IDS.

In this paper we have compared iterative Dichotomizer3 (ID3) [5], J.48 (an extension of C.45 algorithm) and linear regression [6][34]. The comparisons are made by considering multivariable network traffic [8]. Our preliminary results show that by applying ID3 and J.48 on multivariate data, we can cut down the cost by omitting the variables that don't add considerable amount of impact over the decision. Rules that have been developed by applying the algorithms can be used in developing new IDS [10][11][12][38]. These rules also ensure that size of new IDS will be very small and it can adapt to changes. The rest of the paper is organized as follows. In section 2 we discussed the basic algorithm used, that are ID3, J.48 and Linear regression, in section 3. We present implementation details and preliminary evaluation results in section 4 and 5 respectively. Paper is concluded in section 7.

## 2. ID3, J.48 and Linear Regression

ID3 algorithm and J.48 [6] algorithms are developed by Quinlan [7]. J.48 algorithm is a modification of C.45 algorithm that is a modification of ID3 algorithm. C.45 contains all the functionalities of ID3 but it also

overcomes the problems of ID3 that is C.45 can handle continuous attributes [39]. J.48 is often referred as special implementation of C.45. J.48 algorithm prunes the tree that is extra functionality. Pruning [13] means cutting out the un-useful branches that will not contribute in rules and the height of tree will be reduced. Lesser the height of tree easier to predict the rule and lesser will be the complexity. It is based on algorithm given by William of Ockham a French logician and a priest [5][14][15]. The algorithm is famously known as Occam's razor[14] that states that all things are of same importance and simpler solution is the best solution. ID3 algorithm does not always give the minimum tree [16], as it based on heuristic and heuristic is entropy calculated as shown in (1) or information gain calculated as shown in (2). The equations are adapted as explained in [17] [18][19][20]

$$Entropy(S) = \varepsilon - \rho(I) \log \rho(I) \quad (1)$$

In the above formula  $\rho(I)$  is the proportion of S belonging to class I.

$$Gain(S, A) = Entropy(S) - S \left( \frac{|S_v|}{|S|} \right) * Entropy(S_v) \quad (2)$$

Where as

$S$  is each value  $v$  of all possible values of attribute  $A$

$S_v$  = subset of  $S$  for which attribute  $A$  has value  $v$

$|S_v|$  = number of elements in  $S_v$

$|S|$  = number of elements in  $S$

Linear Regression assumes that data must be in numeric form [21]. Linear regression also assumes the data in two dimensions [22], so we have to select the best attribute among. For this the root nodes extracted by ID3 or J.48 can be used as x-axis node. Slope intercept ( $y$ ) can be calculated using (3).

$$y = mx + b \quad (3)$$

$x, y$  = Data subset

$n$  = number of data points

$$m = \frac{n \sum(xy) - \sum x \sum y}{n \sum(x^2) - (\sum x)^2}$$

$$b = \frac{\sum y - m \sum x}{n}$$

Correlation coefficient can be calculated by the following equation

$$r = \frac{n \sum(xy) - \sum x \sum y}{\sqrt{[n \sum(x^2) - (\sum x)^2] * [n \sum(y^2) - (\sum y)^2]}}$$

### 3. Challenges faced during rules extraction

In order to develop certain IDS we need to develop rules that classifies the activities of network to be legitimate or not. Traffic that is received on network has many parameters and if we start considering every parameter's role in classification then comparing the traffic against each rule will take considerable amount of time, that will make network slow. Dimensionality reduction is a way of reducing the co-ordinates [13][23][32]. It helps in considering only those variables that play important role in decision making or classification. We need to use the techniques in such a way that it must reduce dimensions but not at the cost of poor performance. In case of network traffic there are number of variables that come along with single packet. Following variables are important in network traffic packet.

- *Session\_Index* refers to the unique session took place.
- *Start\_Date* tells on which date the session took place.
- *Start\_time* refers to the time of that session.
- *Duration* tells that for how much time the session remained.
- *Service* is the name of the service that was requested to the server to be executed.
- *Source\_port* is the port from where the server will run the service.
- *Destination\_port* is the port of the requesting party requesting for the specific service.
- *Source\_IP* is the IP address of the source.
- *Destination\_IP* is the IP address of the requesting party.

To obtain classified data, we used the classified dataset provided by MIT's Lincoln's Laboratory [8]. It includes two level of classification one is just classifying that whether attack has took place or not and other is type of attack that occurred, for first they used the name Score and to other they called Attack\_name.

- *Score* refers to whether the session was generated by attacker or a legitimate user.

- *Attack\_name* refers to the name of attack made by hacker.

To understand the data Figure-1 tells how the data is distributed. In the figure-1, 1 is session index, 1/23/2007 is the date at which the session took place, and 16:56:12 tells the time at which the session took place and the session lasted for 1 min and 26 seconds. Telnet was the

Session_Index	Start_Date	Start_time	Duration	Service	Source_Port	Destination_Port	Source_IP	Destination_IP	Score	Attack_name
1	1/23/2007	16:56:12	0:01:26	telnet	1754	23	192.168.1.192.168.0.	0-		

Figure1:- Description of network packet

#### 4. Implementation

We took the classified dataset and run ID3, J.48 and Linear regression for rule extraction. Initially we took ID3 to extract useful elements in network traffic. Firstly prune out those variables that have no or very less change, like Start\_Date can be ignored if only single day data is to be considered. Pruning can be done if we find entropy and information gain of all the variables involved. After considerable amount of work we figured out that Start\_Date should be ignored, as it has no contribution. While Session\_Index is continuously changing thus it should not be considered as participating variable else always Session\_Index will get priority. Start\_time will have a same impact as Session\_Index so it can be used in profiling user or profile based IDS else for routing traffic it can be ignored. For rest we will make test with two variables sets, In one we will consider all the variables except omitted one and In next case we will not consider duration, Source\_port and Destination\_port in order to extract out more rules and make comparison between both datasets. After that same rules were employed for J.48 to see its effects on correctly classification of traffic. For linear regression we considered full all variable, in next experiment we considered only Duration, Service, Source\_Port, Destination\_Port, Source\_IP, Destination\_IP, and Score to make a clear comparison on impact of reducing the variables on the result. We took 359 classified records for as our base set for rule extraction, following scatter graph shows the dataset distribution with respect to Session\_Index (On X-axis) and Attack\_Name (On Y-axis). We will consider first division of variables to be testcase1 and second division of variable to be testcase2.

service that was requested on server port number 1754 the requesting port was 23. 192.168.1.0 Is the server's IP while 192.168.0.1 is the requesting party IP. The session was not of hacker so there is no attack name as specified by '-'

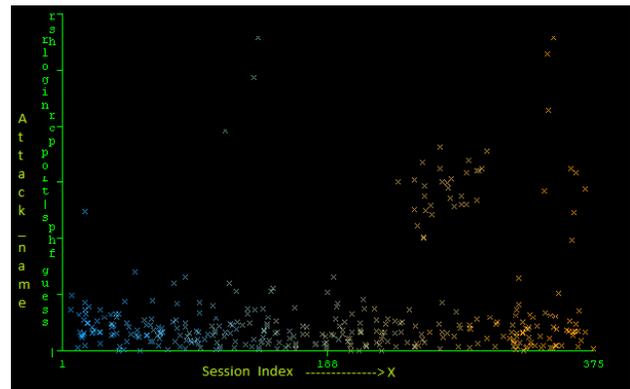


Figure 2:- Occurrences of attack type from classified dataset

#### 5. Results

The results obtained have multivariate analysis of correlation, Mean squared error, Kappa statistics, and Relative absolute error.

- Correlation [4] [24] means the data chosen is dependant on each other that means all the attributes chosen has high dependency.
- Mean squared error refers to the difference between predicted values and actual values and their square overall divided by no of instances.
- Root mean squared [4] [25] error means the taking the square root of mean squared error. This is known to all that error should be minimum.
- Kappa statistic [4] [26] is measured by subtracting actual class value from predicted class value.
- Relative error is predicted value minus actual value squared divided by mean actual error square.

**Table 1: Statistical Analysis of ID3**

Parameters	Instances	Value
Correctly Classified Instances	126	35.0975 %
Incorrectly Classified Instances	3	0.8357 %
Kappa statistic		0.9322
Mean absolute error		0.0024
Root mean squared error		0.0489
Relative absolute error		7.5182 %
Root relative squared error		36.9744 %
Unclassified Instances	230	64.0669 %
Total Number of Instances		359

Relative absolute error is very important because if there are 10 incorrectly classified instance out of 100 examples and there are one incorrectly classified instance out of 10. There is difference between both. Because we cannot say that both data set has same error rate because when we move ahead the things might change. The kappa statistic is measured by adding all correctly classified instances that adding diagonal values in confusion matrix. The maximum value of kappa statistic can be 100%. We also used WEKA [27] the machine learning tool to extract out rules and then tested on traffic.

The results obtained by running both the datasets are promising in development of IDS.

Table 1 shows the statistical test done on ID3 testcase1 and it shows that error rate is very less. Mean absolute error is .0024 that is really ignorable while in ID3 testcase2 is 0.0191 not shown in tables is more. Same is the case with other errors like Relative absolute error is 15.815% in testcase2. This shows that ID3 works better with eight attribute (testcase1). Table 2 shows the statistical tests obtained by J.48 algorithm.

**Table 2: Statistical Analysis of J.48**

Parameters	Instances	Values
Correctly Classified Instances	332	92.4791 %
Incorrectly Classified Instances	27	7.5209 %
Kappa statistic		0.6365
Mean absolute error		0.0317
Root mean squared error		0.1285
Relative absolute error		41.6597 %
Root relative squared error		67.4834 %

made on J.48 for testcase1, In the results shown below Mean absolute error is 0.0317 while for testcase2 the Mean absolute error is 0.0359 (not shown in tables) thus there is not much difference. Root mean square error is 0.1285 in test case1 while for testcase2 it is 0.1375. Thus testcase1 is slightly better than testcase2. This shows that J.48 also works better with eight attribute. Table 3 shows the statistical tests made on Linear Regression for testcase1, In the results shown below Mean absolute error is 0.1449 while for testcase2 the Mean absolute error is

0.2057 (not shown in tables) thus there is slight difference.

**Table 3: Statistical Analysis of Linear Regression for testcase1**

Parameter	Value
Correlation coefficient	0.9165
Mean absolute error	0.1449
Root mean squared error	0.4519
Relative absolute error	19.7949 %
Root relative squared error	39.93 %
Total Number of Instances	361

Same is the case with other tests like Root mean square error is 0.4519 in test case1 while for testcase2 it is 0.3008. Thus testcase2 is far better than testcase1. But Relative absolute error in testcase1 is less than of testcase2 that is 83.126% that is very much and it makes testcase2 strongly un-considerable. Thus in Linear regression the testcase1 (that means whole message must be considered) is better. But the aim to reduce number of variable is not achieved as we have to consider full packet. For this we made different test named testcase3. With five variables same as we considered for j.48 and ID3 gave results near to testcase1 as shown in Table 4. For linear regression if we compare testcase1 and testcase3, they are nearly equal. If we compare Mean absolute error in testcase3 it is 0.1399 that means it is Fairly near to testcase1, similarly the case is similar for Root mean squared error. Thus the testcase3 is nearly equal in performance to testcase1, but the number of variable are only five (5). Thus for Linear regression we will consider testcase3.

**Table 4:- Statistical Analysis of Linear Regression for testcase3**

Parameter	Values
Correlation coefficient	0.9178
Mean absolute error	0.1399
Root mean squared error	0.4485
Relative absolute error	19.1054 %
Root relative squared error	39.6226 %
Total Number of Instances	361

Comparing the ID3, J.48 with each other, first they are classification algorithms and moreover results have shown that correctly classified instances are very higher in J.48 that is about 92% to 35% in ID3, Thus the extension of C.45, As shown in Figure 3 the correctly classified instances are nearly one (1) that means attacks will be detected with high probability. J.48 is to be preferred in case of Classification. While doing comparisons between linear Regression and J.48. J.48

seems to be better but linear regression is very important in clustering theory and only one of the most important techniques in clustering.

Clustering tries to classify the examples or instances in specific class based on linear regression function. The difference between classification and regression function is that in classification the output is generated as Boolean, that is either yes or no or some fixed value. However, when output is numeric then we are not interested in class but we are interested in a function that is continues in nature [35].

We do not have linear function in machine learning but we have training set from which we learn that function. Thus both carry weight as in linear regression about 90% of the time it will detect error. Moreover linear regression based IDS are light weight as they include one linear formula for classification. While J.48 is better as it has simple if-else rules to be implemented thus if a multi comparator processor is installed in IDS, detection of illegal traffic can be done very quickly.

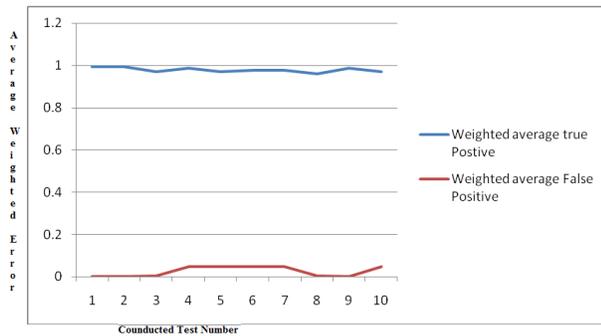


Figure 3:- Correctly Classified instances and Incorrectly Classified instance in J.48

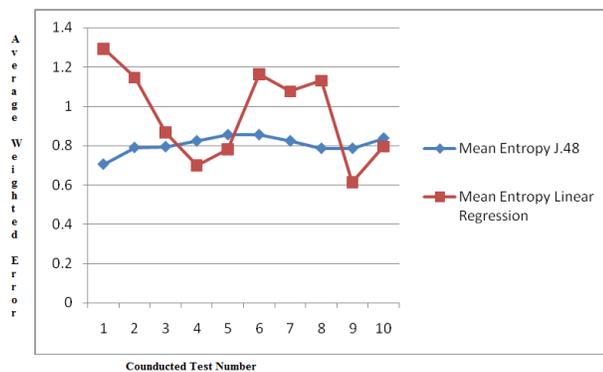


Figure 4: Entropy comparison between J.48 and Linear Regression

As shown in Figure 3 and Figure 4, Entropy in J.48 is consistent and it is steadily working at same level but in linear regression changes are high and this tends to lead in uncertainty thus the above argument also insist on taking J.48 into account for developing of IDS.

## 6. Conclusion

In this paper we presented a comparison of classification algorithm (ID3, J.48) and linear regression. It is found out the both J.48 and linear regression can be used in developing of light weight [28][36] IDS but in case of linear regression the threshold cushion (grey area) will be little bit more than j.48. In future we will make IDS[33][37] that works at multiple levels, at server or entry level we can use Linear Regression as it considers more traffic to be suspicious and after firewall we can use J.48 based IDS to correctly classify the attack and stop it there. Moreover for inside network traffic the traffic going out of network must also be monitored same way in order to avoid misfeasor attacks.

## References

- [1] Information Security: Principles and Practices by Mark Merkow & James Breithaupt ISBN# 0131547291.
- [2] Cryptography And Network Security By William Stallings 4/E, ISBN# 0131873164
- [3] Network Security first step by Tom Thomas ISBN # 81-297-0602-3.
- [4] Morgan Kaufmann Data Mining Practical Machine Learning Tools and Techniques Second Edition Jun 2005.
- [5] J.R. Quinlan (1979): "Discovering Rules by Induction from large Collections of Examples," in D. Michie (ed.): *Expert Systems in the Micro Electronics Age*, Edinburgh University Press.
- [6] J. R. Quinlan. Improved use of continuous attributes in c.45. *Journal of Artificial Intelligence Research*, 4:77-90, 1996.
- [7] J. R. Quinlan. C4.5: Programs for machine learning. Morgan Kaufmann Publishers, 1993.
- [8] <http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/attackDB.html>
- [9] Balamurugan, Subramanian and Rajaram, Ramasamy, "effective and efficient feature selection for large-scale data using Bayes' theorem", *International Journal of Automation and Computing*, pages 62-71, volume 6, issue 1, 2009.
- [10] Sandhya Peddabachigaria, Ajith Abraham, Crina Grosanc, and Johnson Thomas "Modeling intrusion detection system using hybrid intelligent systems", *Journal of Network and Computer Applications* Volume 30, Issue 1, January 2007, Pages 114-132.
- [11] Kruegel Christopher, Toth, Thomas , "Using Decision Trees to Improve Signature-Based Intrusion Detection" LNCS 2003.
- [12] Joong-Hee Lee, Jong-Hyok Lee, Seon-Gyoung Sohn, Jong-Ho Ryu, Tai-Myoung Chung, "Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System", *Advanced Communication Technology*, 2008. ICAC 2008.
- [13] Quinlan, J.R.: *Simplifying decision trees*, *International Journal of Man-Machine Studies*, 27, 221-234, 1987.

- [14] [http://en.wikipedia.org/wiki/Occam's\\_razor](http://en.wikipedia.org/wiki/Occam's_razor). Accessed on 10 Nov 2010.
- [15] [http://en.wikipedia.org/wiki/William\\_of\\_Ockham](http://en.wikipedia.org/wiki/William_of_Ockham). Accessed on 10 Nov 2010.
- [16] J. Safaei, H. Beigy, Boolean Function Minimization: The Information Theoretic Approach, In 15th IEEE/ACM International Workshop on Logic & Synthesis, (IWLS 2006), pp. 37-41.
- [17] [http://en.wikipedia.org/wiki/Entropy\\_\(information\\_theory\)](http://en.wikipedia.org/wiki/Entropy_(information_theory)). accessed on 10 Nov 2010
- [18] [http://en.wikipedia.org/wiki/Information\\_gain\\_in\\_decision\\_trees](http://en.wikipedia.org/wiki/Information_gain_in_decision_trees). Accessed on 10 Nov 2010.
- [19] Mitchell, Tom M. Machine Learning. McGraw-Hill, 1997. pp. 55-58.
- [20] J. R. Quinlan, Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106.
- [21] Numerical Methods with Applications Autar Kaw, Egwu Kalu, (2008).
- [22] Regression Analysis Wiley Series in Probability and Statistics. N.R. Draper, and H. Smith, (1998).
- [23] I. K. Fodor. A survey of dimension reduction techniques. Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory, 2002.
- [24] J. L. Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. The American Statistician, 42(1):59-66, February 1988.
- [25] [http://en.wikipedia.org/wiki/Root\\_mean\\_square\\_deviation](http://en.wikipedia.org/wiki/Root_mean_square_deviation). As accessed on 15th of Nov2010.
- [26] Carletta, Jean. Assessing agreement on classification tasks: The kappa statistic. Computational Linguistics, 22(2), pp. 249-254. 1996.
- [27] <http://www.cs.waikato.ac.nz/ml/weka/> Accessed on 12 Dec 2010.
- [28] M. Roesch. Snort - Lightweight Intrusion Detection for Networks. In 13th Systems Administration Conference - LISA 99, 1999.
- [29] Chris Sinclair, Lyn Pierce, Sara Matzner, "An Application of Machine Learning to Network Intrusion Detection," acsac, pp.371, 15th Annual Computer Security Applications Conference (ACSAC '99), 1999.
- [30] P. Chan and S. Stolfo. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In Proc. Fourth Intl. Conf. Knowledge Discovery and Data Mining, pages 164-168, 1998
- [31] L. Prodrmidis and S. J. Stolfo. Mining databases with different schemas: Integrating incompatible classifiers. In G. Piatetsky-Shapiro R Agrawal, P. Stolorz, editor, Proc. 4th Intl.Conf. Knowledge Discovery and Data Mining, pages 314-318. AAAI Press, 1998.
- [32] L. Prodrmidis and S. J. Stolfo. Pruning meta-classifiers in a distributed data mining system. In Proc of the First National Conference on New Information Technologies, pages 151-160, Athens, Greece, October 1998.
- [33] D. Denning. An Intrusion Detection Model. IEEE Transactions on Software Engineering, 13(2):222-232, 1987
- [34] N. BenAmor, S. Benferhat, and Z. ElOuedi. Naive Bayes vs Decision Trees in Intrusion Detection Systems. In The 19th ACM Symposium On Applied Computing - SAC 2004, Nicosia, Cyprus, March 2004.
- [35] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A Geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. Applications of Data Mining in Computer Security, 2003.
- [36] Su-Yun Wu, Ester Yen, Data mining-based intrusion detectors, Expert Systems with Applications, Volume 36, Issue 3, Part 1, April 2009, Pages 5605-5612.
- [37] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, Wei-Yang Lin, Intrusion detection by machine learning: A review, Expert Systems with Applications, Volume 36, Issue 10, December 2009, Pages 11994-12000.
- [38] Wenke, L., 1999. A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems. PhD dissertation, <http://www.cc.gatech.edu/~wenke/>.
- [39] Shihai Zhang, Shujun Liu, Shizhong Zhang, Jinping Ou, Guangyuan Wang, "C4.5-Based Classification Rules Mining of High-Rise Building SFIO," fskd, vol. 4, pp.467-472, 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2008



**SESSION**  
**PRIVACY AND RELATED ISSUES**

**Chair(s)**

**TBA**



# Anonymous Secure Routing Protocol for Wireless Metropolitan Networks

Ren-Junn Hwang<sup>1</sup>, and Yu-Kai Hsiao<sup>2</sup>

junhwang@ms35.hinet.net<sup>1</sup>, Shiaukae@gmail.com<sup>2</sup>

Department of Computer Science and Information Engineering Tamkang University Taipei, Taiwan

**Abstract**—This paper proposes efficient concepts of anonymous and secure routing protocol considering symmetric and asymmetric communication models for Wireless Metropolitan Networks. A wireless metropolitan network is a group of wireless access points and several kinds of wireless devices (or nodes) in which individual nodes cooperate by forwarding packets for each other to allow nodes to communicate beyond the symmetric or asymmetric model. Asymmetric communication is a special feature of Wireless Metropolitan Network because of the different wireless transmission ranges of wireless devices. With asymmetric communication model, message exchange can be more efficient in metropolitan scale network. Providing security and privacy in Wireless Metropolitan Networks has been an important issue over the last few years. This paper proposes concepts of routing protocol beyond symmetric and asymmetric model, which guarantees security and anonymity of the established route in a hostile environment, such as Wireless Metropolitan Networks. The routes generated by the proposed concept are shorter than those in prior works. The wireless clients out of access point wireless transmission range may anonymously discover a secure route to connect to the access point for Internet access via the protocol based on the proposed concepts. The proposed concepts enhance wireless metropolitan network coverage in assuring security and anonymity.

**Keywords:** Asymmetric communication, Wireless metropolitan networks, Secures routing, Anonymous routing

## 1 Introduction

Wireless Metropolitan network (WMNs) integrates several kinds of networks such as ad hoc networks and wireless infrastructure networks in metropolitan area. This kind of network is formed by access point and wireless clients. Wireless client can be any kind of wireless device. Access points function as a gateway/bridge in negotiating different kinds of networks. It allows wireless devices with different communication protocols to communicate each

other and provide a larger wireless coverage area than traditional wireless networks.

Wireless metropolitan networks (WMNs) combine several kinds of wireless devices. Each device may provide different communication and computation capability. WMNs provide different communication styles. In the WMN scenario in Figure 1, User *S* has a larger transmission range than *A* and *B*. Both *A* and *B* can receive messages from *S* directly, but only *A* can reply to *S* directly. *B* can reply to *S* via *A* indirectly. This paper names the adjacent users of WMNs in communication using the symmetric model if they communicate each other directly such as Users (*S*, *A*) and Users (*A*, *B*) in Figure 1. The user names its partner as the regular-neighbor if they can communicate in the symmetric model. This paper names the adjacent users of WMNs in communication using the asymmetric model if the user can communicate with its partner directly but the partner can only communicate with it via another user indirectly, such as Users (*S*, *B*) in Figure 1. The user names its partner as semi-neighbor if it communicates with its partner directly but the partner can only communicate with it via another user indirectly. The partner names the user as its rev-semi-neighbor. For example, User *B* is User *S*'s semi-neighbor and User *S* is User *B*'s rev-semi-neighbor in Figure 1. The WMN includes both symmetric and asymmetric models for each adjacent user, while the wireless ad hoc network only provides a symmetric model. The WMNs will enhance or provide more functionality based on communication in the asymmetric model.

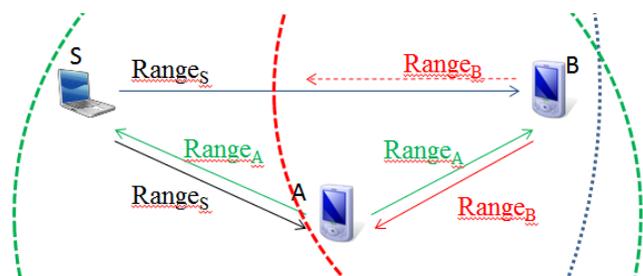


Figure 1. Scenario of communication in symmetric and asymmetric models.

In general, the transmission range of the access point is generally larger or equal to that of the wireless client. Access point not only serve wireless client directly as the traditional wireless network, but also serves wireless clients that can only communicate with it via some other wireless clients indirectly with asymmetric communication model. The wireless client in WMNs functions as both the client and router to made networks work well. The coverage of WMNs will be enhanced by this feature. By this case, the traditional routing protocols are not suitable for WMNs to generate the routing path between access point and wireless client. Some researches [2][5] [12][13][16][18] provide routing protocols with well consideration for this kind of networks. However, these protocols focus only on the efficiency and effectiveness. In wireless metropolitan networks, all data transmissions are usually via wireless transmission. It made eavesdropping, replace and modified message are easy occurred. Wireless metropolitan networks are also vulnerable to several kinds of attacks such as the Sybil attack [10], Rushing Attack [8], and etc... .

The routing protocol will fail to establish a corrected rout because several kinds of attacks corrupt the transmitting data. Secure routing protocols will straighten out these threats. Secure routing protocols have to guarantee data integrity and confidentiality and ensure the data will reach the correct destination. Several secure routing protocols [1][3][6][7][11][14] for wireless ad hoc and sensor networks provide mechanisms that resist attacks and guarantee that the destination will receive the correct transmitted data. These secure routing protocols consider only the symmetric communication model. The secure routing protocol of WMNs should consider both symmetric and asymmetric communication models to enhance the efficiency of WMNs.

Although the routing security protocol provides some security functionalities, the adversary will collect network traffic to analyze user behavior. The adversary may invade the user's privacy and hurt the user's safety. Some researches [1][4][14][16] proposed anonymous routing protocols to preserve privacy. Some studies have also considered anonymous data transmission to prevent the adversary from tracing messages to discover the sender. These anonymous routing protocols also only consider the symmetric communication model.

The communication of wireless client should consider both symmetric and asymmetric models to enhance the efficiency of WMNs. However, previous secure [7][7][9][11][15][17] or anonymous [1][4][14][16] routing protocols cannot work the communication in asymmetric model. Some attacks such as the Sybil attack and Rushing attack occur easily in asymmetric communication model. This paper proposes an anonymous secure routing protocol

for WMNs. The protocol based on proposed concepts will generate an efficient anonymous and secure routing path for the access point and its wireless clients based on symmetric and asymmetric models. This paper first provides the *neighbor discovery concept* for each user to discover its regular-neighbor, semi-neighbor and rev-semi-neighbor in Section 2. Each user in WMNs will use the proposed *neighbor discovery concept* to authenticate its neighbors and establish shared keys with them. The authentication and shared keys are essential to provide reliable data dissemination and ensure data confidentiality and integrity. Section 3 proposes a concept of anonymous and secure routing protocol. The protocol based on the proposed concepts considers when users cannot connect to access point directly, they can perform this protocol to establish an anonymous and secure route to the access point and obtain internet service securely. Access point can serve more users and increase the coverage of WMNs base on the proposed concept. Some simulation results and discussions are included in Section 4. Section 5 makes presents conclusions.

There are two main roles in the WMN: access point and wireless client. Access point integrates different communication protocols and provides internet service for the wireless client in WMNs. Wireless client is the user of WMNs. Access point and wireless client hold public/private key and a broadcast key that use to protect messages when broadcasting to their neighbors. Table 1 defines the notations of the proposed concepts.

TABLE I. NOTATIONS

Notations	Means
$PK_i/SK_i$	The public/private key of role $i$ .
$K_i^b$	The broadcast key of role $i$
$K_{ij}$	The shared secret key of role $i$ and role $j$
$N_i$	A random nonce of role $i$
$Sign_i(M)$	An unrecovered signature of message $M$ that signed by role $i$
$p$	A large prime number.
$g$	A generator of $Z_p^*$
$MAC(K, M)$	Message Authentication Code of message $M$ using key $K$
$E(K, M)$	Encrypt message $M$ using key $K$
$NL_i$	The neighbor list of role $i$ .
$NCL_i$	The neighbor candidate list of role $i$ .
$H()$	A hash function.
$Sign_i(*)$	A unrecovered signature of message before the signature
$MAC(K, *)$	Message Authentication Code of message before it using key $K$
$H(*)$	A hash for message before it.

## 2 Neighbor discovery concept

This section proposes the neighbor discovery concept in wireless metropolitan networks with asymmetric

communication consideration. Each user maintains two kinds of neighbors; one is regular-neighbor and the other is semi-neighbor. The user named a regular-neighbor of the request user, which receives the neighbor discovery message from the request user directly and can authenticate with the request user directly. The user named semi-neighbor of the request user, which receives neighbor discovery message directly but can only authenticate the request user via request user's other regular-neighbors or semi-neighbors indirectly. The user will be named a rev-semi-neighbor of its semi-neighbor. For example, in Figure 1, if  $S$  is the request user,  $A$  authenticates with  $S$  directly but  $B$  can only authenticate  $S$  via  $A$ .  $A$  is a regular-neighbor of  $S$ ,  $B$  is a semi-neighbor of  $S$  and  $S$  is a rev-semi-neighbor of  $B$ .

In the proposed neighbor discovery concept, the user first performs the *regular-neighbor discovery phase* and then performs the *semi-neighbor discovery phase* to discover its regular-neighbors and semi-neighbors. User cannot communicate with its rev-semi-neighbor directly. User should perform the *data forwarding to rev-semi-neighbor method* (as Subsection 2.3) to communicate with its rev-semi-neighbors while it communicates with its regular-neighbors and semi-neighbors directly.

## 2.1 Regular-neighbor discovery phase

Each user and access point in the wireless metropolitan network performs the *regular-neighbor discovery phase* to discover their regular-neighbors. In the scenario of Figure 1, User  $S$  first generates Neighbor discovery message  $T^S_1$ .

$$T^S_1 = \{ID_S || N_S || g^{r_S} \bmod p || \text{Sign}_S(*)\}.$$

Users such as User  $A$  and User  $B$  verify Message  $T^S_1$  and generate the reply message. User  $A$  records  $S$  in its Neighbor Candidate List  $NCL_A$ . User  $A$  chooses a random number  $r_A$  and computes  $g^{r_A} \bmod p$ . User  $A$  computes the shared secret key  $K_{SA} = (g^{r_S})^{r_A} \bmod p$ . Then User  $A$  replies message  $T^A_2$  to User  $S$ .

$$T^A_2 = \{ID_S || ID_A || g^{r_A} \bmod p || \text{Sign}_A(H(ID_S || ID_A || K_{SA}))\}$$

User  $B$  replies a message  $T^B_2$  as  $T^A_2$ , but User  $S$  cannot receive  $T^B_2$  because  $S$  is out of User  $B$ 's transmission range. User  $S$  computes  $K_{AS} = (g^{r_A})^{r_S} \bmod p$  and verifies  $T^A_2$ . User  $S$  records User  $A$  as regular-neighbor in Neighbor List  $NL_S$ . User  $S$  replies message  $T^S_3$  to User  $A$ .

$$T^S_3 = \{ID_S || E(K_{AS}, K^b_S || H(K^b_S)) || \text{Sign}_S(ID_S || ID_A || N_S)\}$$

User  $A$  records  $\text{Sign}_S(ID_S || ID_A || N_S)$  and removes  $S$  from  $NCL_A$  after verifies and decrypts  $T^S_3$ . After above

procedures, each user will recognize its regular-neighbor after the *regular-neighbor discovery phase*. It also gets a shared secret key with each regular-neighbor and each regular-neighbor's broadcast key.

## 2.2 Semi-neighbor discovery phase

If User's neighbor candidate list is not empty after *regular-neighbor discovery phase*, it performs the *semi-neighbor discovery phase* to discover the semi-neighbors from the neighbor candidate list. In the scenario of Figure 1,  $A$  recognizes  $B$  and  $S$  as its regular-neighbors and gets their broadcast keys  $\{K^b_B, K^b_S\}$  and shared secret keys  $\{K_{AB}, K_{AS}\}$  after  $A$  performs the *regular-neighbor discovery phase*.  $B$  recognizes  $A$  as its regular-neighbor and get  $A$ 's broadcast key  $K^b_A$  and shared secret key  $K_{AB}$ , but  $S$  is still keep in  $B$ 's neighbor candidate list  $NCL_B$  after  $S$  and  $B$  perform the *regular-neighbor discovery phase*. To authenticate  $S$ ,  $B$  broadcasts the message  $T^B_4$ .

$$T^B_4 = \{ID_B || NCL_B || \text{MAC}(K^b_B, ID_B || NCL_B)\}$$

User adjuncts to User  $B$  such as User  $A$  verifies  $T^B_4$  and generates the reply message  $T^A_5$  to  $B$ . User  $A$  computes the common neighbor list  $NL_{B,A}$ .

$$NL_{B,A} = NCL_B \cap NL_A.$$

If  $NL_{B,A}$  is not empty, User  $A$  sets the  $\text{SignList}_{B,A} = \{\text{Sign}_j(ID_j || ID_A || N_j) | \forall j \in NL_{B,A}\}$  and replies message  $T^A_5$  to  $B$ .

$$T^A_5 = \{ID_A || NL_{B,A} || \text{SignList}_{B,A} || \text{MAC}(K_{AB}, NL_{B,A} || \text{SignList}_{B,A})\}$$

User  $B$  replies message  $T^B_6$  to User  $S$  via User  $A$  after verifies message  $T^A_5$ .

$$T^B_6 = \{ID_S || ID_B || g^{r_B} \bmod p || \text{Sign}_B(H(ID_B || ID_S || K_{BS}))\}$$

User  $S$  computes shared secret key  $K_{BS}$  via Diffie-Hellman key exchange and record User  $B$  as semi-neighbor in  $NL_S$  after verifies message  $T^B_6$ . User  $S$  replies message  $T^S_7$  to User  $B$ .

$$T^S_7 = \{ID_S || E(K_{BS}, K^b_S || H(K^b_S)) || \text{Sign}_S(ID_S || ID_B || N_S)\}$$

User  $B$  obtains the broadcast key  $K^b_S$  and records  $\text{Sign}_S(ID_S || ID_B || N_S)$ . User  $B$  records User  $S$  as rev-semi-neighbor and User  $A$  as the corresponding common neighbor in  $NL_B$ . After User  $B$  authenticates User  $S$ , User  $B$  re-computes  $NL_{B,j} = NCL_j \cap NL_B$  for each User  $j$  in  $NL_B$ . If  $NL_{B,j}$  is not empty, User  $B$  notifies User  $j$  that they have common neighbors via send the message form as message  $T^A_5$ .

### 2.3 Data forwarding to rev-semi-neighbor method

This subsection proposes *Data forwarding to rev-semi-neighbor method*. When User  $i$  tries to forward message  $m$  to its rev-semi-neighbor User  $j$ . User  $i$  forward  $\{ID_k||E(K_{ik}, ID_j||m||H(*))\}$  to their recognized neighbor User  $k$  which is maintained in  $NL_i$  at the *semi-neighbor discovery phase*. User  $k$  keeps forward the message decrypts and  $\{ID_j||m||H(*)\}$  to User  $j$  after User  $k$  decrypt and verified the message. If User  $j$  is User  $k$ 's rev-semi neighbors, User  $k$  forward message as User  $i$ 's form. Otherwise, User  $k$  sends  $\{ID_j||m||H(*)\}$  to User  $j$  directly.

## 3 A concept of anonymous secure routing protocol

If user in WMNs would like to access Internet, it should first connect to the access point. User that is a regular-neighbor of the access point can communicate with the access point directly to access the Internet. The user that cannot connect to the access point directly must establish a route to the access point to access Internet. It is important to guarantee the data can reach the correct destination and the received data is confident and correct. To protect the user privacy is also an important issue in the connection with the access point. The user's communication behavior cannot be learned by an adversary. This section provides a concept of anonymous and secure routing protocol to establish a route that achieves authentication, confidentiality, integrity and anonymity. The user that cannot connect with the access point directly performs the protocol based on the proposed concept to establish an anonymous and secure route to the access point after performing neighbor discovery concept as Section 2. The proposed concept includes *anonymous route request phase* and *anonymous route reply phase*. The user will establish an anonymous and secure route to access point after performing the protocol based the proposed concept detailed in Subsections 3.1 and 3.2.

### 3.1 Anonymous route request phase

User  $S$  performs the *anonymous route request phase* to discover an anonymous and secure route to the target destination, Access point  $D$ . Source  $S$  first generate the  $ARREQ_S$ .

$$ARREQ_S = \{E(K_S^b, TPK||E(PK_D, ID_D||TSK||PL_S)||Route\_Sec_S||H(*)\}$$

$ARREQ_S$  is formed by three parts. The first part is the  $TPK$ , the temporal public key which is generated by User  $S$  only for this session. User  $S$  also generates the corresponding temporal private key  $TSK$ . The second part is  $E(PK_D,$

$ID_D||TSK||PL_S)$ . User  $S$  uses destination's public key to encrypts destination's real identity  $ID_D$ , the temporal private key  $TSK$  and the  $PL_S$  which is the length of random padding bit  $Padding_S$ . The third part is the  $Route\_Sec_S = E(TPK, ID_S||P_S||K_{SD}||Route\_Sec_0||Sign_S(H(*)))$ . User  $S$  uses  $TPK$  to encrypts User  $S$ ' real identity, the pseudonym  $P_S$ , session key  $K_{SD}$  which is randomly generated by User  $S$ ,  $Route\_Sec_0 = \{ID_D||Padding_S\}$ . User  $S$  hashes above three parts and encrypts these with its broadcast key  $K_S^b$ . Finally User  $S$  broadcasts  $ARREQ_S$  and records  $\{TPK||ID_S||ID_D||P_S||K_{SD}\}$  secretly.

User  $j$  received the  $ARREQ_i$  from its neighbor User  $i$ , User  $j$  first decrypts and check the freshness of the  $ARREQ_i$  by compare the  $TPK$ . Then User  $j$  uses its private key  $SK_j$  to decrypts " $E(PK_D, ID_D||TSK||PL_S)$ " and checks the destination is itself or not. If User  $j$  isn't the destination, he generates pseudonym  $P_j$  and corresponding session key  $K_{P_j}$ . User  $j$  records  $\{TPK||ID_i||P_j||K_{P_j}\}$  and generates  $Route\_Sec_j = E(TPK, ID_j||P_j||K_{P_j}||Route\_Sec_i||Sign_j(H(*)))$  where  $P_j$  and  $K_{P_j}$  are pseudonym and corresponding session key. User  $j$  broadcasts the  $ARREQ_j = \{E(K_j^b, TPK||E(PK_D, ID_D||TSK||PL_S)||Route\_Sec_j||H(*)\}$ . When the target destination  $D$  receives the  $ARREQ_n$  from its neighbor User  $n$ , it first decrypts " $E(PK_D, ID_D||TSK||PL_S)$ " to retrieve the  $TSK$  and then uses  $TSK$  to decrypt  $Route\_Sec_j$ . User  $D$  decrypts  $Route\_Sec_i$  layer by layer to get User  $i$ 's pseudonym  $P_i$  and its corresponding session key  $K_{P_i}$  until retrieve the  $Route\_Sec_0$ . User  $D$  records  $P_i$  and its  $K_{P_i}$  in  $RouteList(=\{P_1||K_{P_1}||P_2||K_{P_2}||\dots||P_n||K_{P_n}\})$ , where  $P_1$  is the pseudonym of the Source  $S$ 's neighbor and  $P_n$  is the pseudonym of Destination  $D$ 's neighbor. Finally, User  $D$  launches *anonymous route reply phase* based on  $RouteList$ .

### 3.2 Anonymous route reply phase

The target destination, Access point  $D$ , performs the *anonymous route reply phase* to confirm the route with User  $S$  based on  $RouteList$  which has discovered at Subsection 3.1. User  $D$  generates the pseudonym  $P_D$  and records  $\{TPK||ID_D||ID_S||P_D||K_{SD}\}$  secretly. User  $D$  generates the  $ARREP_S$  and iteratively generates the route reply message for each User  $i$  on the route as  $ARREP_i$  starting from the neighbor  $P_1$  of Source  $S$  to the neighbor  $P_n$  of Destination  $D$  based on the order of  $RouteList$ , where  $ARREP_0 = ARREP_S$ .

$$ARREP_S = \{P_S||E(K_{SD}, TPK||P_D||PL_D||RouteList||Padding_D)||MAC(K_{SD}, *)\}$$

$$ARREP_i = \{P_i||E(K_{P_i}, TPK||ARREP_{i-1})||MAC(K_{P_i}, *)\}$$

Finally, User  $D$  broadcasts  $ARREP_n = \{P_n||E(K_{P_n}, TPK||ARREP_{n-1})||MAC(K_{P_n}, *)\}$ .

User  $j$  retrieves  $K_{P_j}$  from its record  $\{TPK\|ID_i\|P_j\|K_{P_j}\}$  based on  $P_j$  to verify and decrypts  $ARREP_j$  which is broadcasted by its neighbor User  $k$ . User  $j$  learns it is the source user, if  $TPK$  of decrypted  $ARREP_j$  is its belongings. User  $j$  retrieves and records the pseudonym  $P_D$  and  $RouteList$  from decrypted  $ARREP_j (= ARREP_S)$ . If  $TPK$  in  $ARREP_j$  is not User  $j$ 's belongings, User  $j$  updates  $\{TPK\|ID_i\|P_j\|K_{P_j}\}$  to  $\{TPK\|ID_i\|ID_k\|P_j\|K_{P_j}\}$  and forward the  $ARREP_i$  to User  $i$ .

## 4 Performance evaluation

The proposed concept of anonymous secure routing protocol in Section 3 establishes a route based on the proposed *neighbor discovery concept* of Section 2. Each user performs the proposed *neighbor discovery concept* to discover all neighbors considering symmetric and asymmetric communication models. The anonymous secure routing protocol based on the proposed concept establishes a route considering both symmetric and asymmetric communication models while most prior secure/anonymous routing protocols for WMNs did not consider the asymmetric communication model. This section describes the network performance improvement of the proposed concept. Later subsections compare the neighbor discovery rate, success rate for route establishment and the average route hop count between the symmetric model (*i.e.* most of prior secure/anonymous routing protocols for WMNs) and the proposed concept which considers both symmetric and asymmetric communication models. These simulations set the network area as 2 kilometer  $\times$  2 kilometer. Users were distributed randomly in the network. The user destinies of simulations are from 1 user / (80 meter  $\times$  80 meter) to 1 user / (120meter  $\times$  120meter). The simulations classified the user into two kinds: a user with a larger communication range 250 meters called power user and a user with a smaller communication range 125 meters called a normal user.

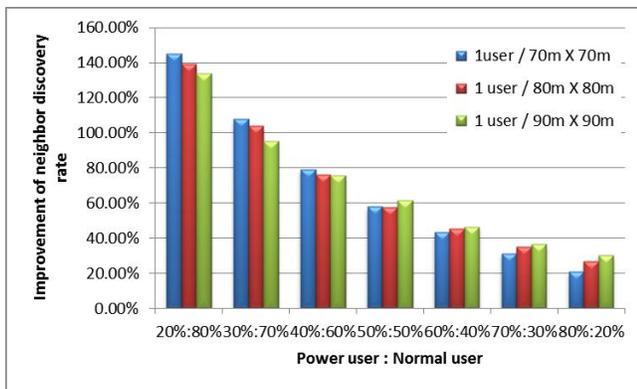


Figure 2. Improvement of neighbor discovery rate of Normal user

### 4.1 Improvement of neighbor discovery rate

This subsection discusses the neighbor discovery rate improvement by the proposed concept considering both symmetric and asymmetric communication. The simulation measures how many neighbors are discovered by each user in the symmetric model only v.s. the proposed concept. Figure 2 shows the neighbor discovery rate improvement by the proposed concept. If 80% of the users are normal users, the neighbor discovery rate improvement is at least 130%. If the number of power user is larger, *i.e.* more users hold larger communication range, the improvement benefits are smaller. However, the neighbor discovery rate of the proposed concept provides at least 20% improvement when the percentage of normal user decreases to 20%.

### 4.2 The average route hop count and route establishment success rate

Section 3 proposed a concept of anonymous routing protocol. This concept can be applied to both symmetric and asymmetric models. The user performs the proposed concept to establish a route to the access point with regular-neighbors and semi-neighbor discovered using the *neighbor discovery phase* as detailed in Section 2. This subsection evaluates the efficiency of Data forwarding. These simulations set the hop counts at 10 and 15. Each simulation chooses 20% normal users and 20% power users randomly to establish a route to the Access point. The access point is located at the center of the network. Figure 3 and Figure 4 illustrate comparisons with the average number of hops. The proposed concept establishes a shorter route. The average length of the route established by the proposed concept is 85%~90% of the average route length of routes considering symmetric model only. Even when the number of normal users decreases the proposed concept still finds shorter routes.

Figure 5 and Figure 6 illustrate the comparisons of successful rate with route establishment between different hop count limitations. More users will establish an anonymous secure route to Access point using the proposed concept in comparison with the symmetric model with different hop count limitations. The results demonstrate that the proposed concept establishes more and shorter routes in WMNs.

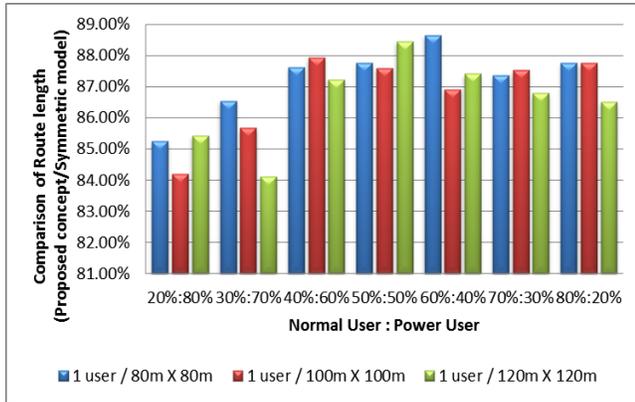


Figure 3. The average hop count of route (hop count = 10)

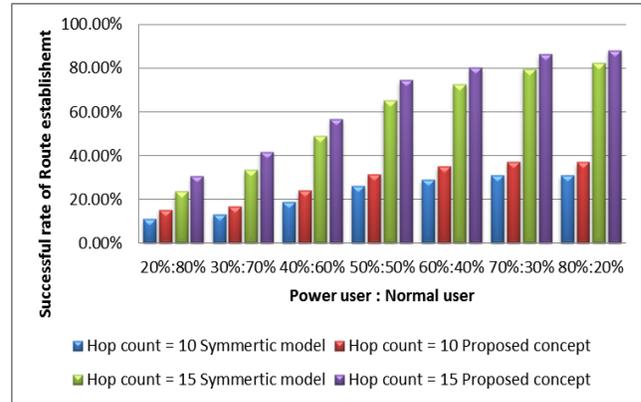


Figure 6. Comparison of route establishment success rate (User Density = 1 user / 110m x 110m)

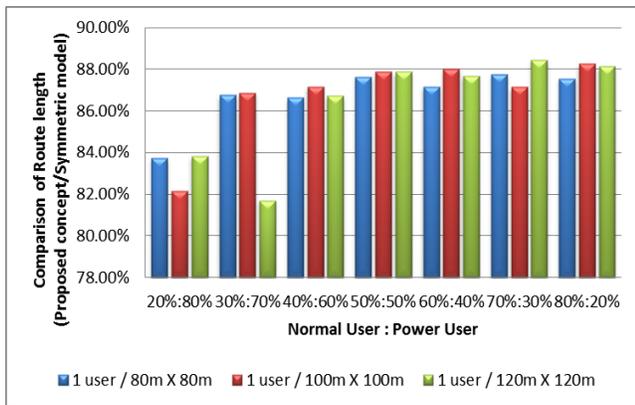


Figure 4. The average hop count of route (hop count = 15)

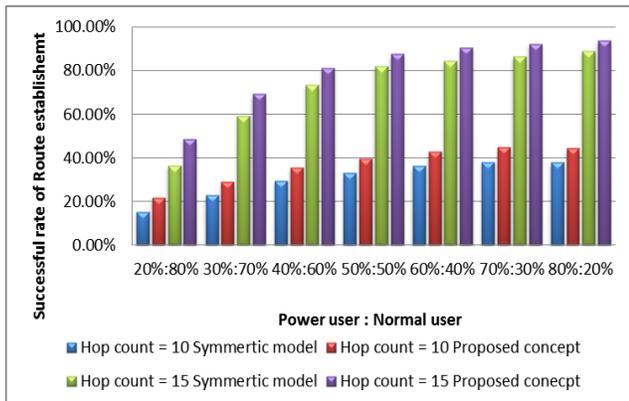


Figure 5. Comparison of route establishment success rate (User Density = 1 user / 90m x 90m)

## 5 Conclusions

Anonymity is a very important feature of Wireless Metropolitan Networks security. This paper proposed concepts of anonymous secure routing protocol with asymmetric communication consideration. The proposed concept ensures both the anonymity and security of the routing protocol. This paper firstly proposed a concept of neighbor discovery beyond symmetric and asymmetric models. Each user will identify as many neighbors as possible in its communication range via the proposed neighbor discovery concept. This allows the user to obtain more resources from his neighbors. A wireless device out of the wireless transmission range of the access point may perform the routing protocol based on the proposed concepts to discover a secure route to the access point anonymously. Therefore, more users can obtain the network service and protect their privacy. The proposed concept establishes a shorter route with a higher route establishment success rate because it considers both symmetric and asymmetric models. The anonymous routing protocol based on the proposed concepts is more efficient and suitable for wireless metropolitan networks.

## Acknowledgements

This work was partially supported by the National Science Council, Taiwan, under grants no. NSC99-2221-E-032-048.

## Reference

- [1] Azzedine Boukerche, Khalil El-Khatib, Li Xu, Larry Korba, An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks, Computer Communications, Volume 28, Issue 10, Performance issues of Wireless LANs, PANs and ad hoc networks, 16 June 2005, Pages 1193-1203,
- [2] Yigal Bejerano, Seung-Jae Han, Amit Kumar, Efficient load-balancing routing for wireless mesh networks, Computer Networks, Volume 51, Issue 10, 11 July 2007, Pages 2450-2466
- [3] Jing Deng, Richard Han, Shivakant Mishra, INSENS: Intrusion-tolerant routing for wireless sensor networks, Computer

- Communications, Volume 29, Issue 2, Dependable Wireless Sensor Networks, 10 January 2006, Pages 216-230,
- [4] Ying Dong, Tat Wing Chim, Victor O.K. Li, S.M. Yiu, C.K. Hui, ARMR: Anonymous routing protocol with multiple routes for communications in mobile ad hoc networks, Ad Hoc Networks, Volume 7, Issue 8, Privacy and Security in Wireless Sensor and Ad Hoc Networks, November 2009, Pages 1536-1550,
  - [5] Jakob Eriksson; Michalis Faloutsos; Srikanth V. Krishnamurthy; , "DART: Dynamic Address RouTing for Scalable Ad Hoc and Mesh Networks," Networking, IEEE/ACM Transactions on , vol.15, no.1, pp.119-132, Feb. 2007
  - [6] Tingyao Jiang, Qinghua Li, and Youlin Ruan. 2004. Secure Dynamic Source Routing Protocol. In Proceedings of the The Fourth International Conference on Computer and Information Technology (CIT '04), Washington, DC, USA, Pages 528-533.
  - [7] Frank Kargl, Alfred Geis, Stefan Schlott, and Michael Weber. 2005. Secure Dynamic Source Routing. In Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences - Volume 09 (HICSS '05), Vol. 9, Washington, DC, USA,
  - [8] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. 2003. Rushing attacks and defense in wireless ad hoc network routing protocols. In Proceedings of the 2nd ACM workshop on Wireless security (WiSe '03), New York, NY, USA, Pages 30-40.
  - [9] Jihye Kim, Gene Tsudik, SRDP: Secure route discovery for dynamic source routing in MANETs, Ad Hoc Networks, Volume 7, Issue 6, August 2009, Pages 1097-1109.
  - [10] L.A.Martucci, A.Zuccato, S.Fischer-Hübner. Identity Deployment and Management in Wireless Mesh Networks. In: The Future of Identity in the Information Society - Proceedings of the 3rd IFIP WG 9.2, 9.6/11.6, 11.7/FIDIS International Summer School. Springer. Aug. 2007. Karlstad, Sweden. Pages.223-234.
  - [11] Rosa Mavropodi, Panayiotis Kotzanikolaou, Christos Douligeris, SecMR - a secure multipath routing protocol for ad hoc networks, Ad Hoc Networks, Volume 5, Issue 1, January 2007, Pages 87-99,
  - [12] Krichene, N.; Boudriga, N.; , "Intrusion Tolerant Routing for Mesh Networks," 2007 IFIP International Conference on Wireless and Optical Communications Networks, 2-4 July 2007, Singapore, Pages 1-7.
  - [13] Nagesh S. Nandiraju; Deepti S. Nandiraju; Dharma P. Agrawal; , "Multipath Routing in Wireless Mesh Networks," 2006 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), Vancouver , Canada, Pages 741-746, 9-12 Oct. 2006.
  - [14] Ronggong Song, Larry Korba, and George Yee. 2005. AnonDSR: efficient anonymous dynamic source routing for mobile ad-hoc networks. In Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks (SASN '05), New York, NY, USA, Pages 33-42.
  - [15] Ming-Yang Su, WARP: A wormhole-avoidance routing protocol by anomaly detection in mobile ad hoc networks, Computers & Security, Volume 29, Issue 2, March 2010, Pages 208-224,
  - [16] Zhiguo Wan; Kui Ren; Bo Zhu; Preneel, B.; Ming Gu; , "Anonymous User Communication for Privacy Protection in Wireless Metropolitan Mesh Networks," IEEE Transactions on Vehicular Technology , vol.59, no.2, Pages.519-532, Feb. 2010
  - [17] Jianliang Zheng, Myung J. Lee, A resource-efficient and scalable wireless mesh routing protocol, Ad Hoc Networks, Volume 5, Issue 6, August 2007, Pages 704-718.

# A First Step towards Privacy Leakage Diagnosis and Protection

Shinsaku Kiyomoto<sup>1</sup>, and Toshiaki Tanaka<sup>1</sup>

<sup>1</sup>KDDI R & D Laboratories Inc.

2-1-15 Ohara Fujimino, Saitama, 356-8502, Japan

**Abstract**— *In this paper, we present a first step for designing a privacy leakage diagnosis and protection system using two privacy definitions and a new definition, and then evaluate a prototype program. The diagnosis is based on major notions of privacy:  $k$ -anonymity and  $(c, l)$ -diversity. Furthermore, the diagnosis include another method that analyze sensitivity of each attribute values. The prototype program realizes a computation time of less than 1 ms for the diagnosis and updating of data. Thus, it provides a privacy-leakage level within a feasible computation time.*

**Keywords:** Privacy, Privacy-Leakage, Diagnosis, Protection, Security

## 1. Introduction

Privacy is an important matter to be considered in user-oriented services such as location-based services and recommendation services. In user-oriented services, a user transmits his or her privacy information in order to receive personalized services or obtain appropriate information. There is a tradeoff between the leakage of user privacy and the precision of information for the service. For example, in location-based services, a user obtains more precise information if the user transmits more detailed location information. The user has to weigh the amount of leaked information against the benefits provided by the service. Generally, a user makes a decision about the amount of leaked information depending on the trustworthiness of the service provider. If the service provider is trusted, users are willing to give detailed private information; however, if the level of trust in the service provider is low, users want to transmit the minimum amount of information in order to protect their privacy.

As Figure 1 shows, there are three steps for protecting user's privacy when sending data: privacy leakage diagnosis, judgment, and data modification. The judgment process determines whether the data should be sent, modified, or terminated. The modification process modifies data to reduce the amount of privacy leakage using privacy protection methods [1], [2]. The privacy leakage diagnosis step evaluates privacy-leakage from the send data. The diagnosis calculates the difference between the data being sent and data of other users for the same service. That is, the diagnosis evaluates

the values of privacy-leakage compared with background data.

This paper mainly focuses on privacy leakage diagnosis of user send data. The problem associated with evaluating privacy leakage of data is that the amount of leaked information sometimes depends on background information such as data from other users. A user is identified and privacy is leaked in cases where only the user has a pair of attribute values; on the other hand, one user is not identified in the situation where many users have the same attribute values. Existing research has provided privacy-leakage "detection" methods; however, it is not applicable to a quantitative analysis of privacy leakage. It is useful for users if the diagnosis system is able to provide several privacy leakage levels as analysis results. Thus, a method of quantitative diagnosis for privacy leakage that considers background information is needed to assist the user to make an informed decision. Furthermore, the diagnosis method has to calculate diagnosis results from hashed attribute values to avoid information leakage. Simple keyword-based pattern-matching methods are not applicable for the diagnosis.

In this paper, we present a diagnosis method for privacy leakage using privacy definitions, implement it on a prototype program, and evaluate performance of the program. The method conducts a quantitative analysis of privacy leakage and the program is a key component to construct a privacy leakage diagnosis and protection system. The rest of the paper is organized as follows: Section 2 provides the privacy definitions that are used in our system. We explain the diagnosis system in Section 3, and then evaluation results of the prototype system are shown in Section 4. We discuss remaining issues on privacy leakage diagnosis and protection in Section 5. Related work is summarized in Section 6. Finally, Section 7 provides our conclusion.

## 2. Notion of Privacy

Samarati and Sweeney [3], [4], [5] proposed a primary definition of privacy that is applicable to generalization methods. A data set is said to have  $k$ -anonymity if each record is indistinguishable from at least  $k - 1$  other records with respect to certain identifying attributes called *quasi-identifiers* [6]. In other words, at least  $k$  records must exist

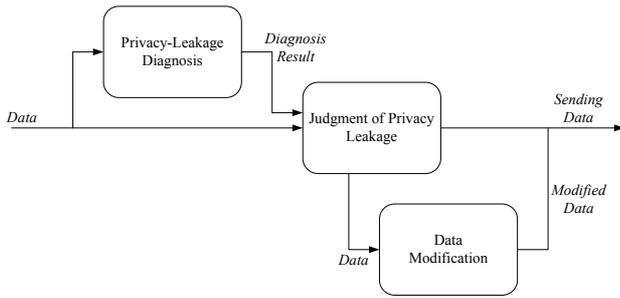


Fig. 1: Privacy Leakage Diagnosis and Protection Procedure

in the data set for each combination of the identifying attributes. Clearly any generalization algorithm that converts a database into one with  $k$ -anonymity involves a loss of information in that database. The definition of  $k$ -anonymity has been widely studied because of its conceptual simplicity [1], [2].

A data set is said to have  $k$ -anonymity if each record is indistinguishable from at least  $k - 1$  other records with respect to certain identifying attributes called *quasi-identifiers* [6]. In other words, at least  $k$  records must exist in the data set for each combination of attributes. More precisely, suppose that a database table  $T$  has  $m$  records and  $n$  attributes  $\{A_1, \dots, A_n\}$ . Each record  $\mathbf{a}^i = (a_1^i, \dots, a_n^i)$  can thus be considered as an  $n$ -tuple of attribute values, where  $a_j^i$  is the value of attribute  $A_j$  in record  $\mathbf{a}^i$ . The database table  $T$  itself can thus be regarded as the set of records  $T = \{\mathbf{a}^i : 1 \leq i \leq m\}$ .

**Definition 1. ( $k$ -Anonymity)** A database table  $T$  is said to have  $k$ -anonymity if and only if each  $n$ -tuple of attribute values  $\mathbf{a} \in T$  appears at least  $k$  times in  $T$ .

The definition of  $k$ -anonymity does not on its own encompass the concept of an adversary who has background knowledge that can help them distinguish records [7]. As a result, several extensions to the basic idea have been proposed, including  $l$ -diversity and recursive  $(c, l)$ -diversity, as well as other suggestions in [8], [9], [10].

The two definitions,  $l$ -diversity and recursive  $(c, l)$ -diversity, are used to evaluate sensitive attributes in a table  $T$ . The definitions are as follows;

**Definition 2. ( $l$ -Diversity)** A database table is said to have  $l$ -diversity if all groups of data that have the same quasi-identifiers contain at least  $l$  values for each sensitive attribute.

**Definition 3. (Recursive  $(c, l)$ -Diversity)** A database table is said to have recursive  $(c, l)$ -diversity if all groups of

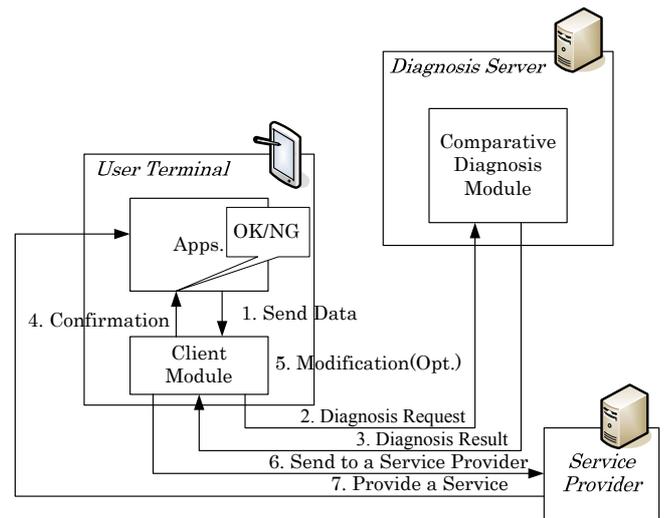


Fig. 2: System Overview

data that have the same quasi-identifiers satisfy the condition  $r_{max} = c(\sum_{i \setminus max} r_i)$ , where  $r_i$  is the number of data sets having the same sensitive attributes, and  $r_{max}$  is the maximum number of all the data sets.

### 3. Privacy-Leakage Diagnosis

In this section, we explain overview of a privacy-leakage diagnosis and protection system that consists a client module on a user terminal and a diagnosis module on a diagnosis server, and then present details of the diagnosis method.

#### 3.1 Overview

Figure 2 shows an overview of the privacy-leakage diagnosis and protection system (PLDPS). The system consists of client modules on user terminals and a comparative diagnosis module on the diagnosis server. The procedure for the system is as follows:

- 1) An application program start to transmit data to a service provider. The data consist of attribute values requested by the service provider.
- 2) A client module of the PLDPS interrupts the data and transmits a diagnosis request to a diagnosis server. To minimize information leakage to the diagnosis server, each attribute value is replaced by its corresponding hash value.
- 3) The diagnosis server checks privacy leakage and transmits the result to the client module.
- 4) The client module returns the result to the application and confirms whether the original data are to be transmitted to the service provider. If the send data

need to be modified, the client module modifies the send data according to the diagnosis result.

- 5) If the application program allows to the data to be transmitted, the client module transmits the interrupted data to the service provider and send its hash values to the diagnosis server.
- 6) The service provider receives the data and provides his or her service to the user through the application program. The diagnosis server updates the database.

The user terminal transmits personal information (such as location information and personal logs) to a service provider and receives a personalized service. We assume that the data being transmitted consist of  $n$  attributes that is labeled non-sensitive, quasi-identifier, or sensitive by a pre-defined data format for each service. If no label is assigned for a attribute, the attribute is executed as a non-sensitive attribute. The client module interrupts the data being transmitted and derives  $n$  attributes from the data. The module calculates the hash values of all attributes and transmits the hashed attributes to the diagnosis program. The client module receives the diagnosis results from the diagnosis program and determines the privacy leakage level of the data using the results and the privacy policy of the user. Finally, the client program notifies the privacy leakage level and requests confirmation from the user. The details of the determination of the privacy leakage level are discussed in 5.2. Our diagnosis system is applicable to a multi-service platform; the client module transmits data with a service name in a multi-service platform and the diagnosis server prepares a database for each service.

### 3.2 Diagnosis Method

A diagnosis program receives data from client modules and stores it in a database. The database is used for the diagnosis of data and updates when new data are received. A diagnosis server is assumed to be honest and a trusted party for users. The diagnosis method has to calculate diagnosis results from hashed attribute values. Thus, a simple pattern-matching method based on some keywords is not suitable for the diagnosis.

We use two different diagnoses as follows; An anonymity-based diagnosis evaluates the value of the information for each group of records that has the same attribute values as the data. Thus, this diagnosis cannot distinguish which attribute value is the most sensitive. On the other hand, an attribute-based diagnosis evaluates the sensitivity of values for each attribute in the data. By combining these two diagnostic methods that are based on different approaches, the program obtains more precise results about privacy leakage. We describe the details of two diagnoses in the following subsections.

#### 3.2.1 Anonymity-based Diagnosis

In the system, we apply  $k$ -anonymity concepts to evaluation of the data being transmitted. An anonymity-based diagnosis is based on  $k$ -anonymity,  $l$ -diversity and recursive  $(c, l)$ -diversity definitions. First, the data being transmitted are separated into two parts: non-sensitive data and sensitive data. Then, the data are evaluated in the following steps:

- 1) A diagnosis program analyzing privacy leakage as a part of diagnosis module counts the number of data  $D$  that are stored in its database and have the same values of non-sensitive attributes as the non-sensitive attributes of the evaluation data. Then the program outputs  $k$  where the number is  $k - 1$ .
- 2) The diagnosis program selects the sensitive attributes of the data  $D$  and calculates  $l$  of  $l$ -diversity for each sensitive attribute by using the sensitive attributes of the evaluating data and the data  $D$ . Then, the program outputs a minimum value  $l_{min}$  of the  $l$  values. If non-sensitive attribute exists on the data, the program skips the steps 2) and 3).
- 3) The diagnosis program calculates  $c_i = r_i / (r_1 + r_2 + \dots + r_s - r_i)$  for each sensitive attribute, where  $r_1, r_2, \dots, r_s$  are the numbers of data in each group that have the same sensitive attribute and  $r_i$  is the number of data having the same sensitive attribute as the evaluating data. Then the program outputs a maximum value  $c_{max}$  of the  $c_i$  values.

The results  $k$ ,  $l_{min}$  and  $c_{max}$  are output as anonymity-based diagnosis values.

*k*-Anonymization. The diagnosis program can give useful information for modification of the data on the user terminal. If the program executes bottom-up  $k$ -anonymization using an attribute tree of hashed values, the program can obtain the generalization level of each attribute which denotes the number of bottom-up generalizations for each attribute, and sends it to the user terminal. The user terminal reduces privacy leakage satisfying  $k$ -anonymity, to generalize each attributes based on the suggested generalization level.

#### 3.2.2 Attribute-based Diagnosis

We use an index for the evaluation of privacy leakage from the data,  $\delta_i$  for each attribute. The index evaluates changes in the amount of information where the data are added to current data sets. The indices  $\delta_i$  for each attribute ( $1 \leq i \leq n$ ) are based on the entropy of the attribute value of a record. The index  $\delta_i$  is calculated as:

$$\delta_i = \alpha_i \left| \log \left( \frac{M_i}{m_i} \right) - \log \left( \frac{M_i - 1}{m_i - 1} \right) \right| (1 \leq i \leq n)$$

where,  $m_i$  is the number of records that have the same attribute value of the  $i$ th attribute in the evaluated data,  $M_i$  is the total number of records that have the  $i$ th attribute and  $\alpha_i$  is a weighted factor for the  $i$ th attribute. Note that we define  $\log((M_i - 1)/(m_i - 1)) = 0$  where  $m_i = 1$ . The results  $\delta_i$ s are output as the attribute-based diagnosis values.

*Modification of Attributes.* From results of the attribute-based diagnosis, the user can obtain information as to which attributes are more sensitive. One possible solution is that the user removes the sensitive attribute values from the sending data based on the diagnosis results

## 4. Prototype Module

We implement a prototype of the diagnosis module on a PC that has a 2.8 GHz Core 2 Duo processor and 2 GByte of memory.

### 4.1 Implementation

In the implementation, we first consider the performance of the diagnosis program. To respond on a real-time basis, the diagnosis program extracts all the data to physical memory to reduce the cost of loading the data. We configure a window size for the data and the oldest data are removed from the window when the window is filled with data from users. We also use an efficient data structure for storing records of user data and are thereby able to reduce the computation cost of the diagnosis. The data structure includes two types of data: ring buffer and red-black tree for sensitive and non-sensitive attributes, respectively. The ring buffer stores raw data records transmitted from users and keeps them along with the time they were received. The ring buffer is FIFO (First In, First Out); the oldest data are removed when the buffer is full. We can reduce the transaction time for adding one record to the top and removing one record from the end of the ring buffer. The red-black tree realizes  $O(\log(n))$  search operations for a node that has the same combination of attributes as the evaluated data.

*Red-Black Tree for Attribute Values.* A node of the red-black tree for non-sensitive data holds the number of records ( $k$  value) in a group that has the same attributes,  $l$  value of the group,  $r_i$  and  $m_i$  values of each attribute in the group. The program also has another table that stores  $M_i$ .

The procedure of updating the red-black tree is as follows:

- 1) The program searches for a group that has the same attributes as the new record on the red-black tree.
- 2) If it appears on the tree, the program calculates diagnosis values using the parameters on the tree. If not, the program calculates diagnosis values.

- 3) When the program receives the data finally sent to the service provider, the program updates values,  $k$ ,  $l$ ,  $r_i$  and  $n_i$  of the group or makes a new group on the red-black tree and stores the calculated values.
- 4) If an old record was removed from the ring buffer, the program updates the values of a corresponding group.

The program can compute diagnosis values from the parameters of the group, not from the data as a whole. Thus, the computational cost is reduced using the red-black tree.

## 4.2 Evaluation

In the first experiment, we randomly select one record from the data and use the record as transmitted data. We evaluate the transaction time for diagnosis computation of send data and updating data using seven types of data sets: 5 quasi-identifiers and 1 sensitive attribute, 10 quasi-identifiers and 1 sensitive attribute, 20 quasi-identifiers and 2 sensitive attributes, 30 quasi-identifiers and 3 sensitive attributes, 50 quasi-identifiers and 5 sensitive attributes, 100 quasi-identifiers and 10 sensitive attributes, 200 quasi-identifiers and 20 sensitive attributes. Figure 3 and Figure 4 respectively show transaction time for the diagnosis of send data and updating of the data sets, where the number of quasi-identifiers is  $n$  and the number of sensitive attribute is  $s$ . The transaction times are the average times of 1000 trials with the random selection. The transaction time of diagnosis computation is less than  $50 \mu\text{s}$  in the case of 100,000 records of the 100 quasi-identifiers and 10 sensitive attributes data sets. The transaction time for updating data is less than  $160 \mu\text{s}$  in the same case. The transaction times increase in proportion to the increase in the number of attributes, but are not sensitive to increases in the number of records due to our efficient implementation described in 4.1.

In the second experiment, we examined transaction time using data sets in [11]: 2-anonymized Adults Database, 2-anonymized Census-Income, and 2-anonymized LandsEnd. Table 1 shows results of the second experiment. The total transaction time would be feasible even if we consider transaction time for data exchange between the user terminal and the diagnosis server.

We evaluated memory usage of the system as shown in Table 2. The memory usage increases in proportion to the increase of the numbers of attributes and records. The system uses 1.2Gbyte of memory in the case of two million records.

## 5. Discussion

In this section, we discuss some remaining issues for PLDPS.

### 5.1 Parallelization and Database Size

Diagnosis requests from user terminals are concentrated on a diagnosis server in our architecture. Thus, we should

Table 1: Transaction Time for Real Data Sets

	No. of Attributes	No. of Records	Diagnosis (us)	Updating (us)
Adults Database	15	32,561	7.946	10.275
Census-Income	42	199,523	15.893	25.221
LandsEnd	8	4,591,581	6.075	12.672

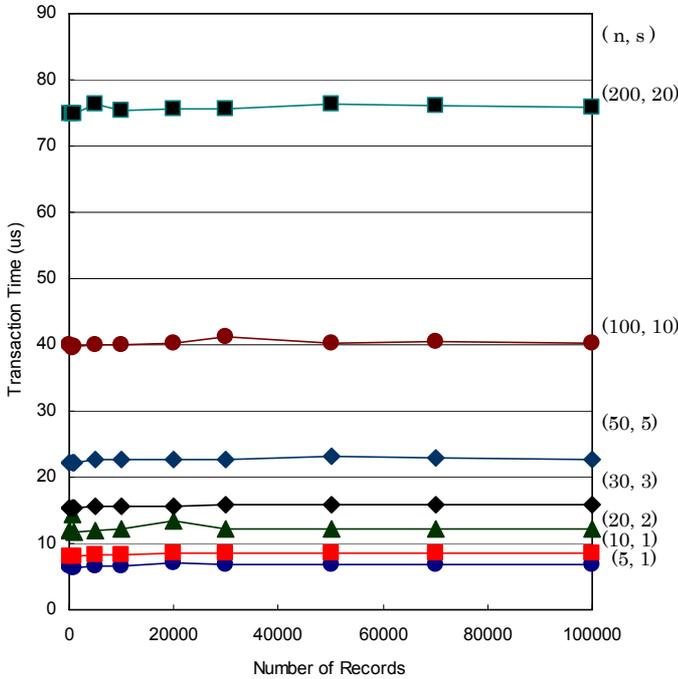


Fig. 3: Transaction Time of Diagnosis Computation

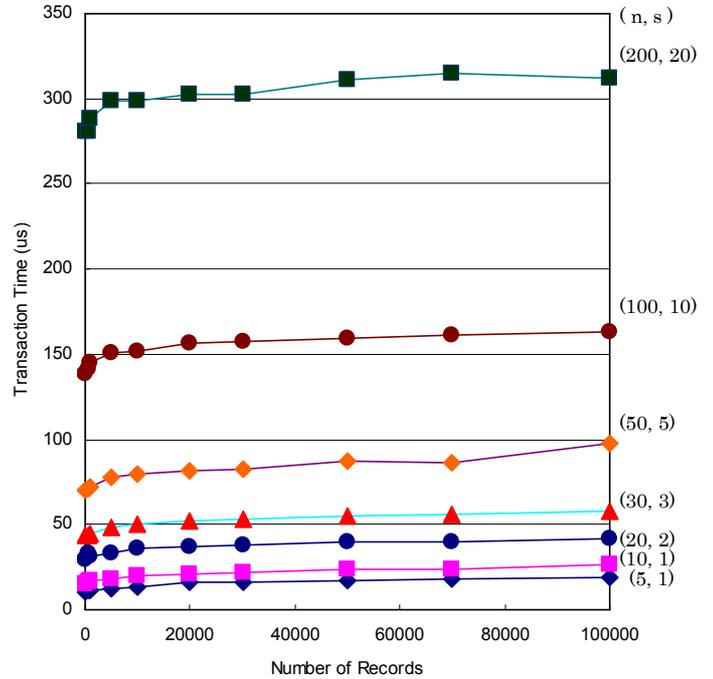


Fig. 4: Transaction Time for Updating Data

Table 2: Memory Usage

No. of Records	No. of Attributes	Memory Usage (MB)
1,000,000	10	641
1,000,000	20	1,157
1,000,000	30	1,707
1,500,000	10	1,123
2,000,000	10	1,233

consider a load-balancing method for the diagnosis server. One possible solution is to construct diagnosis servers that have a dedicated database for each service and the diagnosis requests are sent to each diagnosis server according to the service name of the send data. Another solution is to parallelize diagnosis servers that share a single DB and use a traffic load-balancing mechanism for sorting diagnosis requests. We will evaluate an upper limit for the number of simultaneous access to the diagnosis server and effectiveness of a load-balancing method in a future study.

Another issue for implementation is how to define the maximum size of a database for privacy leakage diagnosis.

The size depends on service and privacy requirements. As part of our ongoing research, we will examine what constitutes a suitable database size for real services.

### 5.2 Judgement and Modification

The next step in the privacy-leakage diagnosis and protection system is to determine the privacy-leakage level as for the judgment step. The client module should display a comprehensible privacy-leakage level to the user (instead of received diagnosis results), due to ease of user's judgment. The privacy-leakage level should be selected based on the diagnosis result and the privacy policy.

An example of a privacy policy is shown in Table 3. The parameter  $x_{max}$  is the maximum value of a diagnosis results  $xs$ , and  $x_{min}$  is the minimum value of  $xs$ . The parameters  $t_d$ ,  $t_k$  and  $t_c$  in the security policy are threshold values for anonymity-based diagnosis and attribute-based diagnosis. The anonymity-based diagnosis checks all the data that consist of some attributes and the attribute-based diagnosis verifies the sensitivity of each attribute; thus, we

Table 3: Example of Privacy Policy

PL Level	Condition of Diagnosis Result
Level 5	$k = 1, \delta_{max} \geq t_d$
Level 4	$k = 1, \delta_{max} < t_d$
Level 3	$k \leq t_k, l_{min} \leq t_l, c_{max} \leq t_c, \delta_{max} < t_d$
Level 2	$k \leq t_k, l_{min} > t_l, c_{max} > t_c, \delta_{max} < t_d$
Level 1	$k > t_k, l_{min} > t_l, c_{max} > t_c, \delta_{max} < t_d$

can efficiently evaluate privacy-leakage using two different indices. The parameters  $t_d$ ,  $t_k$  and  $t_c$  in real services depends on a privacy policy of each user for the services.

The user makes a decision to transmit data based on the privacy-leakage level and trustworthiness of a service provider. For instance, Level 3, Level 4 and Level 5 information can only be transmitted to a trusted service provider.

Another alternative for service use using high-PL-level data is one where the user can modify the data to remove privacy information and then request the diagnosis server to check the modified data again. Adding extra functions that automatically modify data transmitted to the client module are useful extensions of the system that allow more convenient privacy protection.

The modification is executed as the third step, data modification, and generalization algorithms [1], [2] are examples of algorithms that modify the data to prevent privacy leakage. As described in 3.2.1 and 3.2.2, the diagnosis program can produce useful information for data modification.

### 5.3 Limitation of Data Types

It is assumed that the send data consist of  $n$  attributes that is labeled non-sensitive, quasi-identifier, or sensitive by a pre-defined data format for each service. Thus, our system is not suitable for services that send binary data or data that have an unfixed data format; In order to realize universal privacy leakage diagnosis for all types of data, we have to add another diagnosis method for binary data and transformation schemes that change unfixed format data to fixed format data. For binary data diagnosis, a signature-based diagnosis on the user terminal is effective and the transformation schemes should be defined for each service. This remaining issue will be addressed in our future work.

## 6. Related Work

There has already been research on data leakage detection. Papadimitriou and Garcia-Molina [12], [13] proposed data allocation strategies that inject "realistic but fake" data records in distribution data sets and detect data leakage and identify the guilty party. Bhattacharya *et al.* [14] proposed a privacy violation detection mechanism using data mining techniques. Bottcher and Steinmetz [15] proposed an identifying algorithm that detects sets of suspicious queries from

logs of an XML database. Ahmed *et al.* [16] presented an audit mechanism for leakage detection and identification of adversaries by counting the frequency of queries. Chow *et al.* [17] described a theoretical framework for inference detection using corpus-based association rules. Kim and Kim [18] proposed a monitoring approach for detecting information leakage using static rules. CutOnce [19] is an information leakage detection program for email systems. There are also several other studies that focused on information leakage detection in email systems [20], [21].

We proposed a privacy-leakage diagnosis method using quantitative analysis based on privacy notions, which is potentially applicable to several applications.

## 7. Conclusion

In this paper, we presented a privacy-leakage diagnosis module that evaluates privacy leakage when transmitting data. The diagnosis is based on two major notions of privacy, and sensitivity of each attribute values. A prototype program for the diagnosis achieved a computation time of less than 1 ms computation for the diagnosis and updating of data. The result suggests that our system is able to obtain diagnosis results within a feasible computation time.

In our future work, we will consider evaluating the feasibility of the evaluation results using real experiments, from the perspective of whether the examinee's expectations coincide with the results.

**Acknowledgement.** This work has been supported by the Japanese Ministry of Internal Affairs and Communications funded project, "Study of Security Architecture for Cloud Computing."

## References

- [1] B. Bayardo and R. Agrawal, "Data privacy through optimal  $k$ -anonymity," in *Proc. of ICDE 2005*, 2005, pp. 217–228.
- [2] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient full-domain  $k$ -anonymity," in *Proc. of SIGMOD 2005*, 2005, pp. 49–60.
- [3] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," in *Proc. of the 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (PODS'98)*, 1998, p. 188.
- [4] P. Samarati, "Protecting respondents' identities in microdata release," *IEEE Trans. on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.
- [5] L. Sweeney, "Achieving  $k$ -anonymity privacy protection using generalization and suppression," in *J. Uncertainty, Fuzziness, and Knowledge-Base Systems*, vol. 10(5), 2002, pp. 571–588.
- [6] T. Dalenius, "Finding a needle in a haystack—or identifying anonymous census record," in *Journal of Official Statistics*, vol. 2(3), 1986, pp. 329–336.
- [7] A. Machanavajjhala, J. Gehrke, and D. Kifer, " $l$ -diversity: Privacy beyond  $k$ -anonymity," in *Proc. of ICDE'06*, 2006, pp. 24–35.
- [8] —, " $t$ -closeness: Privacy beyond  $k$ -anonymity and  $l$ -diversity," in *Proc. of ICDE'07*, 2007, pp. 106–115.

- [9] X. Sun, H. Wang, J. Li, T. M. Truta, and P. Li, " $(p^+, \alpha)$ -sensitive  $k$ -anonymity: a new enhanced privacy protection model," in *Proc. of CIT'08*, 2008, pp. 59–64.
- [10] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang, " $(\alpha, k)$ -anonymity: an enhanced  $k$ -anonymity model for privacy preserving data publishing," in *Proc. of ACM SIGKDD'06*, 2006, pp. 754–759.
- [11] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [12] P. Papadimitriou and H. Garcia-Molina, "Data leakage detection," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, 2011, pp. 51–63.
- [13] ———, "A model for data leakage detection," in *Proc. of ICDE 2009*, 2009, pp. 1307–1310.
- [14] J. Bhattacharya, R. Dass, V. Kapoor, and S. K. Gupta, "Utilizing network features for privacy violation detection," in *Proc. of First International Conference on Communication System Software and Middleware*, 2006, pp. 1–10.
- [15] S. Bottcher and R. Steinmentz, "Detecting privacy violations in sensitive XML databases," in *Proc. of Secure Data Management 2005*, 2005, pp. 143–154.
- [16] M. Ahmed, D. Quercia, and S. Hailes, "A statistical matching approach to detect privacy violation for trust-based collaborations," in *Proc. of the First International IEEE WoWMoM Workshop on Trust, Security and Privacy for Ubiquitous Computing*, vol. 3, 2005, pp. 598–602.
- [17] R. Chow, P. Golle, and J. Staddon, "Detecting privacy leaks using corpus-based association rules," in *Proc. of the 14th ACM SIGKDD*, 2008, pp. 893–901.
- [18] J. Kim and H. J. Kim, "Design of internal information leakage detection system considering the privacy violation," in *Proc. of International Conference on Information and Communication Technology Convergence (ICTC) 2010*, 2010, pp. 480–481.
- [19] R. Balasubramanian, V. R. Carvalho, and W. Cohen, "Cutonce - recipient recommendation and leak detection in action," in *Proc. of The AAAI 2008 Workshop on Enhanced Messaging*, 2008.
- [20] N. Boufaden, W. Elazmeh, Y. Ma, S. Matwin, N. El-Kadri, and N. Japkowicz, "Peep - an information extraction based approach for privacy," in *Proc. of International Conference on Email and Anti-Spam (CEAS 2005)*, 2005.
- [21] C. Kalyan and K. Chandrasekaran, "Information leak detection in financial e-mails using mail pattern analysis under partial information," in *Proc. of the 7th International Conference on Applied Information and Communications (AIC'07)*, 2007, pp. 104–109.

# Secure Common Web Server Session

Sharing object data across deployed Java web applications on the same web server

Chad Cook and Lei Chen

Department of Computer Science, Sam Houston State University, Huntsville, TX 77341, USA

**Abstract** - *When web applications are deployed to a Java web server, there is no consistent or easy way to share object data among them. In this paper, we propose a mechanism, the Secure Common Web Server Session (SCWSS), which allows object data to be shared across deployed web applications, independent of the web server or any other implementation specifics, in a manner similar to storing session objects in Java. In SCWSS, the byte representation of the object data is first encoded to ASCII format, then encrypted (currently using DES), and finally saved in a cookie with a name supplied by the developer at the root level. Data can then be retrieved by any other application deployed to the same web server that can supply the correct encryption key. The proposed mechanism has been implemented, tested using various browsers, and analyzed for shortcomings and possible improvement.*

**Keywords:** Secure, web Applications, Java, session, cookies, encryption

## 1 Introduction

When developing Java web applications, there is often a need to store data, objects, for use elsewhere in the application. This can be conventionally done by saving data for that specific user's session which can then be retrieved elsewhere in the application. While useful for a single web application, it cannot be used to share data across web applications deployed to the same web server. Some web application vendors offer settings and mechanisms for allowing session data to be shared across web application, but these require administrator level access and developers may not have access to these permissions.

One method that is available for sharing data across web applications is by storing a cookie in the client's browser. However, this method only stores string data, which is not very useful to an object-oriented language such as Java. This paper proposes a method of using this cookie mechanism to share data in a secure manner between web applications by converting the object data into a string representation of that object, encrypting it, then storing it in a cookie for retrieval by any web application that has the proper key to decrypt it, and converting the string back into the original object.

The rest of the paper is structured as follows. Next in Section 2, we discuss the advantages and disadvantages in the existing ways to share data across web applications. We outline a number of key terms and concepts used throughout the paper in Section 3. An overview of the Data Encryption Standard (DES) scheme, the encryption standard we used to share data across web applications in a secure way, is introduced in Section 4. This is followed by description of the detailed implementation in Section 5, which describes object data sharing, implementation environment, diagrams of Java classes developed, and the code flows of object storage and retrieval processes. The implementation environment is described in Section 6 and is tested and analyzed in Section 7. At the end of the paper we discuss some of the known shortcomings of our implementation in Section 8 and areas for future enhancements and improvements in Section 9.

## 2 Background

Every now and then web application developers need to store data or objects to be used later in applications. Therefore session data need to be store in a reliable and sometimes secure way to preserve data confidentiality and integrity. When only a single web application is involved, the above method may be quite competent. However in a multiple web applications environment, there exist a number of potential problems, the most noticeable problem being the settings and mechanisms for allowing such session data sharing require administrator level access which developers and web applications normally do not have.

Currently, when sharing data between web applications deployed to the same web server, there are a couple of ways to do so. Some vendors offer settings at the server level to provide ways to share session data (also known as context sharing). For example, when using WebSphere Application Server V5, the WebSphere extension to the servlet 2.3 API allows sharing session context across an enterprise application [14] [15]. Session attributes must be serializable to be processed across multiple Java Virtual Machines (JVMs) [11]. Reliability and availability of a user's session state must be guaranteed. The In Process and Out of Process methods used by ASP.NET can be configured to maintain session state [10] in a reliable way. While this can be useful, it does not work if the applications need to be deployed to a

different vendor's web server that does not offer context sharing or if the developer does not have access or a way to modify the server's settings to enable context sharing.

Data has also been shared by writing its binary counterpart to a database, where other applications can then access it. While this approach is web server vendor independent and the developer does not necessarily have administrator access, it is a very tedious approach, requiring changes to the database whenever new objects need to be saved and possible modification to existing database metadata when changes to the objects occur. When the database environment changes, all tables must be copied over, which can be tedious if this is a switch between vendors.

Another method that allows data to be shared is by using a cookie [5] [8]. A cookie is string data that is stored in the client's browser and resides at the level of the web application that creates it. However, the path of the cookie can be created to the root level, giving all web applications deployed to the server access to it. While this approach does not require any administrative changes and is vendor independent, the fact that it uses cookie data limits the data that can be shared to strings. But if there is a way to convert object data to a string and vice-versa, a simple, vendor independent approach that requires no administrative assistance becomes available to the developer. Our goal was to develop such a solution. In the next section, we introduce a number of terms frequently used in our secure object data sharing mechanism.

### 3 Key terms and concepts

Below are a number of the key terms and concepts used in this paper:

- **Client:** the end user, typically referring to the end user's computer or computing device. The term client also refers to client side applications. The client sends requests for data and/or to perform actions to a server.
- **Server:** the machine where the web server and web applications deployed to it are running. The term server also refers to the server applications. When the server receives a request from a client, it sends a response after processing the request.
- **Web Server:** a piece of software that is run on a machine with an internet connection. It processes requests for web pages and other internet related data and typically sends data back after processing the request.
- **Web Application:** an application which is based in the web browser. A single web application can be run for many different users at once, each with their own session.

- **Cookies:** string data which is saved on the client's machine. This is typically informational data and can be included if a user has accessed this web application before and any user (client) preferences.
- **Data Encryption Standard (DES):** a widely used encryption standard. DES is covered in detail in section IV.
- **Java Session:** when a client first accesses a web application on a web server, a unique session is created for that specific web application. This session "lives" while the client interacts with the web application. They can store and retrieve information in the session while they interact with the web application. However, they cannot share data with other web applications and once the client terminates their interaction with the web application the session is removed.

### 4 Data Encryption Standard (DES)

The Data Encryption Standard Scheme (DES) [3] [4] is a standard encryption scheme, used both by the government and privately [12]. DES is a symmetrical encryption algorithm as it uses the same 64-bit key (56 bits for encryption, 8 bits for parity checking) to both encrypt and decrypt the data. This is done by providing a key to encrypt the plaintext data, resulting in encrypted text, or ciphertext. To decrypt the ciphertext, the same key used to encrypt the data must be supplied to decrypt the data back to its plaintext form.

To perform the actual encryption, data is split into 64 bit blocks and then fed through 16 rounds of processing. To perform each round of processing, the 64 bit block of text is split into two 32-bit halves. Each half is then expanded by using substitutions and permutations of bitwise shifts and reordering, resulting in 48 bits. Then a subset of the key is combined with the 48 bits using an XOR operation and the block is again divided into smaller pieces where each piece is fed into a substitution box where a non-linear transformation is used to ensure that the cipher will not be trivially breakable. To perform the decryption, the process is run in reverse order.

Since a brute force attack against encryption using a 56 bit key is relatively easy to perform [6], the DES process can be repeated two more times, called Triple DES and in effect creates an encryption key of 56, 112 or 168 bits to use, depending on which of the three the keying options is used [13]. This can still be brute force attacked, but it does increase the security of the encryption algorithm to some extent. While it was reported that Triple DES may suffer from meet-in-the-middle (not man-in-the-middle) attack [7], Triple DES is more than sufficient for data confidentiality and integrity in our project. We discuss the possibility of using the Advanced Encryption Standard (AES) [1] in the last section Future Work.

## 5 Implementation

### 5.1 Object data sharing

Taking the advantage of cookies that can be shared amongst all web applications we developed a process to convert the specified object into a consistent string representation for storage in the cookie. This string representation can then be retrieved by any other application deployed to that web server, converting the string back into its original object for use by that application. Figure 1 below depicts how this process works in a nut shell.

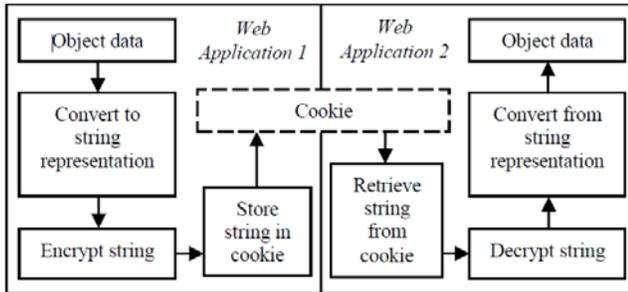


Figure 1. Object sharing mechanism

To start, the object is supplied and converted into a byte array. This is done via a byte array output stream supplied to an object output stream where the object is written out as a byte representation. The byte representation is saved as a byte array and is then encoded via Base64 encoding, to an ASCII format, which allows for the byte array to be easily converted into a string. The encoded byte array is then encrypted with a supplied encryption key and then saved in a cookie with a name supplied by the developer at the web server's root level. By encrypting the data, any other web application wanting to read the data must know the key, preventing unauthorized web applications which are also deployed to the same web server from accessing the data.

When a web application wants to read the data, the above mentioned process is reversed. The developer provides the cookie's name and the decryption key. The cookie is then retrieved (if it exists) and the cookie value is decrypted. The string representation must be decoded back into a byte array using Base64 decoding. The resulting byte array can then be fed into a byte array input stream which is then given to an object input stream. The object input stream can then read the object from the array and return it to the developer.

Once the actual conversion process is completed, then next goal is to implement an easy way for the developer to use this technology. Our goal is to mimic the storage and retrieval of session data in Java, thus two methods were devised to allow for working with the common web server data. For saving data, a `setAttribute()` method was developed, taking the name to save the object under, the

object to be saved, the response object for working with the cookie and the encryption key to encrypt the data with. For retrieving data, a `getAttribute()` method was developed, taking the name of the object to look for, the request object for working with the cookie and the encryption key for decrypting the data with. By using the same method names as the methods used to work with data in the session, we will create an intuitive and easy way to also share data between deployed web applications. All the required classes are stored in a single Java Archive (JAR) file and by including the JAR file in the application's or server's classpath working with session data is relatively straightforward.

For data encryption and decryption, Data Encryption Standard (DES) scheme was used. As the purpose of this research was not related to how data is encrypted, but rather that the data could be encrypted to prevent unauthorized access, DES scheme was used as a proof of concept.

### 5.2 Implementation environment

The implementation environment used the following applications:

- Eclipse Helios SR1
- Java SE 1.6
- Apache Tomcat 6.0.20
- Mozilla Firefox 3.6.12
- Internet Explorer 8.0.7600.16385

Our secure common web session object was developed in Eclipse using Java. We had two web applications to test our implementation, one for setting the data and the other for retrieving the data. Both web applications were deployed to the same instance of Apache Tomcat. To ensure there would be no problems with the cookies, the setting and retrieval of data was tested with both Firefox and Internet Explorer to help ensure cross-browser compatibility.

### 5.3 Class diagrams

The following are the diagrams of the Java classes developed.

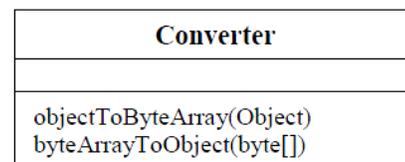


Figure 2. Converter class diagram

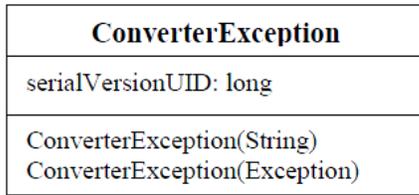


Figure 3. ConverterException class diagram

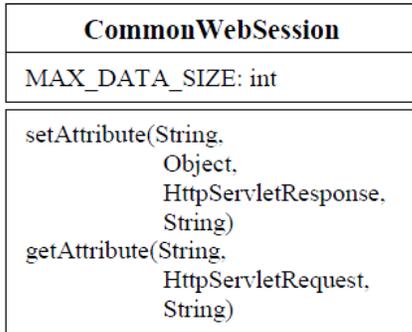


Figure 4. CommonWebSession class diagram

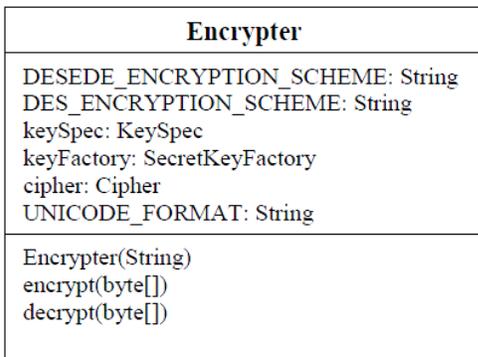


Figure 5. Encrypter class diagram

### 5.4 Code flow

The following figures demonstrate the flow of code for object storage and object retrieval processes.

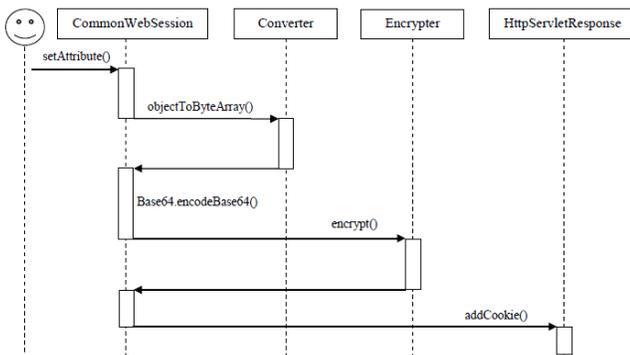


Figure 6. Code flow for object storage process

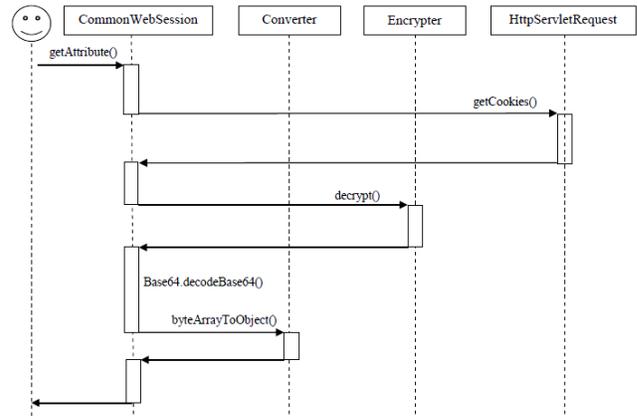


Figure 7. Code flow for object retrieval process

## 6 Testing

To test our implementation we created two separate web applications in Eclipse – one would write data to the common web session and the other would read the data back out. To further ensure the validity of the implementation, custom objects were created which contained standard Java objects and other custom objects, ensuring the entire object and all data it contained, including other objects, would be saved and retrieved properly.

The first web application would run a simple html page which submits values to a test servlet. The servlet would then take these values out of the request and use them to populate the custom object. The custom object would then be saved using the common web server session. The second web application would run a JavaServer Pages (JSP) program which would request data from its test servlet. The test servlet would retrieve the custom object from the common web session and then construct an URL from the values from the custom object and forward to this URL, which would call the JSP program and display the contents of the object.

By allowing for form submission of the data in the custom object, it was easy to quickly test that various combinations of data were being written to the common web session and were being retrieved properly as well. We could also make multiple attempts to save and retrieve the same object and ensure the consistency of the data that we were working with.

## 7 Known shortcomings

While our research and implementation showed that we developed a simple, reliable, and secure way to share object data across web applications deployed to the same server, we also identified some limitations.

The main problem we encountered was in converting the object data to a string representation. Because we were using an object output stream, objects must be serializable

to be written to a cookie. This means that any object which does not implement the Serializable interface cannot be saved with our implementation. The other limitation regarding serializable objects is that any data which is marked transient will not be output, meaning certain member data can be lost. We encountered this in our testing as some of the member data was marked transient in our test objects and the application which read from the common web session had empty data for these values as they were not output (or saved).

As the main method for saving and sharing data relies on cookies, we are also constrained by any limitations due to using cookies. The first is cookie size, which cookie specifications state should not be longer than 4k [9]. This means that we are limited to a string representation of approximately 3,800 characters, leaving room for the name of the cookie and other header information. While even arrays of one hundred of our custom objects only generated 300 character string representations, it is still possible that the cookie size limit could be hit, thus we had to put in checks to ensure the generated string length will all be saved in the cookie.

As the data is stored in cookies, any client that has cookies disabled will be unable to take advantage of this functionality.

## 8 Future work

As one of the major weaknesses in the implementation was the omission of non-serializable and transient data, identifying a way to include this, and in turn all object data, will be very useful. More research into ways to obtain the data from the objects and in turn reconstruct the objects would be beneficial. Some preliminary research into decomposing an object into a string found potential solutions [2], but there was no non-trivial way to reconstruct the object from the decomposition.

As the client must have cookies enabled for our proposed implementation to work, identifying other simple ways to save the string data will eliminate this requirement. As one of the goals of this work was to avoid involving a database or any component that is not “always” included in a web session, we will try to avoid any solution that involves the use of a database.

When continuing with a cookie approach, the current implementation does not set an expiration date on the cookie data, defaulting to the client’s browser’s expiration date. Developing a “timeout” for the cookie by determining a specific expiration date/time will be very useful. Not only will this value need to be set when the cookie is created, but periodically it will need to be checked so that if the cookie is not accessed for a period of time it does not expire while the user is interacting with other parts of the web application.

The DES scheme was used as a simple proof-of-concept to show that the data can be stored securely from prying eyes. More advanced encryption schemes, such as AES, can be used. In addition to more advanced security, other encryption schemes that can offer better data compression will be useful in helping to avoid any character limits of the cookie or any other location the data could be stored in the future.

As the current encryption scheme, DES, is hard-coded into the implementation without any way of overriding it, this greatly reduces the implementation’s maintainability and lifespan. Allowing a mechanism to override the default encryption scheme would be useful, especially for those that are required to use a certain scheme or a more powerful encryption scheme. Also, encryption schemes change with the times as they become more powerful and allowing for a way to override the encryption scheme used will increase the lifespan of the implementation a great deal.

## 9 References

- [1] *Announcing the Advanced Encryption Standard (AES)* (2011, March 15). Retrieved from <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- [2] *Apache commons: lang.* (2011, March 15). Retrieved from <http://commons.apache.org/lang/>
- [3] *Data Encryption Standard.* (2011, March 15). Retrieved from <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>
- [4] *Data Encryption Standard (DES).* (2011, March 15). Federal Information Processing Standards Publication 46-2. Retrieved from <http://www.itl.nist.gov/fipspubs/fip46-2.htm>
- [5] David M. Kristol, “*HTTP Cookies: Standards, privacy, and politics*”, *ACM Transaction on Internet Technology (TOIT)*, vol. 1 issue 2, Nov. 2011
- [6] Gilmore, John, "*Cracking DES: Secrets of Encryption Research, Wiretap Politics and Chip Design*", 1998, O'Reilly, ISBN 1-56592-520-3.
- [7] *Meet-in-the-middle attack* (2011, March 15). Retrieved from [http://en.wikipedia.org/wiki/Meet-in-the-middle\\_attack](http://en.wikipedia.org/wiki/Meet-in-the-middle_attack)
- [8] Michael Nelte and Elton Saul, “Cookies: weaving the Web into a state”, *Crossroads*, vol. 7 issue 1, September, 2000
- [9] *Number and size limits of a cookie in internet explorer.* (2011, March 15). Retrieved from <http://support.microsoft.com/kb/306070>

- [10] *Selecting the Method for Maintaining and Storing ASP.NET Session State.*(2011, March 15). Retrieved from <http://technet.microsoft.com/en-us/library/cc784861%28WS.10%29.aspx>
- [11] *Session management for clustered applications.* (2011, March 15). Retrieved from <http://www.oracle.com/technetwork/articles/entarch/session-management-092739.html>
- [12] T. Schaffer, A. Glaser, S. Rao and P Franzon, "A Flip-Chip Implementation of the Data Encryption Standard (DES)", *IEEE Multi-Chip Module Conference (MCMC '97)*, pp 13-17.
- [13] *Triple DES Encryption* (2011, March 15). Retrieved from <http://publib.boulder.ibm.com/infocenter/zos/v1r9/index.jsp?topic=/com.ibm.zos.r9.csfb400/tdes1.htm>
- [14] *Websphere application server v5: sharing session context.* (2011, March 15). Retrieved from <http://www.redbooks.ibm.com/abstracts/tips0215.html?Open>
- [15] *Websphere application server version 6.1: assembling so that session data can be shared.* (2011, March 15). Retrieved from [http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tpres\\_sharing\\_data.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tpres_sharing_data.html)

# Private Information Retrieval in an Anonymous Peer-to-Peer Environment

A. Michael Miceli<sup>1</sup>, B. John Sample<sup>2</sup>, C. Elias Ioup<sup>1,2</sup>, D. Mahdi Abdelguerfi<sup>1</sup>

<sup>1</sup>Computer Science, University of New Orleans, New Orleans, Louisiana, United States

<sup>2</sup>John C. Stennis Space Center, Hancock County, Mississippi, United States

**Abstract** - *Private Information Retrieval (PIR) protocols enable a client to access data from a server without revealing what data was accessed. The study of Computational Private Information Retrieval (CPIR) protocols, an area of PIR protocols focusing on computational security, has been a recently reinvigorated area of focus in the study of cryptography. However, CPIR protocols still have not been utilized in any practical applications. The aim of this paper is to determine whether the Melchor Gaborit CPIR protocol can be successfully utilized in a practical manner in an anonymous peer-to-peer environment.*

**Keywords:** Anonymous peer-to-peer, Private information retrieval (PIR), peer-to-peer

## 1 Introduction

The first amendment of the United States constitution guarantees freedom of speech, which allows citizens to express their opinions without fear of persecution. Freedom of speech is necessary for a functioning democracy. However, not every country has this freedom and there also have been many instances in the United States where freedom of speech has been restricted. For instance, after the publication of the over 250,000 leaked United States diplomatic cables in 2010, there was rash reaction from lawmakers. Sen. Lieberman introduced the "Protecting Cyberspace as a National Asset Act of 2010", which would have allowed for government control of Internet service providers (ISP)s, arguing that in a national emergency the government should be able to 'shut down' the Internet. This vague law with no checks and balances would have greatly destroyed freedom of speech. Only in 2011, when Egypt was able to prevent Internet access to citizens in an attempt to unsuccessfully prevent a revolution was the bill revised to remove this 'shut down' idea. This ebb and flow of censorship and regulation has shown a need for the ability to communicate freely without government control.

## 2 Anonymous Peer-To-Peer Systems

Any distributed application where peers process tasks or work between them is considered a peer-to-peer network. There are 3 different types of peer-to-peer networks: centralized, semi-centralized, and decentralized. A centralized peer-to-peer system uses a central server either to distribute work, or help locate information among nodes. Examples of a central system are Napster and BitTorrent. Semi-centralized peer-to-peer systems do not have a central server; however, they do have nodes that are more important than other peers. These peers accept more traffic and may control a part of the total network. Semi-centralized systems include Kazaa [1] and Skype [2]. Finally, there are fully decentralized peers where all nodes have the same priority and are equally valuable to the network. Freenet and GUNet are examples of fully decentralized peer-to-peer networks. Anonymous peer-to-peer networks are peer-to-peer applications in which peers are anonymous to each other. Peers cannot know who is requesting data and who is sending data.

The purpose of anonymous networks is usually to provide freedom of speech. They are a gray area of the law. While information about corrupt governments can be spread throughout the network without the whistleblower being held responsible, so can child pornography and other illegal information.

## 3 Freenet

The prototype peer-to-peer network in this paper is heavily influenced by Freenet. In the original paper [3], Clarke et al. lay out a framework for a distributed, censorship-resistant, peer-to-peer network called Freenet. It provides both sender anonymity and receiver anonymity to the level of possible innocence, i.e. nontrivial possibility that the sender/receiver was not the sender/receiver. The network was created with five goals in mind: provide anonymity for producers and consumers, provide deniability for

maintainers of data, be resistant to attempts to deny access to information, have efficient routing and storage, and be decentralized. In Freenet, every peer contributes part of his hard disk drive space for use in a large distributed data store, where nodes store other nodes' data. Each data item stored on each node's hard disk drive is encrypted using AES (Rijndael), where the passphrase determined by the creator of the data. While the decryption keys are theoretically available to the nodes that store this data from other users, they are not obliged to find these keys and determine the contents on their local data store. This provides plausible deniability for nodes that are holding incriminating content from other users, because the node does not and cannot easily know what data it is holding, because it is hard to determine what the decryption key is unless the node knows the passphrase. These nodes only know that they are holding data that has been inserted into Freenet. The main idea of Freenet is that if the user running a Freenet peer is discovered to be holding illegal content, there is no way to know for sure the peer knew this.

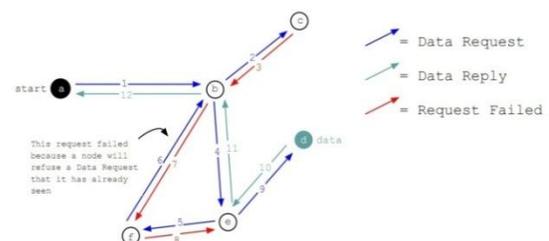
When a peer requests a file, it will spread throughout the network (See 4 Routing). Therefore, the more popular an item is, the more available it will be for other nodes in the network. As nodes' caches become full, least recently accessed items will be removed. The only way a file will be removed from all of Freenet is if no node requests the file until all caches purge the item. This satisfies Freenet's goal of providing resistance to deny access to information and it also removes ownership of sensitive information.

## 4 Routing

Freenet is anonymous because requests are routed through intermediate nodes, which prevents knowledge of where the original request came from. To find a file in Freenet, a simple routing algorithm is used. When a request for data is received by a node in the Freenet network, it will first look in its local data store. If the data is found, then the node will respond saying it has the data. If the item is not in the local data store, the node will ask a neighbor that it thinks is most likely to know which node has the data. This is determined by nodes that have returned "similar" items before. Similarity is defined as the lexicographical difference between the hashes of the description of the data described previously. If a node cannot find a close neighbor, he will send the data requested to a random neighbor. All nodes will recursively do this until the data is found or the number of hops is exceeded. If a destination is found

that has the specified hash, all nodes on the route back to the requestor will cache the data to provide faster access for subsequent requests for that data. They may also randomly decide to change the source to themselves. This provides source anonymity, by preventing a requestor from knowing who originally had the data exactly. Note that on a successful request all nodes in the successful route will cache the data. To prevent loops each message has an ID and each node keeps track of recent IDs it has received. Also, the hops to live (HTL) – number of hops – value can be set to any value in a small range by the requesting node to create requestor anonymity. The main idea with this routing algorithm is that the network will adapt to requests, and over time become very efficient.

Figure 1: Example of a data request



To insert an item into Freenet, a user sends a special message that will attempt to insert the data. A request is sent to many neighbors and if no neighbors report a collision, all nodes will store a copy of the data. If the data is successfully inserted into the network, it will remain until all nodes' datastores drop the file.

## 5 Private Information Retrieval

Private information retrieval (PIR) protocols are protocols that provide a way for clients to request data from a database without the database knowing which data was requested. They are useful in several application domains, such as stock market databases and patent databases. PIR protocols are a weakened form of 1 out of n oblivious transfer - introduced in 1981 - where database privacy is not a concern. PIR protocols were first introduced in 1995 by Chor, Goldreich, Kushilevitz and Sudan in [4]. The paper proposed a system that was information theoretic secure and relied on multiple non-communicating servers. They also proved that to be information theoretic secure, you must have multiple non-communicating servers. Later in 1997, the idea of Computational Private Information Retrieval (CPIR) schemes was introduced by both Chor and Gilboa, and Ostrovsky and Shoup. [5] [6] Computational

privacy is defined as privacy that is guaranteed against computationally bounded attackers. CPIR protocols are not information theoretic secure and do not have to have multiple non-colluding servers. After Kushilevitz and Ostrovsky proposed the first successful CPIR scheme, there have been many papers describing new schemes. Each protocol focuses on reducing the communication complexity.

When studying PIR protocols, the privacy of the requestor considers the inability of the server to determine which elements are queried. It is not the confidentiality of the requestor.

## 6 Practicality

There has been a debate over the usefulness of CPIR protocols. An influential paper in 2007 by Sion and Carbunar argue that all single server CPIR protocols are not only impractical today, but also will not be useful in the near future assuming both Nielson's law and Moore's law are stable. [7] While the main goal of (C)PIR protocols has been to minimize communication, Sion and Carbunar show that a low communication complexity is worthless if the computation complexity is so large that the trivial implementation of PIR would take less time. The trivial implementation of a (C)PIR scheme is to respond to every request with the entire database. This is very bandwidth inefficient; but does ensure client privacy. However, it is very computationally efficient compared to CPIR protocols at the time of Sion and Carbunar's paper. Pitting CPIR protocols against the trivial solution seemed like a good idea during the PIR's initial development, but computational complexity was ignored. *Figure 2* shows the time taken for two CPIR protocols at the time of Sion and Carbunar's paper. The database size was 3 GiB and the query was for a 3 MiB file.

*Figure 2: Common times a typical CPIR scheme would take at the time of Sion and Carbunar's paper*

PIR Protocol	Download Time
Limpaa	33 hours
Gentry and Ramzan	17 hours

These numbers are unreasonable considering today's network speeds. According to Sion and Carbunar, an average home computer can download about .75 Mb per second (6 Mbps). So, downloading a 3.0 GiB database (trivial solution) would take about 68 minutes. Consequently, Sion and Carbunar's paper has been used by other researchers to dismiss single server PIR protocols as impractical. [8] In response to this, two papers developed new protocols that were

much more efficient overall, because of increased network utilization: the Melchor and Gaborit protocol, and the Trosle and Parrish protocol. [9] [10] This paper uses the Melchor and Gaborit's protocol, because it had faster transfer times. Melchor and Gaborit focused on removing the extreme bound on communication, allowing more information to be sent between the CPIR server and client. Their scheme is one hundred times faster in terms of time than previous single database schemes. On top of that, Melchor, et al. takes advantage of the linear algebraic characteristics of their protocol to create further optimizations. Using a GPU programmed with CUDA, they obtained several orders of magnitude faster than their original implementation. [11] With these optimizations, the same query as *Figure 1* takes close to 10 minutes. A recent paper by Olumofin and Goldberg, confirm that Melchor and Gaborit's scheme is indeed practical, refuting their own earlier claims. [12]

## 7 Overhead

The goal of this section is to determine which parameters would be the most practical in a network where retrieving a random file is a common operation. For each query, there is a noticeable bandwidth overhead, which depends on different parameters. In a database with very heterogeneous file sizes, large files will ruin any performance gains on small files, because smaller database elements are filled up using a standard padding technique. This prevents query and response sizes from leaking information about which of the database elements was queried. For instance, querying a 32 KiB file on a database that also has a 500 MiB file will require downloading at least 500 MiB of data. This overhead is too much for such a relatively small file. This is an area of concern on the practicality of CPIR schemes in general. However, in application domains with homogenous data sets, this is not a concern. For an anonymous peer-to-peer network that can store any data, this is of upmost concern.

## 8 Parameters

Determining how to deal with heterogeneous databases is difficult. Chunking files in a database will create a homogeneous database, which keeps query response sizes small, but it also increases the number of files in a database, which negatively affects the query size. Partitioning a database into many databases reduces the number of files in a database, but it also reduces anonymity. A combination of partitioning and chunking could create a scheme that has acceptable download speeds

and acceptable anonymity. The first goal is to determine what to chunk. If a larger chunk sizes is used and only files larger than the chunk size get chunked, then there will be no internal fragmentation. The files that are smaller than the chunk size should not be chunked to avoid fragmentation. This removes unnecessary padding.

Figure 3: Download speeds of databases with large chunk sizes on a GPU enabled node.

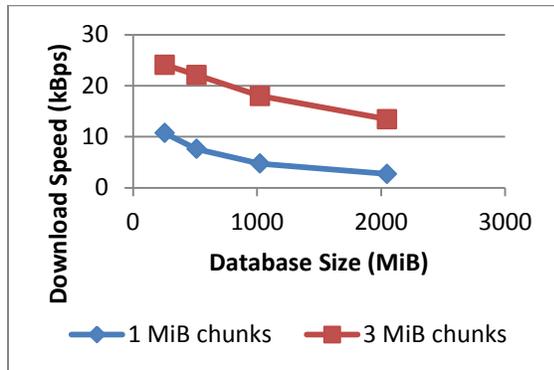
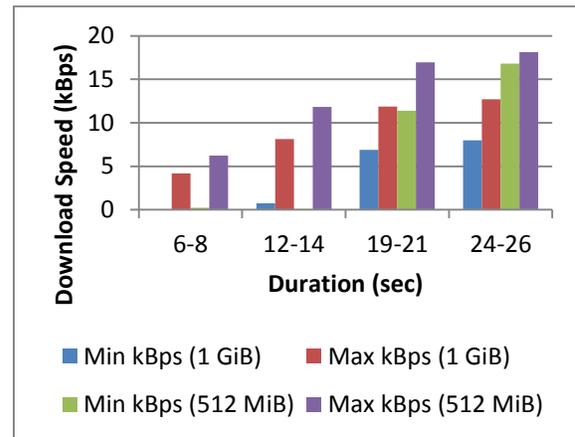


Figure 3 shows that having a large chunk size (3 MiB) provides decent download speeds. To determine whether 3 MiB chunk sizes are acceptable, the download speed on files less than 3 MiB should have to be acceptable as well. Figure 4 shows the minimum and maximum download speeds recorded on a database of either 512 MiB or 1 GiB with files less than 3 MiB whose file sizes were distributed evenly using a GPU enabled node assuming the average network download speed is assumed to be 6 Mbps and the average upload speed is assumed 2.64 Mbps. [13] To create Figure 4 every file in the database was downloaded and the total download time was observed. For each timespan, the largest and smallest files downloaded in that time range were noted. This determined the minimum and maximum download speeds. Figure 4 shows that the total database size can be relatively large and still have decent performance.

A decent tuning to optimize a GPU node's download speed and bandwidth is to have two separate databases: one for large files, which will be chunked into 3 MiB chunks; and one for files smaller than 3 MiB, which will not be chunked. The database for larger files is partitioned into 1 GiB partitions and the database for smaller files is partitioned into 512 MiB partitions.

Figure 4: Download speeds for random sized data under 3 MiB for a GPU enabled node.

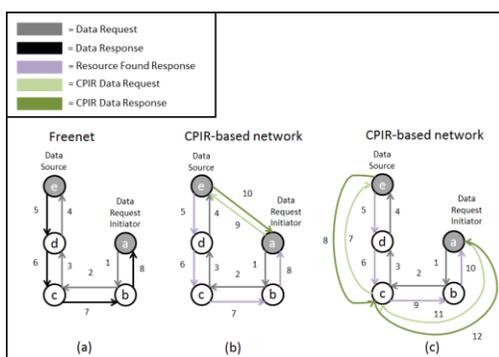


Testing found that a CPU cannot give usable download speeds, while still maintaining privacy. This issue casts a dim light on the practicality of using this peer-to-peer network, because the average user does not have a CUDA enabled video card; however, this is becoming increasingly less likely.

## 9 Experimental CPIR Based Scheme

Freenet is very popular but has a certain bandwidth and latency overhead. All data requested is cached at intermediate nodes. This paper proposes the idea that a new network could be created using CPIR protocols to obtain information from a node, instead of forwarding data through many intermediate nodes. When a node has data being requested, instead of sending the data to the requesting node (which is most likely an intermediate node), it simply responds by saying it does have the requested data. These responses are forwarded back until the original requestor obtains the response. The original requestor then requests the data by using the CPIR protocol (See (b) of Figure 5). To prevent timing attacks and to spread data throughout the network, intermediate nodes may intervene and cache the data before responding (See (c) of Figure 5).

Figure 5: Experimental CPIR-based idea.



By using CPIR, requestor anonymity is protected without the need of going through intermediate nodes. Remember, when using a (C)PIR protocol the sender does not know which file the requestor is requesting. In the Freenet paper Clarke et al. discredit the use of (C)PIR protocols in an anonymous peer-to-peer environment. They argue that in most cases, the act of contacting a particular server itself is incriminating and should be avoided. In Freenet a node never knows when someone is actually requesting data for itself. The data may always be for another node. The authors may have overlooked using PIR along with intermediate nodes to provide plausible deniability for requesting nodes, ensuring there is no more risk in contacting a Freenet data node than being a member of Freenet itself.

The goal of this paper is to create a system similar to Freenet with lower latency and requiring less overall network bandwidth, while still providing the same level of anonymity for the sender, receiver and all participating nodes in the network. This system would have to be practical. Currently, it is possible to create a network using CPIR systems with much less bandwidth usage than Freenet, but the computational costs would be too high to be practical.

## 10 Details

This scheme is very similar to Freenet, but with a few notable changes. The first change is the way files are downloaded from the network. All file downloads are through Alguilar and Gaborit's CPIR protocol. This gives requestor anonymity. With this in place, there is much less need to cache data in a network. Freenet caches data in its network for three reasons: to provide anonymity when sending data, to provide deniability for requesting data, and to provide deniability for holding the data. When sending incriminating data, the node can claim that it is forwarding data and does not know what the data is

nor who uploaded it. When requesting incriminating data, the user can claim that he is caching it locally and did not request the data. The user can also claim he did not know what the data was. Finally, when caught storing incriminating data the user can claim that he has only cached it for someone else and that he never requested the data. By using Melchor and Gaborit's CPIR protocol, the first two reasons for caching data become unnecessary. Caching is still necessary because we need to provide deniability for storing data; however, not every node needs to cache requests. This has the potential to decrease latency dramatically. The percentage of nodes in a request route that request data will also affect the spread of data in a network, which is necessary for network usefulness.

Routing is very similar to Freenet. Like Freenet this network uses the steepest-ascent hill-climbing search with backtracking, requesting data from nodes that have returned data most similar to the requested hash before. On a successful route though, most nodes will not cache the data. This can be done randomly or by some other preference. Nodes that do cache the data can arbitrarily change the source to themselves. The sender is much less responsible for spreading incriminating data in this network, because the sender is not able to determine which data was requested and consequently which data he actually sent. The criteria for determining whether a successful request should be cached could be file size, or computational load at the moment, or a strict percentage. In the prototype network tests, a strict percentage is used, but there are advantages of other methods

## 11 Experiments

The goal of these experiments is to show that this prototype network is practical. Many full scale tests were run on a local machine to determine some statistics. These results were compared against a typical Freenet node running on the same machine. There are many things to determine. One limiting factor is the computation. Each node will have requests that it must fulfill and CPIR operations are computationally expensive. Statistics of Freenet are published by nodes who wish to help determine the health of Freenet. After running a Freenet node whose cache was 2 GiB with the default settings on a 2 GHz machine with 2.00 GiB of RAM for twenty-four hours, the total load on the node was determined. Bulk requests are requests whose latency isn't as important. A real-time requests demand faster latency. Most of the Freenet requests that were measured were real-time requests. When a Freenet node is busy, it will drop requests; so the standard

deviation within each hour is very little. Each request is for a 32 KiB segment. The results depend on configuration settings. The default settings were chosen for all options, except the database size: a 2 GiB cache was chosen. During the most demanding hour of the test the node had to process 620 requests: 600 real-time and 20 bulk requests. This means that every minute of the peak hour there were, on average, 11 find requests being successfully responded to: 10 from real-time requests and 1 from bulk-requests. Stress testing the machine, which had a CUDA enabled GeForce 8800 GTX video card, was able to produce upwards of 8 real-time CPIR requests per minute. Limiting real requests per minute to 7 real-time requests per minute and limiting bulk requests to 2 per minute, the node is able to process all requests. This would limit the total requests that a node could handle to be 420 real-time requests per hour. The recorded measurement of 620 requests is too high and the node would have to limit these requests for peak hours. This restriction does not stop the network from functioning but it does limit the amount of requests that a node can handle. Testing this on the prototype network did not seem to affect network conditions greatly. The network was able to recover and route data around busy nodes. Not using a CUDA enabled video card, a node can only handle almost 1 bulk-request per core per minute. The same goes with real-time requests. This is because the request uses almost all CPU for almost an entire minute. The server processing time is almost 50 seconds alone.

It is impractical to think that a node would be willing to spend all CPU/GPU processing time to handle requests for other nodes in a network. Most nodes have ample hard drive space and asking to give up a small percentage makes sense. Limiting the number of requests per minute to such a low number where CPU/GPU is not being constantly worked at near maximum capacity makes the network very unresponsive. By using experimentation, there is not a way that could be found to create a network that would be practical using CPIR exclusively for downloading files. Every method that would have decent download speeds requires too much CPU/GPU use. Average users do not want to sacrifice their computer for freedom. Besides the crippling factor of the CPU/GPU utilization, the bandwidth overhead is too large to be close to what Freenet's bandwidth has.

## 12 Conclusion

The network this paper proposes, while not being more efficient than Freenet, can be practical in some

sense. It is possible to use this network with even though it has very high bandwidth overhead by limiting the total requests handled per hour and having a CUDA enabled GPU. However, most computers do not have CUDA enabled video cards, yet. So, it is very difficult to imagine this network being practical. Also, overall there is much more bandwidth used than originally anticipated. For instance, downloading a 3 MiB request for a 1 GiB database – the recommended parameters – required at least a total of 41.039 MiB of bandwidth. In Freenet this would have been, on average, significantly less. So, while this network is practical it is not very useful unless a target audience would be willing to accept such performance penalties for added anonymity.

## 13 Future Work

The prototype network uses a lot of bandwidth and computational power and probably isn't too useful in the near term. However, there are a few ideas that could be researched that would allow this network to be practical.

One idea is a Freenet plugin that would use this CPIR-based protocol on more secure requests. This would stop the server from knowing what is obtained while creating a small overhead in bandwidth and computation. Since few requests use CPIR, a node can handle CPIR requests with only a CPU and not notice much overhead. This could be done on top of Freenet's current network and allow nodes to optionally participate when they either have available CPU.

Another avenue of research should be in CPIR requests today. Only recently was computational complexity taken into account for CPIR optimizations. So, a more efficient protocol could be developed that uses both less computation and less bandwidth. If so, then this scheme may become more practical.

## References

- [1] Understanding Kazaa. Liang, Jian, Kumar, Rakesh and Ross, Keith W. 2004.
- [2] An Analysis of the Skype Peer-to-Peer Internet Telephony. Baset, Salman A. and Schulzrinne, Henning. 2004 : s.n.
- [3] Freenet: a distributed anonymous information storage and retrieval system. Clarke, Ian, et al. Berkeley : Springer-Verlag New York, Inc., 2001, Workshop on designing privacy enhancing technologies, pp. 46-66. 3-540-41724-9.

- [4] Private Information Retrieval. Chor, B, et al. Washington, D.C. : IEEE Computer Science, 1995, FOCS '95, pp. 41-50. 0-8186-7183-1.
- [5] Computationally Private Information Retrieval. Chor, Benny and Gilboa, Niv. El Paso : ACM, May 4, 1997, STOC 1997, pp. 304-313. 0-89791-888-6.
- [6] Private Information Storage (Extended Abstract). Ostrovsky, Rafail and Shoup, Victor. 1997.
- [7] On the Computational Practicality of Private Information Retrieval. Sion, Radu and Carbunar, Bogdan. 2007, In Proceedings of the Network and Distributed Systems Security Symposium.
- [8] Privacy-preserving Queries over Relational Databases. Olumofin, Femi and Goldberg, Ian. July 2010, PETS 2010, pp. 75-92.
- [9] Efficient Computationally Private Information Retrieval From Anonymity or Trapdoor Groups. Trosle, Jonathan and Parrish, Andy. [ed.] Mike Burmester, et al. s.l. : Springer Berlin / Heidelberg, 2011, Lecture Notes in Computer Science, Vol. 6531, pp. 114-128. 978-3-642-18178-8\_10.
- [10] A Fast Private Information Retrieval Protocol. Melchor, Carlos Alguilar and Gaborit, Phillippe. Toronto : s.n., July 6-11, 2008, ISIT 2008, pp. 1848-1852. 978-1-4244-2256-2.
- [11] High-speed Private Information Retrieval Computation on GPU. Alguilar-Melchor, Carlos, et al. 2008, SECURWARE '08, pp. 263-272. 978-0-7695-3329-2.
- [12] Revisiting the Computational Practicality of Private Information Retrieval. Olumofin, Femi and Goldberg, Ian. June 2010, CACR Tech Report, p. 16.
- [13] Household Upload Index. Net Index. [Online] <http://www.netindex.com/upload/allcountries/>.

## On Querying Encrypted Databases

Moheeb Alwarsh and Ray Kresman

Department of Computer Science  
Bowling Green State University  
Bowling Green, OH 43403  
{moheeba, kresman}@bgsu.edu

### Abstract

This paper presents a new range query mechanism to query encrypted databases that reside at third-party, untrusted, servers. This paper is a continuation of work done by others [1]; our scheme seeks to improve the precision of querying encrypted data sets, increase the utilization of server side processing and reduce the computation and memory utilization on the client side. We compare our algorithm with previous work, and quantify the performance improvements using numerical results.

Key words: Trust, encrypted database, bucketization, binary search.

### 1 Introduction

Data is a vital asset for business and educational enterprises. In house storage and maintenance of such assets have an impact on the bottom line of enterprises [7]. Database as service (DAS) is marketed as an outsourcing solution that can reduce the total cost of ownership of these assets. DAS allows for the full utilization of databases with professional support and maintenance by these service providers [3]. This brings up a related issue: storage of sensitive data at the providers' site may jeopardize the security of the stored information. One solution, then, is to store the data in encrypted form, and have clients download the relevant encrypted tables from the remote database, decrypt and do query processing [2]. This process expends bandwidth and client resources for decryption and intermediate storage.

Researchers have proposed various solutions to split the work between the client and server sites, while addressing the privacy concerns [4], [8-9]. This paper is a continuation of work done by other researchers. We propose a new algorithm,

Binary Query Bucketization (BQB), that seeks to improve the precision of querying encrypted data sets, increase the utilization of server side processing and reduce the computation and memory utilization on the client side. We evaluate the performance of our approach with other researchers.

The paper is organized as follows. Section 2 reviews work done by others. Section 3 introduces our algorithm, BQB. Section 4 addresses performance comparison with other algorithms. Concluding remarks appear in Section 5.

### 2 Related Work

One approach to querying encrypted data from untrusted sites is Bucketization [10]. It divides a column in a table into several buckets where each bucket has an ID and a range that defines the minimum and maximum values in this bucket. The client holds this indexing information, which identifies the *range* of each of the buckets at DAS, while the actual data, in encrypted form, is stored at DAS (also known as server).

We use an example to illustrate Bucketization. Suppose the database is about student GPA as shown in Table 1, and we use two buckets. Suppose range of Bucket 1 covers GPA  $\leq 1$ , while bucket 2 covers the rest. For a query on students with GPA 2 or better, the client consults the local index and asks the server to send just bucket 2; it then decrypts the data and responds to the user query. However, a query for GPA  $< 2$  will mean downloading, and decrypting all of the two buckets, buckets 1 and 2. In either case, the client has to filter out *false positives* – unwanted data that is outside of the query range. Thus, the technique may lack precision when the queried

data is not on the buckets boundaries. False positives help measure the precision of queries when dealing with encrypted databases. Other researchers have focused on ways to reduce the number of false positives while maintaining the privacy of data.

Query Optimal Bucketization (QOB) [1], is predicated on the assumption that queries are uniformly distributed. By associating a cost to each bucket (see below), QOB spreads the data over the, given  $M$ , buckets in an attempt to reduce the number of false positives while minimizing the total cost of buckets. The algorithm has two parts.

Consider again the GPA information in Table 1. The first part of QOB builds a frequency table to house each distinct GPA value and its frequency; for example, an entry in the frequency table is (1.5, 3) since there are 3 students with GPA 1.5.

QOB finds optimal bucket boundaries in a set of values  $V$ ,  $V = \{v_1, \dots, v_n\}$ , using at most  $M$  buckets. Boundaries of QOB are identified by finding the minimum Bucket Cost (BC) for each bucket, with each bucket holding holds values in the range  $[v_i, v_j]$ . BC is defined as:

$$BC(i,j) = (v_j - v_i + 1) * \sum f_i$$

where  $f_i$  is the frequency of each value in the range. As shown in Table 2, the cost for a bucket that stores all GPAs between [0.7 to 1.2] is  $BC(5 - 1 + 1) * 6 = 30$ .

The second part builds the optimal bucket partitions. This is achieved by attempting all possible data distributions across the  $M$  buckets and computing cost for each distribution, and finally choosing the distribution that yields the least cost. Suppose  $M = 2$ . Then, try out all possible distributions between the two buckets: [(0.7-0.8), (0.9-3)], [(0.7-0.9), (1.0-3)], and so on. The cost for the first distribution is easily be shown to be  $BC(2 - 1 + 1) * 2 + BC(7 - 3 + 1) * 12 = 4 + 60 = 64$ . It can also be readily shown that the least cost distribution is given by Table 3 (see [1] for complete details).

Deviation Bucketization (DB) [5] extends QOB by further reducing false positives, but at the expense of additional buckets: while QOB uses  $M$  buckets, DB needs at most  $M^2$  buckets. Intuitively, the number of false positives will decline as it is inversely proportional to the granularity of the bucket values.

DB is divided into three steps. In Step 1, the QOB bucket output is generated and named as first level buckets, and mean value for each bucket is also computed. Step 2 computes a deviation array that captures the deviation of each data value from the mean found in Step 1. In Step 3, QOB is applied again, but to the deviation array of Step 2 to yield new second level buckets. Unlike Step 1, Step 3 does not use the frequency information for each data value. An example of the second level table ranges for the GPA data is shown in Table 4; as compared to QOB, DB needs three additional, second-level, buckets.

The second level buckets repartition each bucket (of DB) into at most  $M$  additional buckets based on how far a value in the original bucket deviates from the mean. Said another way, while QOB may hold low and high frequency data in the same bucket, DB tries to split them – recognizing that high frequency ones are more likely to be queried -- to permit a reduction in the number of false positives. Similar to QOB, DB keeps index information at the client. Additional details of DB may be found in [5].

### 3 Binary Query Bucketization

The advantage of DB is that the number of false positives decline as each bucket becomes more granular. But the down side is more client storage needs, and granular data has more privacy issues than less granular data. In this section, we propose an improved algorithm, Binary Query Bucketization (BQB).

Like QOB and DB, BQB stores DAS data in encrypted form along with the bucket ID. For example, EncryptionOfStudentGPAInformation ("0.7, 2121212, John") yields ("sldfjkl23k4jl234jklkj23l4kj23l4kj23lk4j"), as shown by the DAS data in Table 5.

While QOB and DB need only the first two attributes of this table, BQB [11] maintains an extra plaintext attribute, autoID for each tuple, as shown by the third column of Table 5. The AutoID field is not related to the bucket ID; it can be generated either dynamically by creating a view that has an auto number combined with the encrypted table, or by adding the auto number with the encrypted table when uploading the encrypted data. AutoID is just a monotonically increasing number assigned with

each tuple and has *no* relation to the actual data contained in that tuple. The client side index is the same as that of QOB. We also note that BQB employs  $M$  buckets, and adopts the same strategy as QOB in determining the bucket distribution. Now, we are ready to discuss the details of BQB.

Our algorithm, shown in Figure 1, attempts to reduce the number of false positives by doing a binary search on the encrypted data housed at DAS. Suppose the user's query range is  $< V$ , i.e. the query retrieves GPA values  $< V$ . Using the client index, the algorithm starts by identifying the buckets that are needed to answer the query. It then constructs a query for the server to fetch two quantities: the min and max AutoID among all of these buckets. For example, if the buckets of interest are the first four buckets, the min will be the AutoID of the first tuple of the first bucket while max will be the AutoID of the last tuple of the fourth bucket. While QOB and DB would have retrieved and decrypted all of the tuples in all of these buckets, BQB retrieves much less, as shown below; for now, it retrieves just these two AutoIDs. For convenience, let us call these two AutoIDs as  $x$  and  $y$ .

This sets in motion a binary search algorithm to focus on the region  $(x, y)$ . The client then asks the server for the encrypted tuple at the midpoint of  $x$  and  $y$ , i.e. tuple  $z$ ,  $z = (x + y) / 2$ . The client decrypts tuple  $z$ , and extracts the data value (GPA), say  $z_v$ , for this tuple. If  $z_v > V$ , then, our new region of interest is  $(x, z)$  else our region of interest is  $(z, y)$ . The binary search continues, in a similar fashion, in this new region. Eventually, the binary search terminates when the region is null. At that point, we would have obtained the AutoID, say  $q_{id}$ , corresponding to the original GPA query value  $V$ .

Armed with  $q_{id}$ , we make one final query to the server to retrieve all tuples with AutoIDs between 1 and  $q_{id}$ . With this, the query is complete, i.e. the query to retrieve GPA values  $< V$  corresponds to the decryption of these tuples whose AutoIDs lie between 1 and  $q_{id}$ .

The number of steps needed to complete the binary search is easily shown to be bounded above by  $\log r$ , where  $r = y - x$ .

#### 4 Performance Results

The dataset we used in this experiment is similar to the one in [1] and [5]; it contains more than  $5 \times 10^5$  tuples (576,097 to be exact, for a total of 35 Mb of non-encrypted data) taken from "Forest CoverType" Archive database [6] with the actual data values ranging from 1 to 360. We ran the three algorithms with 50 different values for  $M$ , the number of buckets, varying  $M$  from 4 to 54. Each run was repeated for 1000 queries.

We stored the DAS data in encrypted form, but for brevity we discuss the results for non-encrypted DAS storage. At the end of the section, we remark on the performance for encrypted DAS storage.

We measured three quantities to characterize the performance of the three algorithms:

- number of false positives;
- size of superset – total number of tuples returned to the client. It includes data within the query range, and false positives or data outside of the query range; and
- turnaround time – elapsed time, from start to finish.

The number of false hits for the three algorithms is depicted in Figure 2. Note that for  $M = 4$ , this number is about sixty-eight thousand for QOB, and around sixteen thousand for DB. With BQB, the number of false hits is no more than 20. The large number of hits for the former two algorithms may be attributed to their inability to retrieve single tuples.

Figure 3 represents the size of the superset, the volume of the number of tuples retrieved for each of the three schemes. As seen in Figure 3, the size of superset for QOB, DB and BQB are 358,403, 306,430, and 289,710 respectively, for  $M = 4$ . Note that the superset counts the total number of tuples retrieved from DAS, those that don't match the user's query – false positives – and those that do. Relating Figure 3 to Figure 2 can provide insight into error rate – the percent of records that don't match user's query; for example, error rates for QOB and DB are 20% and 5% ( $70,000 / 358,000 = 20\%$ ;  $16738 / 306430 = 5\%$ ), while for BQB the error rate is close to 0 ( $18 / 289710$ ), all for  $M = 4$ .

As noted in [1], DB performs better than QOB since it employs more buckets with finer partitioning. Consider the case where data values are whole numbers in the range 1 to 360. The maximum number of buckets that can be

produced in this range is  $360/2$  or 180 buckets. Thus, when 13 buckets are used by QOB, DB would employ up to  $169 (= 13^2)$  buckets, and DB would reach the maximum number of 180 buckets when QOB's bucket use goes up by one more. However, privacy may be a concern for DB since the data range is much smaller, and there is not much that separates the buckets or the values inside of these buckets. For example the bucket partitioning looks like [1~2], [3~4], [5~6], and so on. Of course, BQB does not exhibit this problem.

Figure 4 compares the mean turnaround time for various bucket sizes. For smaller bucket sizes -- 4 through 12 -- we deployed a sequential version of the three algorithms, and a parallel -- or multiple process -- version for larger bucket sizes. As shown in Figure 4, BQB has a slower turnaround than either of DB or QOB; for  $M = 4$ , the QOB, DB and BQB turnaround times are 1, 1 and 3 seconds, and for  $M=20$ , these values are 43, 76 and 93 seconds. Slower turnaround time of BQB can be attributed to the network delay experienced during the binary search process when BQB retrieves single tuples.

Figure 5 shows the size of the returned false hits in mega bits; while this information can also be extrapolated from Figure 2, it nevertheless provides an easy handle on the bandwidth consumption for the three algorithms.

### Encrypted DAS

For brevity, we did not show the performance results when DAS data is stored in encrypted form. Decryption is quite time consuming/expensive, and is an added penalty for *all* of the three schemes. Since the other two schemes have to do lot more decryption than BQB (see Figure 2, for the number of false hits), they take many hours to complete the decryption process for large datasets, such as the one noted

in the beginning of this Section. So, we use a smaller, but *encrypted* data set to illustrate this point, see Figure 6 for an example. As one would expect, BQB terminates much earlier than the other two algorithms.

Recall that the number of false positives with BQB is bounded above by  $\text{Log } r$ , where  $r$  equals the range of the user's query. Thus, the overall performance of BQB with encrypted storage at DAS should be significantly better than that of the other two algorithms. In fact, when decryption is factored in, the small penalty due to multiple client requests - as experienced by BQB's binary search - is more than offset by the gain in significantly fewer decryptions that are needed for the proposed algorithm.

### 5 Concluding Remarks

In this paper we proposed a new algorithm for querying encrypted data that is stored at untrusted servers. Our approach is a variant of a scheme used by others. The novelty of our scheme is that it employs a new, binary search step to precisely determine the range of the actual data that is relevant to the user's query. During the binary search process we decrypt one tuple at a time instead of doing bulk decryption of, often unwanted data -- as is common with other approaches -- and discarding it later.

Following the binary search, BQB retrieves the relevant data in bulk and decrypts it. Since all of this data correspond to the user's query, no false positives are generated following the binary search process. This results in a significant reduction both in data transmission, and the number of decryptions at the client. However, the number of client-server interactions, while bounded above by  $\text{Log } r$ , is a bit higher with the proposed scheme as compared to the other two schemes.

GPA	SSN	Name
0.7	2121212	john
0.8	6545545	mike
0.9	5121123	rob
1.0	5482123	tom
1.0	2384865	steve
1.2	5315422	ali
1.5	6689555	ahmed
1.5	5165888	adam
1.5	9954521	cres
3.0	5458862	cris
3.0	7148787	rob
3.0	2342336	mike
3.0	3334445	rich
3.0	2222234	amanda

Table 1: GPA Data

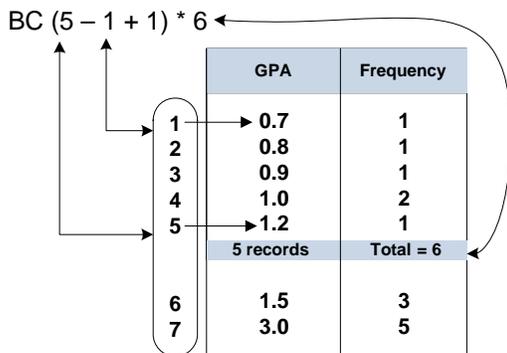


Table 2: Computation of BC [1]

GPA	ID
[0.7 ~ 1.0]	Bucket_1
[1.2 ~ 3.0]	Bucket_2

Table 3: QOB Bucketization (M = 2) – Index Information

**Algorithm BQB (V)**  
**Input:** Required value to search for  
**Output:** Query result with zero false positive.  
 Select all buckets from QOB table > or < V  
 Construct select statement to retrieve (Max, Min) AutoID and their encrypted records  
**if** (returned set != user query)  
     Mid = Min + ((Max-Min)/2)  
     Query="Select Etuple From ETable Where AutoID= Mid"  
     Decrypt = Decrypt (Query) → extract V  
**While** (Max > Min)  
     **If** (Decrypt > V)  
         Max = Mid - 1  
     **else**  
         Min = Mid + 1  
     Mid = Min + ((Max-Min)/2)  
     Query="Select Etuple From Etable Where AutoID= Mid"  
     Decrypt = Decrypt (Query) → extract V  
**End While**  
**end if**  
 Return: Select eTuple From Etable Where AutoID > or < Mid

Figure 1: Binary Query Bucketization – Binary Search

Partition	BID
[0.7 ~ 0.8]	Bucket_1_1
[0.9 ~ 1.0]	Bucket_1_2
[1.2 ~ 3.0]	Bucket_2_1

Table 4: Second Level Bucketization

Bucket	eTuple	AutoID
Bucket_1	Sdd342kjk23ksdfsdfsdfsd234sdf453453ed	1
Bucket_1	234lj2kdjsfi33345345ergfdgdfgdfyjkklj5	2
Bucket_1	Pqasdososoifwgksdjkhlkerkjldkfjleieywhsdf	3
Bucket_1	2#Kskdjkskqwsrjk34j5k0s9fksdjf09345jkdf	4
Bucket_2	si4@sdkjweqq345jfdg09345kjdfg0;lskfgkfg	5
Bucket_2	Sidfjkeierjek334e34jk509345098dfjkfg;pwj	6
.....	.....	...

Table 5: DAS Encrypted Table

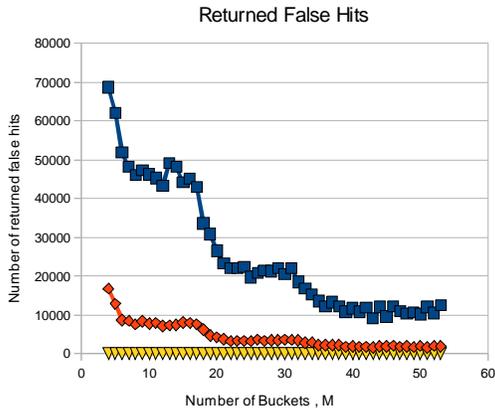


Figure 2: False Positives

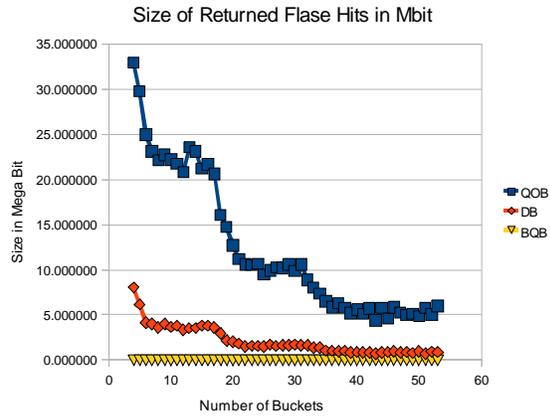


Figure 5: Bandwidth Usage

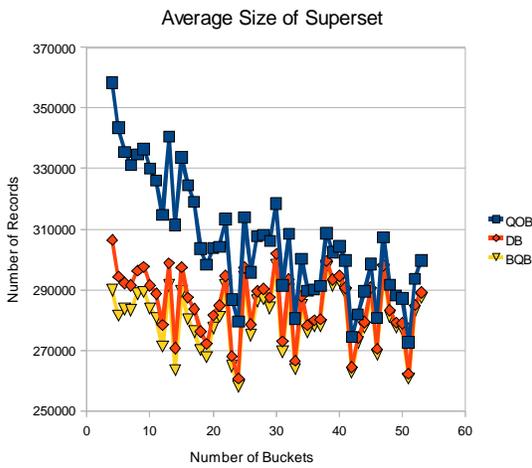


Figure 3: Number of Tuples Processed by the Client

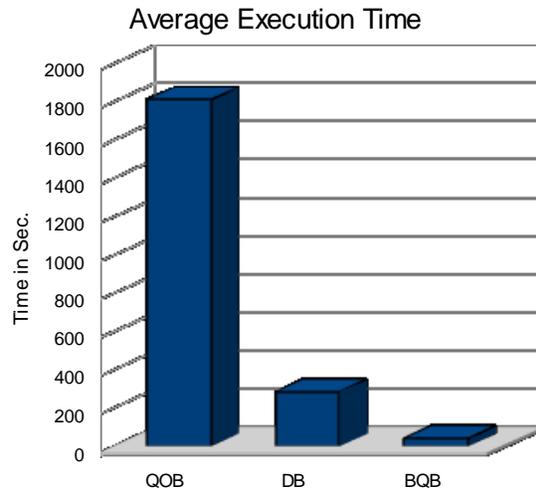


Figure 6: Performance on Encrypted Data - Mean Turnaround Time.

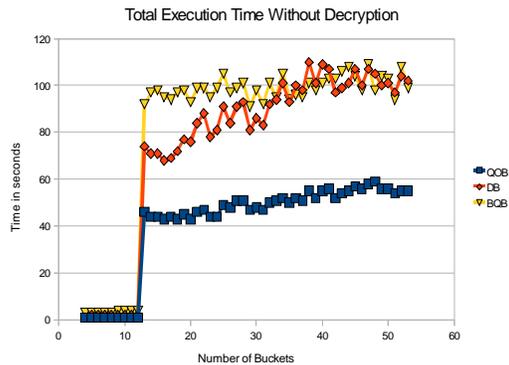


Figure 4: Process Turnaround Times

## Reference

1. Hore, B., Mehrotra, S., Tsudik, G.: A Privacy-Preserving Index for Range Queries. In: International Conference on Very Large Data Bases, pp. 720–731, 2004.
2. Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y.: Order Preserving Encryption For Numeric Data. In: Book Order preserving encryption for numeric data, Series Order preserving encryption for numeric data, ed., Editor ed.^eds., pp. 563-574, ACM, 2004.
3. Hacigumus, H., Iyer, B., and Mehrotra, S.: Providing Database as a Service. In: IEEE International Conference on Data Engineering (ICDE), San Jose , California , 2002.
4. Mykletun, E., Tsudik, G.: Incorporating a Secure Coprocessor in the Database-as-a-Service Model. In: International Workshop on Innovative Architecture for Future Generation High Performance Processors and Systems, 2005.
5. Yvonne Y., and Huiping G.: An Improved Indexing Scheme for Range Queries. In: International Conference on Security and Management(SAM'08)", Las Vegas, 2008.
6. The UCI KDD Archive. Forest CoverType Database  
<<http://kdd.ics.uci.edu/databases/covertime/covertime.html>>
7. Ozcelik, Y., Altinkemer, K. : Impacts of Information Technology (IT) Outsourcing on Organizational Performance: A Firm-Level Empirical Analysis. In: 17th European Conference on Information System, 2009.
8. Haber, S., Horne, W., Sander, T., Yao, D.: Privacy-Preserving Verification of Aggregate Queries on Outsourced Databases. In: Technical Report HPL-2006-128, HP Labs, 2006.
9. Damiani, E., Vimercati, S., Jajodia, S., Paraboschi, S., Samarati, P.: Balancing Confidentiality and Efficiency in Untrusted Relational DBMSs. In: Proc. 10th ACM Conf. On Computer and Communications Security, Washington, DC, pp. 93-102, 2003.
10. Hacigümüs, H., Iyer, B., Li, C., Mehrotra, S.: Executing SQL Over Encrypted Data in the Database-Service-Provider Model. In: SIGMOD Conference, pp. 216-227, 2002.
11. Alwarsh, M.: An Improved Algorithm for Querying Encrypted Database. Bowling Green State University, Master's project, 2010.

# A Comparative Study Of Two Symmetric Encryption Algorithms Across Different Platforms.

Dr. S.A.M Rizvi<sup>1</sup>, Dr. Syed Zeeshan Hussain<sup>2</sup> and Neeta Wadhwa<sup>3</sup>  
Deptt. of Computer Science, Jamia Millia Islamia, New Delhi, India

**Abstract :** *The world of digital communications is expanding day by day, For secure communications over the unsecure mediums, Cryptography plays a crucial role and Symmetric Encryption algorithms do the real part of encoding data before transmission. The deep analysis of their security and speed become the necessity of safe digital communication. In this paper, we study the two popular symmetric cryptographic algorithms BLOWFISH and CAST. We analyze their security issues and then compare their efficiency for encrypting text, image and sound with the official encryption standard AES(Advanced Encryption Standard) across different widely used Operating Systems like Windows XP, Windows Vista and Windows 7. The simulation results reveal Which algorithm performs better on Which Operating system for encrypting What kind of data.*

**Keywords:** BLOWFISH, CAST, Symmetric Encryption.

## 1 Introduction

Cryptography is the art and science of encoding data so that it can travel to any place without threat of being theft in the way. This science is basically categorized in two categories : Symmetric and Asymmetric. Symmetric Cryptography is to encode and decode data with one and the same key whereas Asymmetric Cryptography works with a pair of key, one key is to encode the data and with the other we can decode. Further Symmetric Ciphers are of two kinds : Block and Stream. Block ciphers encrypts a fixed block of bits at a time and Stream ciphers encrypts bit by bit. Block ciphers are one of the fundamental building blocks for cryptographic systems[1].

The life cycle of Symmetric Cryptography mainly starts from the birth of DES(Data Encryption Standard). In 1977 ,IBM's submission LUCIFER (Feistel), adopted as DES[2] by NBS(National

Bureau of Standards) now NIST( National Institute of Standards and Technology). Since its evolution, many cryptanalysts have attempted to break it and finally DES encrypted message was cracked in only 22 hours in 1998. Then again there was a need of a new encryption standard, NIST put out a call and 15 algorithms were selected as the first round finalists ,CAST was one of them. In second round out of 5 one was Twofish which is the descendant of Blowfish.

Finally , Rijndael won the competition and become AES(Advanced Encryption Standard) in 2001[3].

In this paper we do the comparative analysis of Blowfish and CAST with the AES on different latest platforms like Windows XP, Windows Vista and Windows7. This analysis shows which algorithm is best suited in which environment.

The rest of the paper is organized as follows: section II gives the brief review of the algorithms and discuss their security issues; section III outlines the related work ; section IV describes the implementation details and shows the simulation results; finally, the conclusions and future work is followed in section V.

## 2 Overview of Algorithms

### 2.1 Blowfish

Blowfish was designed in 1994 by Bruce Schneier, it works on 64-bit units with key lengths from 32-bits up to 448-bits [4] .Each 64-bit block is divided into two 32-bit words, it encrypts every block by performing 16 rounds of encryption. Basically the algorithm consists of two parts: a key-expansion part and a data- encryption part. Key expansion converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes. The time-consuming subkey-generation process adds considerable complexity for a brute-force attack. The subkeys are too long to be

stored on a massive tape, so they would have to be generated by a brute-force cracking machine as required.

Exhaustive search of the keyspace could be the effective way of breaking it, because designer himself admit the existence of weak keys. But so far no one has succeeded in breaking full strength Blowfish encryption. It is unpatented and license-free, means Blowfish is a fast, secure and free alternative encryption method.

## 2.2 CAST

CAST is the first round finalist of AES competition. It is developed by Carlisle Adams and Stafford Taveres in Canada, it uses 64-bit block for 64-bit and 128-bit key size variants and 128-bit block sizes for the 256-bit key version. The complete specification of CAST algorithm is given in [5].

It uses an f-function that produces a 32-bit output from a 32-bit input, and each round consists of modifying one 32-bit quarter of the block by XORing it with the f-function of another 32-bit quarter of the block. There are 48 rounds in total, which are organized in groups of four, called quadrounds. Encryption begins with six forwards quadrounds, and then continues with six reversed quadrounds, which are reversed exactly as would be necessary for decryption. Means, for decrypting data, it is only necessary to change the order in which the subkeys are used.

CAST cipher can be broken up to only 5-round. However, if the degree of the round function is lower, the CAST cipher could be broken up to more number of rounds[6]. CAST encryption procedure has been under rigorous analysis among cryptanalysts for the last 10 years. Minor weaknesses have been found like non-surjective attack, HOD attack but nothing extendable beyond 5-6 rounds.

## 2.3 AES

AES has Non-Feistel structure, based on a sophisticated mathematical design. It's simple structure attracts cryptographers and cryptanalysts. It encrypts 128 bit block size with 128/192/256 bit key for 10/12/14 rounds.

The complete specification and the above structure of AES encryption scheme is given in [3]. No one can

break it beyond 5-6 rounds with today's computational power.

## 3 Related Work

In research of [7-8] CAST ciphers with random S-boxes are proposed. It is shown that when randomly generated S-boxes are used, the resulting cipher is resistant to both differential and linear attack.

A Crypto++ Library [9] analyze some common encryption algorithms. It showed that Blowfish and AES have the best performance compared with other encryption algorithms.

Nadeem and Kader, did performance evaluation of few symmetric encryption algorithms like AES, DES, and 3DES, RC6, Blowfish and RC2. They concluded from the simulation results that Blowfish has better performance as compared to other encryption algorithms for different file size, followed by RC6. AES has better performance than RC2, DES, and 3DES. 3DES still has low performance compared to algorithm DES. RC2 is the slowest. However they conducted the experiments on only one platform: Windows OS[10-11].

Krishnamurthy in [12] demonstrated the energy consumption of different common symmetric key encryptions on hand-held devices.

Salama and Elminaam have done a comparison between encryption algorithms (AES, DES, and 3DES, RC2, Blowfish, and RC6) at different settings like different sizes of data blocks, different data types, CPU time, and different key size. The algorithms were tested on two different hardware platforms. The results indicated that the Blowfish had more efficient compared to other algorithms. And AES had a better performance than 3DES and DES[13].

The study in[14] tested the encryption algorithms such as RC4, AES and XOR to find out the overall performance of real time video streaming. The results showed that AES has less time overhead than the overhead using RC4 and XOR algorithm. So, AES is more efficient to secure real time video transmissions.

Most of the above parallel research focus on performance analysis of different symmetric encryption algorithms on different settings for various kinds of input data with different modes. In this paper, we are analyzing Blowfish and CAST on 3 different Operating Systems for encrypting 3 kinds of data :text, image and sound.

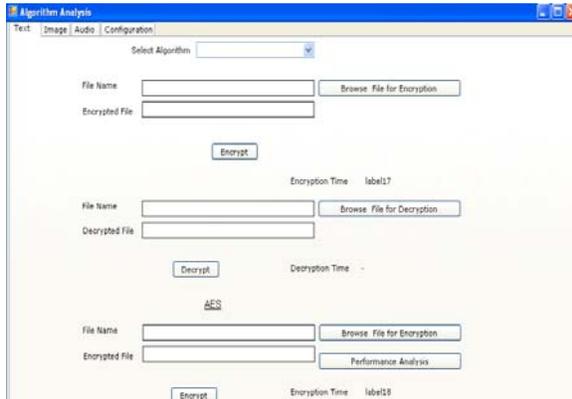
### 4 EXPERIMENTAL DESIGN

We implemented the algorithms according to their standard specifications in .Net environment using C#, and a tool has designed, which calculates the encryption time in ms(milli seconds) of each algorithm .The no. of different types of files like text file, images and audio files have been encrypted with the designed tool and their execution time is calculated.

For our experiment, we use three laptops of 32bit configuration:

1. Intel Pentium® Dual Core with Windows XP.
2. Intel Pentium® Dual Core with Windows Vista.
3. Intel Pentium® Dual Core with Windows 7.

The tool's front end look like as:



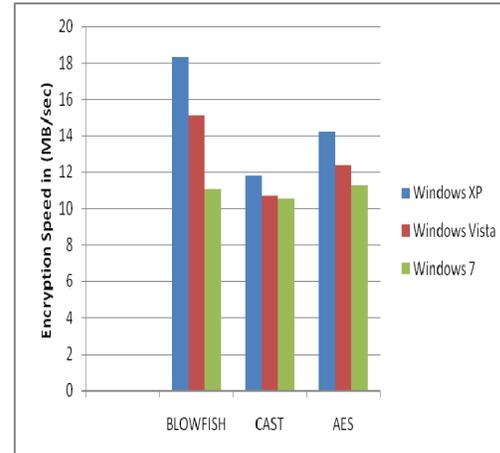
We encrypt 60 text files of size ranges between 500KB to 50MB, 60 images ranges between 20 KB to 200KB, 60 audio files ranges between 2- 50MB. First we tabulated their encryption time in ms(milli seconds) and then calculated their mean execution speed in MB/sec (MegaBytes per second) .

**Table 1:** Encryption Speed ( in MB/sec) of BLOWFISH, CAST and AES on different OS for text data

OS →	Win XP	Win Vista	Win 7
Encryption ↓			
BLOWFISH	18.3	15.1	11.1

CAST	11.8	10.7	10.5
AES	14.2	12.4	11.3

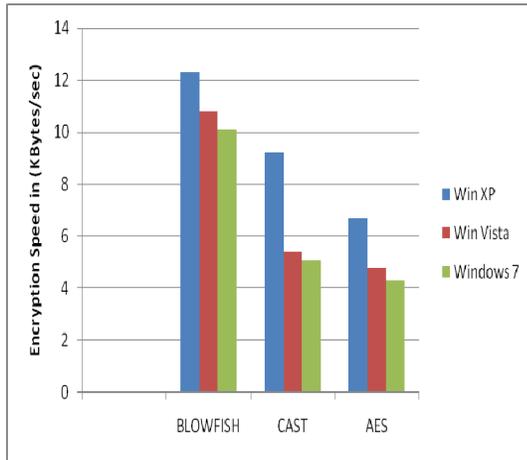
**Figure 1:** Execution speed for encrypting text data: Comparison between different OS



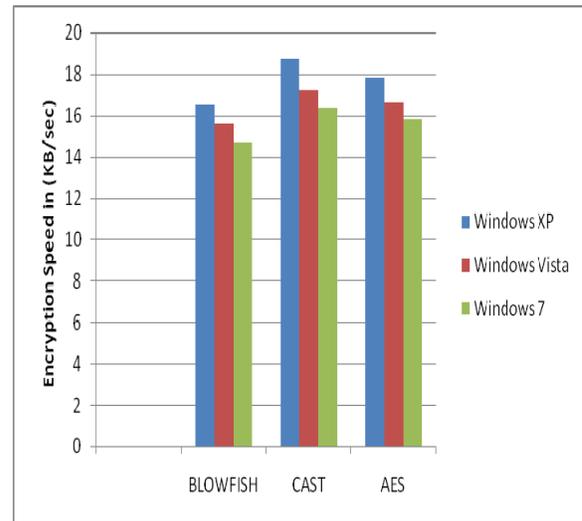
**Table 2:** Encryption Speed ( in KB/sec) of BLOWFISH, CAST and AES on different OS for image data

OS →	Win XP	Win Vista	Win7
Encryption ↓			
BLOWFISH	12.3	10.8	10.1
CAST	9.2	5.4	5.1
AES	6.7	4.8	4.3

**Figure 2:** Execution speed for encrypting image data: Comparison between different OS



**Figure 3:** Execution speed for encrypting audio data: Comparison between different OS



**Table 3:** Encryption Speed ( in KB/sec) of BLOWFISH, CAST and AES on different OS for audio data

OS →	Windows XP	Win Vista	Win 7
Encryption ↓			
BLOWFISH	16.5	15.6	14.7
CAST	18.7	17.2	16.4
AES	17.8	16.6	15.8

## 5 Conclusion

### For Text data :

All algorithms run faster on Windows XP , but Blowfish is the most efficient and CAST runs slower than AES.

### For Image data :

Blowfish encrypts images most efficiently on all 3 platforms, even CAST runs faster on Windows XP than AES. But on Windows Vista and Windows7, AES and CAST perform at the similar speed .

### For Sound data:

CAST performs better than BLOWFISH and AES on Windows XP for encrypting audio files, but on Windows Vista and Windows7, there is no significant difference in performance of CAST and AES, however BLOWFISH encrypts audio files at less speed.

In future , we try to incorporate good features of BLOWFISH and CAST in a single algorithm, which can perform well on all latest platforms for all types of data.

## References

- [1] B.Schneier, *Practical Cryptography*, Wiley, 2003.
- [2] W. Diffie, M. Hellman, "Exhaustive cryptanalysis of the NBS data encryption standard," *Computer*, p. 74-78, June 1977.
- [3]. J. Daemen and V. Rijmen, " AES Proposal: Rijndael" ,1999.

- [4]. B. Schneier, "The blowfish encryption algorithm -one year later," Dr. Dobb 's Journal, 1995.
- [5]. C.M.Adams, "The CAST-128 Encryption Algorithm," Request for Comments (RFC) 2144, NetworkWorking Group, Internet Engineering Task Force, May, 1997.
- [6]. Shiho Moriai, Takeshi Shimoyama, "Higher Order Differential Attack of a CAST Cipher", S. Vaudenay (Ed.): Fast Software Encryption FSE'98, LNCS 1372, pp. 17-31, 1998, Springer-Verlag Berlin Heidelberg 1998.
- [7]. H.M.Heys and S.E.Tavares, "On the security of the CAST encryption algorithm" ,Canadian Conference on Electrical and Computer Engineering, pp.332-335, 1994.
- [8] J. Lee, H. Heys, and S.Tavers, "Resistance of a CAST-like Encryption Algorithm to Linear and Differential Cryptanalysis", Designs, Codes, and Cryptography, vol. 12,no.3,pp.267-282,1997.
- [9]. Results of Comparing Tens of Encryption Algorithms Using Different SettingsCrypto++ Benchmark, Retrieved Oct. 1, 2008.
- (<http://www.eskimo.com/weidailbenchmarks.html>).
- [10]. A. Nadeem and M. Y. Javed, "A performance comparison of data encryption algorithms,"Information and Communication Technologies, ICICT 2005, pp.84-89, 2005.
- [11]. W.S.Elkilani, "H.m.Abdul-Kader, "Performance of Encryption Techniques forReal Time Video Streaming, BIMAConference, Jan 2009, PP 1846-1850.
- [12]. N. Ruangchaijatupon and P. Krishnamurthy, "Encryption and power consumption in wireless LANs-N,"The Third IEEE Workshop on Wireless LANs, pp. 148-152,Newton, Massachusetts, Sep. 27-28,2001.
- [13] D. Salama, A. Elminaam and etal, "Evaluating The Performance of Symmetric Encryption Algorithms", International Journal of Network Security, Vo1.10, No.3, PP.216-222, May2010.
- [14] W.S.Elkilani, "H.m.Abdul-Kader, "Performance of Encryption Techniques forReal Time Video Streaming, BIMAConference, Jan 2009, PP 1846-1850.



## **SESSION**

# **SECRECY METHODS AND RELATED ISSUES + CRYPTOGRAPHY + CRYPTOSYSTEMS + WATERMARKING**

**Chair(s)**

**TBA**



# Cryptanalysis on the RFID ACTION Protocol

H.M. Sun<sup>1</sup>, S.M. Chen<sup>1</sup>, and K.H. Wang<sup>2</sup>

<sup>1</sup>Department of Computer Science, National Tsing Hua University, Taiwan

<sup>2</sup>Hong Kong Institute as Technology, Hong Kong

**Abstract**—*There are increasing concerns on the security of RFID usages. Recently, Lu et al. presented ACTION, a privacy preservative authentication protocol for RFID. It is claimed that it achieves high level of security even if a large number of tags is compromised. However, we found that this protocol is vulnerable to two severe attacks: Desynchronizing attacks and Tracking attacks. In this paper, we present how these two attacks work even if the protocol parameters are moderated.*

**Keywords:** Desynchronizing attack, Privacy, RFID, Tracking attack

## 1. Introduction

Radio Frequency IDentification (RFID) systems have drawn increasing concerns in recent years. It is designed to replace traditional barcode systems for its shorter access time, longer reading distance and rewritable property. Usually, a RFID system is composed of three components: RFID tags, a back-end server to maintain the data related to tags, and a RFID reader that connecting with a back-end server. Nevertheless, due to the benefit of contactless sensing, if the tags are not protected with proper security measures, it may arise security or privacy problems that does not happen in barcode systems [1], [2], [3], [4], [5], [6], [7], [8], [9]. For example, the information of a tag-attached medicine container will reveal the treatment applied to the patient; tag-attached underwear will disclose its size or color. The importance of protect tag's privacy is obviously. In particular, there are two severe attacks launched on RFID tag that should be aware. They are *Tracking Attacks* and *Desynchronizing Attacks*.

*Tracking Attacks:* Due to the contactless access property of RFID systems, a RFID reader can freely scan a tag if the tag is not protected properly. If a tag always responses the same data when it is queried, it would arise privacy problem where the attacker could know the existence of the tag. In other words, the tag is tracked by an attacker. For example, when a tag attaching to a book borrowed from a library may track by an attacker, the privacy of the borrower would be invaded. Another example is that if a thief can track a tag-attached valuable item stored inside a public locker, then the thief would break the locker and steal that item.

*Desynchronizing Attacks:* In order to communicate securely, a tag and the back-end database are required to synchronize an authentication key. In a desynchronizing

attack, the attacker attempts to interfere the communication between a tag and a reader so that the keys stored in the database and the tag are out-of-sync. As a result, the reader will no longer be able to read the tag correctly. This attack is severe in some applications. For example, when a tag attached to a vulnerable item in a shop is desynchronized with the reader, a thief can steal it without triggering the RFID security alarm system.

There is a simple solution to protect the system from tracking attacks [2]. However, it requires  $O(N)$  searches in the back-end database where  $N$  is the number of tags. Therefore, it is inefficient in a large scale RFID system. Some research have been proposed to provide relatively lower search complexity to protect the system [4], [5], [6], [7]. In 2009, Lu et al. proposed an ACTION protocol [1] that provides  $O(\log N)$  search complexity in the back-end database. In addition, it maintains tag privacy even if a large number of tags is compromised. However, we found that the ACTION protocol may suffer tracking attacks. An attacker can track a tag without even compromising any tag. The authors also proposed a mechanism to update the key stored in the back-end server and the tag. However, we found that this key update mechanism is vulnerable to a desynchronization attack.

The rests of this paper are organized as follows. We review the ACTION protocol in Section 2 and analysis the ACTION protocol in Section 3. The solutions are proposed in Section 4. Finally, a conclusion will be given to wrap up the paper.

## 2. Background

In this section, we review how the ACTION protocol works. The ACTION protocol involves of four phases: *system initialization*, *tag identification*, *key-updating*, and *system maintenance*. We will describe these four phases in the following subsections. The notations used in the ACTION protocol are listed in Table 1 while the authentication processes of the ACTION protocol are shown in Figure 2.

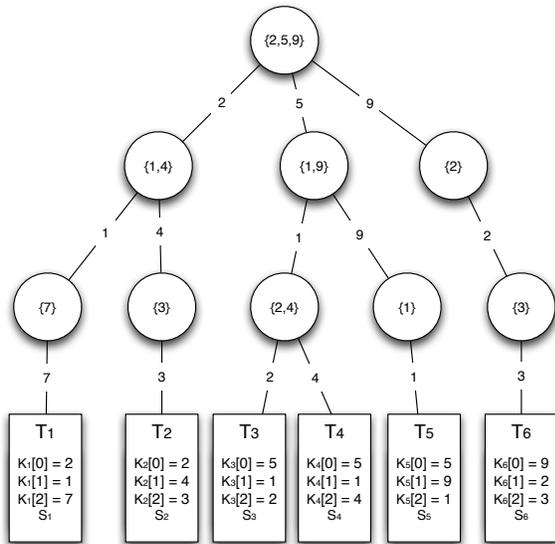
### 2.1 System Initialization

A RFID tag  $\mathcal{T}$  and a reader  $\mathcal{R}$  share two keys: a path key  $k_i$  and a secret key  $s_i$ .<sup>1</sup> When the system is initialized,

<sup>1</sup>It is assumed that the reader and the back-end database have real time secure connection in the ACTION protocol. Thus the role of the database is diminished in the system.

Table 1: Notation Table

Symbol	Description
$\mathcal{R}$	A RFID reader
$\mathcal{T}$	A RFID tag
$n_i$	Nonce $i$
$l_n$	Length of $n_i$
$\mathcal{H}()$	One-way hash function
$l_h$	Length of the output $\mathcal{H}()$
$k_i$	Path key of $\mathcal{T}$
$k_i[x]$	The $x$ -th subkey of the path key.
$l_k$	Length of a subkey
$s_i$	Secret key of $\mathcal{T}$
$\delta$	Branching factor of the key tree, i.e., the largest possible degree of each node in the key tree.
$T$	A key tree stored in the database.
$d$	The depth of the key tree
$c$	The number of times that <i>TagJoin</i> runs

Fig. 1: An example of key tree  $T$ .

a  $d$ -level key tree  $T$  is built such that each tag is pseudo-positioning at a leaf node of the tree as shown in Figure 1. Each node of the tree has a  $l_k$ -bits index. No node with duplicated indices is allowed under the same parent node. Assume that a tag  $\mathcal{T}$  has a path  $P$  from the root node to the leaf node. The path key  $k_i$  for  $\mathcal{T}$  is the concatenation of the indices of the nodes on  $P$ . We denote the path key  $k_i = k_i[0]||k_i[1]||\dots||k_i[d-1]$  such that each  $k_i[x]$  is a subkey of the path key, where it is also the index of a node on  $P$ . For example, in Figure 1, the path key of tag  $T_1$  is  $(2||1||7)$ , where  $k_i[0] = 2$ ,  $k_i[1] = 1$ ,  $k_i[2] = 7$ . The branching factor  $\delta$  is denoted as the maximum branching number of the node where  $\delta = 2^{l_k}$ . The secret key  $s_i$  is a random generated key with length  $l_k \times d$ .

1.  $\mathcal{R} \rightarrow \mathcal{T}$  : Request,  $n_1$ ;  $n_1 \leftarrow^R \{1\}^n$
2.  $\mathcal{T}$  :  $U = \{n_2, \mathcal{H}(n_1||n_2||k_i[0]), \mathcal{H}(n_1||n_2||k_i[1]), \dots, \mathcal{H}(n_1||n_2||k_i[d-1]), \mathcal{H}(n_1||n_2||s_i)\}; n_2 \leftarrow^R \{1\}^n$
3.  $\mathcal{T} \rightarrow \mathcal{R}$  :  $U$
4.  $\mathcal{R}$  :  $\{i, k_i, s_i\} \leftarrow \text{Identify}(U)$
5.  $\mathcal{R}$  :  $\text{TagLeave}(k_i, i)$   
:  $\{k'_i, s'_i, c\} \leftarrow \text{TagJoin}(k_i, s_i, i)$   
: Update  $k_i \leftarrow k'_i, s_i \leftarrow s'_i$
6.  $\mathcal{R} \rightarrow \mathcal{T}$  :  $\sigma = \{c, \mathcal{H}(n_1||n_2||k'_i), \mathcal{H}(n_1||n_2||s'_i)\}$
7.  $\mathcal{T}$  :  $\{k'_i, s'_i, c\} \leftarrow \text{Verify}(\sigma)$   
: Terminate if not verified  
: Update  $k_i \leftarrow k'_i, s_i \leftarrow s'_i$

Fig. 2: ACTION Protocol

## 2.2 Tag Identification

The tag identification is executed when a reader tries to read a tag. This phase includes four steps: (*Step 1*)  $\mathcal{R}$  transmits the message “Request” and a random nonce “ $n_1$ .” (*Step 2*) Once  $\mathcal{T}$  received the messages, it will generate a nonce “ $n_2$ ” and calculate  $d$  times of hash operations as  $\mathcal{H}(n_1||n_2||k_i[j])$ , where  $0 \leq j \leq d-1$ . (*Step 3*)  $\mathcal{T}$  transmits  $U$  to  $\mathcal{R}$ , where  $U$  is the collection of  $n_2$ ,  $d$  hash values and  $\mathcal{H}(n_1||n_2||s_i)$ . (*Step 4*) When  $\mathcal{R}$  receives  $U$  from  $\mathcal{T}$ ,  $\mathcal{R}$  iteratively compares the messages  $\mathcal{H}(n_1||n_2||k_i[j])$  by computing  $\mathcal{H}(n_1||n_2||x)$  where  $x$  is the node index of  $k_i[j]$  for  $0 \leq j \leq d-1$ . The matched value  $x$  will be identified as  $k_i[j]$ . We take  $T_2$  as an example in Figure 1.  $\mathcal{R}$  first computes  $\mathcal{H}(n_1||n_2||x)$ , where  $x$  equals to 2, 5, and 9. Since  $\mathcal{H}(n_1||n_2||2)$  matches with the first received hash value,  $\mathcal{R}$  will move to the node with index 2 on second level, i.e., the left most node containing the indices 1 and 4.  $\mathcal{R}$  will compare all the values iteratively until it reaches to a leaf node. When a leaf node is reached,  $\mathcal{R}$  computes and verifies the message  $\mathcal{H}(n_1||n_2||s_i)$ . If the message is verified, then  $\mathcal{T}$  is authenticated.

## 2.3 Key-updating

Once  $\mathcal{T}$  passed the authentication processes,  $\mathcal{R}$  and  $\mathcal{T}$  will update the path key and the secret key. (*Step 5*)  $\mathcal{R}$  executes *TagLeave* and *TagJoin* functions which will be described in the sub-section of System Maintenance. *TagJoin* function outputs the  $(c, k'_i, s'_i)$  where  $k'_i$  and  $s'_i$  are the new path key and the new secret key, respectively.  $k'_i$  is computed as a series of hash functions by  $c$ ,  $s_i$ , and  $k_i$ , while  $s'_i = \mathcal{H}(n_1||n_2||s_i)$ . Then,  $\mathcal{R}$  updates the secret key as  $s'_i$  and the path key as  $k'_i$ . Notice that the value of  $c$  usually equals to 1 unless a rare scenario happened. (*Step 6*)  $\mathcal{R}$  sends a synchronization message  $\sigma = \{c, \mathcal{H}(n_1||n_2||k'_i), \mathcal{H}(n_1||n_2||s'_i)\}$  to  $\mathcal{T}$ . (*Step 7*) Once obtained  $\sigma$ ,  $\mathcal{T}$  first verifies the message  $\mathcal{H}(n_1||n_2||s_i)$ . Next, it uses  $k_i$ ,  $s_i$ , and  $c$  to generate  $k'_i$  and verifies the message  $\mathcal{H}(n_1||n_2||k'_i)$ . If they are identical,  $\mathcal{T}$  updates its keys by

$k'_i$  and  $s'_i$  to synchronize finishing the key-updating process. It finishes the authentication protocol and  $\mathcal{R}$  may further access the data on  $\mathcal{T}$ .

## 2.4 System Maintenance

This phase is composed of two functions: *TagJoin* and *TagLeave*.

### 2.4.1 TagJoin

It will be executed when keys are updated or a new tag is deployed to the system. This function takes  $k_i$ ,  $s_i$ , and  $i$  as the inputs and produces  $k'_i$ ,  $s'_i$ , and  $c$  as the outputs, where  $i$  is the identity of the tag. When a tag is newly deployed to the system, the inputs  $k_i$  and  $s_i$  will be two random strings with length of  $d \times l_k$  bits.  $\mathcal{R}$  computes  $k'_i = \mathcal{H}(n_1 || n_2 || k_i || s_i)$  and tries to assign the tag to the leaf node pointed by the path key  $k'_i$ . For example, in Figure 1, if a new tag  $\mathcal{T}_2$  is joining to the tree and the path key  $k'_2 = (2||4||3)$  is computed. It transverses from the root node of the tree to the node with index 2. Then, it creates a node with index 4 under that node and transverses to the newly created node. Finally, it creates a node with index 3 under the node with index 4. The leaf node will be assigned to the tag  $\mathcal{T}_2$ . If the leaf node has already taken by an existing tag,  $\mathcal{R}$  regenerates  $k_i^c = \mathcal{H}(n_1 || n_2 || k_i^{c-1} || s_i)$  with ( $k_i^1 = k_i$ ) until the leaf node is newly generated. With negligible probability exceptions,  $c$  usually equals to 1. Finally, it computes  $s'_i = \mathcal{H}(n_1 || n_2 || s_i)$  and assigns  $k'_i = k_i^c$ . The function returns  $k'_i$ ,  $s'_i$ ,  $c$  as the outputs of the function.

### 2.4.2 TagLeave

It is executed when key-update phase took place or when a tag is removed from the system. It takes  $(k_i, i)$  as the inputs of the function. Assume  $\mathcal{T}$  is leaving from the system,  $\mathcal{R}$  will erase the corresponding leaf node from the key tree  $\mathcal{T}$ . Then, it recursively transverses upward from the leaf node and erases its parent node unless the parent node contains more than one child node. For instance, we remove  $\mathcal{T}_2$  from Figure 1. The second left most leaf node, and its parent node which marked with  $\{3\}$  inside the circle will be removed from the tree. The label  $\{1, 4\}$  of left most node at the second level will be changed to  $\{1\}$  as well.

## 3. Cryptanalysis on the ACTION protocol

In this section, we illustrate how the ACTION protocol vulnerable to a tracking attack and a desynchronizing attack. Before describing these attacks, the capabilities of the adversaries should be carefully stated. We assume that an adversary is able to use the wireless channel freely, including eavesdropping, fabricating, interrupting, and modifying messages. An adversary tries to create a counterfeit RFID tag to communicate with a reader and also communicates with a

legitimate RFID tag without the presence from a legitimate RFID reader. Depends on different attacks described below, the adversary has different goals. We say a protocol is vulnerable to some attacks if an adversary can accomplish her goals with non-negligible probability and with limited computation and communication resources.

### 3.1 Active Tracking Attack

An active tracking attacker intends to identify a tag by associating two readings. It means if the attacker is able to determine the tag she reads is indeed the one of the tag she has ever read before, we say the attacker is success. The attack works as follows.

- 1) The attacker randomly generates a nonce  $n_1$  and activates the protocol by sending it to  $\mathcal{T}$ .
- 2)  $\mathcal{T}$  replies  $U$  according to the protocol.
- 3) The attacker terminates the communication.

Notice that the protocol is terminated in the middle, so the tag will not update its key. The attacker repeats the above steps and reads the tag again sometimes later. She will obtain another message  $U'$  in Step 3.

Now, the attacker can associate these two readings together. Since each of the  $k_i[0], k_i[1], \dots, k_i[d-1]$  are only 4-bits long as described in the ACTION protocol, the attacker can enumerate 16 possible values and compute each of the hash  $H(n_1 || n_2 || k_i[0]), H(n_1 || n_2 || k_i[1]), \dots, H(n_1 || n_2 || k_i[d-1])$ . The values of  $k_i[0], k_i[1], \dots, k_i[d-1]$  can therefore be obtained from the message  $U$ , and the values of  $k'_i[0], k'_i[1], \dots, k'_i[d-1]$  can be obtained from the message  $U'$ . If there is no legitimate reading between these two attacker's readings, the value of  $k_i$  remains the same. It means the attacker can perform an active tracking attack without difficulty.

An intuitive though to fix the scheme is to increase  $l_k$  (the length of each  $k_i[\ ]$ ) to a larger number, say 20, which requires much more computation effort to figure out the value  $k_i[\ ]$ . However, it will also increase the branching factor as well. Moreover, it significantly increases the average number of search. Figure 3 shows the average number of search versus  $l_k$ . If  $l_k$  is increased to 30, where the value can still be inverted by brute force, the average numbers of search required by the reader for  $N$  is 0.1, 1, or 10 millions tags are the same as the trivial solution described in [2].<sup>2</sup> It implies the protocol fails to prevent an active tracking attack.

### 3.2 Desynchronizing Attack

In desynchronizing attack, an attacker intends to interrupt a legitimate reader from reading a tag. We say an attacker succeed if the key stored in the tag is different from the key stored in the reader.

<sup>2</sup>A trivial solution utilizes the message  $\mathcal{H}(n_1 || n_2 || s_i)$  as the identifier.

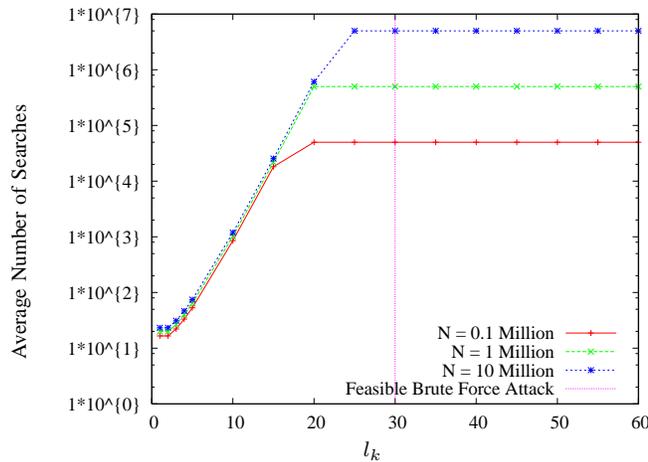


Fig. 3: Search numbers versus the length of  $k_i$  [ ] and  $l_k$ .

An attacker hides herself near a victim tag. When a legitimate reader reads a tag, the attacker remains silent until the reader sends the third message  $\sigma$ . The attacker jams the radio such that the tag can not receive, or receives with the wrong  $\sigma$ . Since  $\sigma$  appears to be invalid, the tag will refuse to update its key. However, the reader has already updated its key and removed the past key from its key tree. Therefore, the keys are desynchronized and the reader is unable to read the tag anymore.

## 4. Solutions

We had figured out two weaknesses of the ACTION protocol to show that it can not resist to an active tracking attack and a desynchronizing attack. In the section, we tries to present the improved ideas.

In ACTION, the authors claimed that in order to maintain best search complexity, the branching factor  $\delta$  is suggested to set as 16. The parameter  $d$  is hence set to 4 bits only. Therefore, the indices of each node are just the value from 1 to 16. In other words, an attacker can enumerate all 16 possible vaules to compare with the correct hash value in  $U$ . If  $d$  is set as 5,  $\delta$  will become 32 which significantly increases the search numbers of the reader. This fatal weakness happens due to the formula of  $\delta = 2^d$  in [1]. Our solution comes: increase the size of  $d$  but keep  $\delta$  fixed . For example, we set the values  $\delta$  and  $d$  as 16 and 60, respectively. The solution not only increases the length of  $d$  to prevent an active tracking attack, but maintains the same search complexity. Since the branching factor is still set to a small number, say 16, there are still only 16 possible indices of a node. More importantly, the size of each index is 60 bits. While the attacker is difficult to try all  $2^{60}$  possible values to compare with the eavesdropped hash value, the reader, instead, actually knows these 16 indices values which are with length of 60 bits. Consequently, when identifies the

messages  $U$ , the reader will obtain the correct value of each node with at most 16 times test.

The desynchronization problem of ACTION is because of it lacks of a key-updated report mechanism.  $\mathcal{R}$  could not assure whether  $\mathcal{T}$  has updated the key or not. Our solution comes:  $\mathcal{R}$  keeps the last copy of the path key and secret key until  $\mathcal{T}$  replies a message to indicate the key update procedure is finished. For example, after updated the key,  $\mathcal{T}$  replies  $\alpha = \mathcal{H}(n_1 || n_2 || k'_i || k_i)$  to  $\mathcal{R}$ .  $\mathcal{R}$  will verify it's correctness. If it is correct,  $\mathcal{R}$  will store the updated key. Otherwise,  $\mathcal{R}$  will re-read  $\mathcal{T}$ .

Whenever the report message is received by  $\mathcal{R}$ , we can assert that the keys stored in  $\mathcal{T}$  had been updated. If the message does not arrive at  $\mathcal{R}$ , it can be either the case: 1)  $\mathcal{T}$  did not update its key and terminated the protocol for some reason, or 2)  $\mathcal{T}$  updated its key to  $(k'_i, s'_i)$  (the new key pair that also generated at  $\mathcal{R}$ ). In any case,  $\mathcal{R}$  will re-read  $\mathcal{T}$  until the key-updated report message is received. Meanwhile, all key pairs generated in previous trials will also be trashed to avoid phantom tags (records that does not exist).

## 5. Conclusion

This paper has shown that the ACTION protocol is vulnerable to a tracking attack and a desynchronizing attack. To improve it, the basic idea is to unconnect the relation between the branching factor and the path key. As described in the above section, the improved solution not only maintains the same search complexity, but significantly improves the security strength of each part of path key. Meanwhile, in order to fix the protocol to protect against desynchronizing attacks, the reader keeps the last copy of the path key and secret key for the tag, which similars to the SASI [3]. Another method is to read the tag again after updated the key. If the tag responses with the message created by the past key, it indicates the protocol has not yet been completed.

We are now in progressing of another work that facilitates the above two improved ideas in detail. We will simulate the newly design protocol to show its performance and analysis it's secrecy with a formal method. In addition, we also calculate the best parameter set to make the protocol more efficient.

## References

- [1] L. Lu, J. Han, R. Xiao, and Y. Liu, "ACTION: Breaking the privacy barrier for RFID systems," in *INFOCOM 2009, IEEE*, 2009, pp. 1953–1961.
- [2] S. Weis, S. Sarma, R. Rivest, and D. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Proceeding of Security in pervasive computing: First International Conference*. Springer, 2004, pp. 201–212.
- [3] H.-Y. Chien, "SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity," *Dependable and Secure Computing, IEEE Transactions on*, vol. 4, no. 4, pp. 337–340, 2007.

- [4] T. Dimitriou, "A secure and efficient RFID protocol that could make big brother (partially) obsolete," in *Proceeding of Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference*, 2006, pp. 269–275.
- [5] D. Molnar and D. Wagner, "Privacy and security in library RFID: Issues, practices, and architectures," in *Proceedings of the 11th ACM conference on Computer and communications security*. ACM New York, NY, USA, 2004, pp. 210–219.
- [6] D. Molnar, A. Soppera, and D. Wagner, "A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags," *Selected Areas in Cryptography*, vol. 3897, pp. 276–290, 2006.
- [7] L. Lu, J. Han, L. Hu, Y. Liu, and L. Ni, "Dynamic key-updating: Privacy-preserving authentication for RFID systems," in *Proceeding of Pervasive Computing and Communications, 2007. PerCom'07. Fifth Annual IEEE International Conference*, 2007, pp. 19–23.
- [8] H.-M. Sun and W.-C. Ting, "A Gen2-based RFID authentication protocol for security and privacy," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 8, pp. 1052–1062, Aug 2009.
- [9] H.-M. Sun, W.-C. Ting, and K.-H. Wang, "On the security of Chien's ultra-lightweight RFID authentication protocol efficient time-bound hierarchical key management," *Dependable and Secure Computing, IEEE Transactions on*, no. 99, 2009.

# Reversible data hiding scheme using improved hiding tree

Jang-Hee Choi<sup>1</sup> and Kee-Young Yoo<sup>2</sup>

<sup>1</sup>School of Electrical Engineering and Computer Science, Kyungpook National Univ., Daegu, South Korea

<sup>2</sup>School of Computer Science and Engineering, Kyungpook National Univ., Daegu, South Korea

(Corresponding author : Kee-Young Yoo)

**Abstract**—*The difference between other data hiding schemes and Wu et al.'s scheme proposed in 2009 is that the histogram shifting and difference expansion techniques are not used. The high capacity is achieved by new technique, hiding tree. However Wu et al.s did not consider quality of the stego image. Compression codes is created instead of stego image in Wu et al.'s scheme. It can be not a steganography scheme, because a steganography scheme has an cover image as an input and stego image as a output.*

*In this scheme, the hiding tree is improved. Although the nodes of hiding tree is fixed, the level of the tree is changed and determined to control the changed pixels by the threshold value according to each block. The block size affects the quality of the stego image although the amount of the additional information is changed. According to improved hiding tree and these few mechanisms, the quality of the stego image can be improved until perceptible distortion is disappeared.*

**Keywords:** Reversible, Data hiding, Steganography, Hiding tree

## 1. Introduction

Recently, an enormous amount of information is transmitted over the Internet. Because this communication channel is open to anyone, any unauthorized person can intercept the transmitted information. Some sensitive information need to be kept from unauthorized use. There are two ways to provide confidentiality of the sensitive information: cryptography and data hiding.

Cryptography is a popular technique to keep the confidentiality. The meaningful plain text is changed into meaningless cipher text in cryptography. And the data hiding also provide confidentiality by concealing the secret data.

Communication of cipher text over the Internet can be known to an attacker. If an attacker can decrypt the cipher text, the confidentiality is broken. In contrast with cryptography, it is very difficult that an attacker becomes aware of whether there is secret data or not in the communicated information when secret data is concealed.

In data hiding, digital contents such as video, audio or digital image are used to conceal the secret data. The data hiding schemes [1-14] are called 'reversible' when the cover image can be recovered after extracting the secret data from

stego image. Otherwise, the data hiding schemes are called 'irreversible'.

The first reversible schemes was proposed by Hongsinger et al. in 2001 [7]. This scheme is different with previous data hiding schemes. The previous data hiding schemes were not reversible. The other words, the cover image could not be recovered. Although Hongsinger et al.'s scheme could be used in sensitive area such as military image or medical image, there is salt and peppers noise problem in stego image.

Ni et al. proposed data hiding scheme based on histogram shifting in 2006 [14]. A peak point which is used to embed the secret data and zero point which is not presented pixel value in the cover image is searched in histogram. And other values are shifted to zero point by one. Since Ni et al. proposed the scheme, many reversible data hiding schemes [6][9]based on histogram shifting are proposed to improve the quality of the stego image and increase the capacity.

The difference expansion (DE) technique [13] that can provide adjustable embedding capacity depending on a predetermined threshold is also base of many data hiding schemes [2][4][5][7][12].

Many data hiding schemes are based on histogram or DE technique. In contrast with these schemes, Wu et al. proposed new data hiding scheme[1]. They used hiding tree to embed secret data instead of histogram and DE technique.

Wu et al. did not consider quality of the stego image. The difference between cover image and stego image is larger than other data hiding schemes. To make up for the fault, the compressed codes are created instead of a stego image. If an attacker know that the codes are compressed and can decompress the codes, he or she can easily notify that the cover image modified.

In this paper, the proposed scheme is based on improved hiding tree. The difference between the values of the cover image and the values of the stego image is decreased by using improved hiding tree when the secret data is embedded into the cover image. The threshold value and block size are used to control the capacity and quality of the stego image. It is more adaptive method than Wu et al.'s scheme to embed the secret data into the cover image. Therefore, there is no problem to generate stego image.

The organization of this paper is as follows. In section 2, related works about a prediction technique and Wu et al.'s scheme are introduced. section 3 explains the proposed

scheme more detail. The experimental results of the proposed scheme are described in section 4. Finally section 5 contains conclusion of this study.

## 2. Related works

In this section, The binary tree is reviewed to use in the data hiding. And after the edge prediction technique is introduced, Wu et al.'s scheme is briefly described.

The edge prediction technique is discussed to calculate prediction error value. This prediction technique has better performance than other prediction technique.

### 2.1 Review of binary tree

The binary tree is one of the data structures. Two nodes, child nodes, are connected to one node, parent node, in binary tree. This relationship can be used in reversible data hiding. A values is changed to two other values by secret data in data hiding. It seems like that the parent node is changed to one node of child nodes.

Therefore the binary tree can be used to embed and extraction the secret data in data hiding. When secret data is embedded into a cover image, the value of the cover image pixel can be regarded as a parent node in the binary tree. If the secret data is '0', the pixel value is changed to left child node in the binary tree. Otherwise, the pixel value is changed to right child node in the binary tree. When secret data is extracted from stego image, the pixel value of the stego image can be regarded as a child node of the binary tree. If the pixel value is left child node in the binary tree, the pixel value is '0' and the pixel value of the stego image is changed to parent node in the binary tree as the pixel value of the cover image.

### 2.2 The edge prediction technique

The edge prediction technique has another name, JPEG-LS [15]. This technique was used at lossless compression in JPEG. The prediction values are calculated by the this prediction technique. By finding the difference between the pixel values and the prediction values, the prediction error values are generated, called prediction coding. And the prediction error values are compressed by entropy coding.

The prediction values  $x'$  are calculated by neighboring pixels shown in Fig. 1.  $x$  is the current pixel to be predictive. And  $a, b$  and  $c$  are neighboring pixels of  $x$ . This edge prediction technique detects the vertical or horizontal edge[1]. When there is a horizontal edge such that difference between  $a$  and  $c$  is bigger than difference between  $b$  and  $c$ ,  $x' = a$ . When there is vertical edge such that difference between  $b$  and  $c$  is bigger than difference between  $a$  and  $c$ ,  $x' = b$ . Otherwise,  $x' = a + b - c$ .

The calculation of  $x'$  is represented as the following equation (1).

$c$	$b$
$a$	$x$

Fig. 1: Prediction template

$$x' = \begin{cases} \min(a, b) & \text{if } c \geq \max(a, b) \\ \max(a, b) & \text{if } c < \min(a, b) \\ a + b - c & \text{otherwise} \end{cases}, \quad (1)$$

where  $\min( )$  and  $\max( )$  are functions to obtain minimum and maximum values of the parameters, respectively.

The prediction error values ( $ev_{ij}$ ) are generated by the following equation (2).

$$ev_{ij} = x - x'. \quad (2)$$

Prediction decoding is to recover the original pixel value. It is archived by adding the prediction values and the prediction error.

### 2.3 Review of Wu et al.'s scheme

Wu et al. proposed a steganography scheme based on hiding tree and prediction coding for high capacity.

The embedding procedure of Wu et al.'s scheme is described below

Algorithm : Embedding

Input: Cover image ( $I$ ), secret data.

Output: Compressed bit string

Step 1: Generate the encrypted secret data ( $S$ ).

Step 2: Calculate prediction values of  $I$  using equation (1).

Step 3: Calculate prediction error values using equation (2).

Step 4: Building hiding tree. The nodes of hiding tree consist of absolute values of prediction errors. The values of each node of hiding tree are selected by frequencies of absolute values of prediction errors. More frequent values are selected to determine low level nodes except of first node. The first node consists of zero in all cases. Finally, the absolute values of prediction errors which do not occur are connected to the last node. The values connected last nodes of the hiding tree are not fixed. So, any values which is not occur can be connected to the last nodes.

Step 5: Modify prediction errors. prediction error ( $e_{ij}$ ) is modified by following equation (3).

$$ne_{ij} = \begin{cases} ae_{left} & \text{if } e_{ij} \geq 0 \text{ and } S = 0 \\ ae_{left} \times (-1) & \text{if } e_{ij} < 0 \text{ and } S = 0 \\ ae_{right} & \text{if } e_{ij} \geq 0 \text{ and } S = 1 \\ ae_{right} \times (-1) & \text{if } e_{ij} < 0 \text{ and } S = 1 \end{cases}, \quad (3)$$

where  $ae_{left}$  and  $ae_{right}$  mean the absolute value of prediction error of the left and right child node, respectively, and  $ne_{ij}$  is modified prediction error value. If  $ae_{ij}$  are last node of the hiding tree, the secret data is not embedded. The prediction errors are just changed by following equation (4).

$$ne_{ij} = \begin{cases} ce \times (-1) & \text{if } ev_{ij} < 0 \\ ce & \text{otherwise} \end{cases}, \quad (4)$$

where  $ce$  is a node which are connected to last node of the hiding tree.

Step 6: Compress modified prediction error values. Entropy coding is used to compress modified prediction error values without losing any data. Finally, the compressed bit string is generated

Next, extraction procedure is described below.

Algorithm : Extraction

Input : The compressed bit string, hiding tree

Output : Cover image  $I$ , secret data.

Step 1: Decompress the compressed bit string. The modified prediction errors can be obtained after this step.

Step 2: Extract encrypted secret data. If the value of the modified prediction error is left child node of parents node in the hiding tree, the encrypted secret data is '0', otherwise '1' except of connected nodes.

Step 3: Recover prediction error value  $e_{ij}$  using following equation (5).

$$e_{ij} = \begin{cases} ae_{parent} & \text{if } ne_{ij} \geq 0 \\ ae_{parent} \times (-1) & \text{if } ne_{ij} < 0 \end{cases}, \quad (5)$$

where  $ae_{parent}$  is the parent node of the absolute value of the modified prediction error.

Step 4: Decrypt encrypted secret data.

Step 5: Recover  $I$  using prediction decoding.

The large distortion of stego image is generated Step 5 and the hiding tree in embedding procedure of Wu et al.'s scheme. The difference between the parent nodes and child nodes in hiding tree can be significant because the nodes of the hiding tree is constructed by the rule that the frequent absolute values of the prediction error are the value of the low level nodes of hiding tree. The character of hiding tree affects the quality of stego image. If the difference of the child nodes and parent nodes is big, the distortion of the stego image is significant. Because of this problem, the stego image can not be generated.

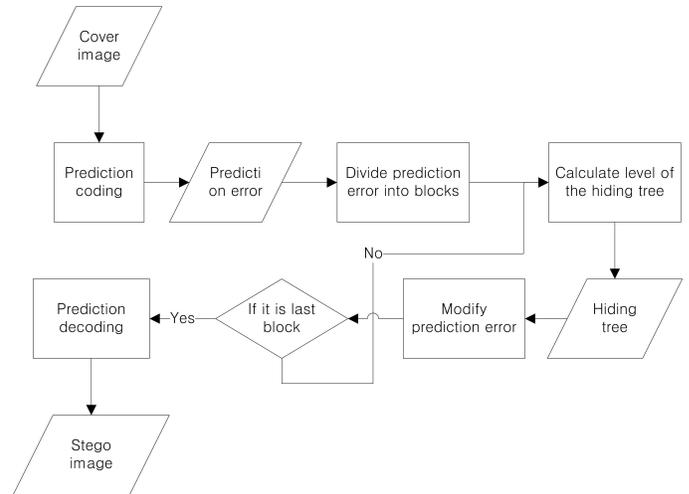


Fig. 2: Embedding procedure

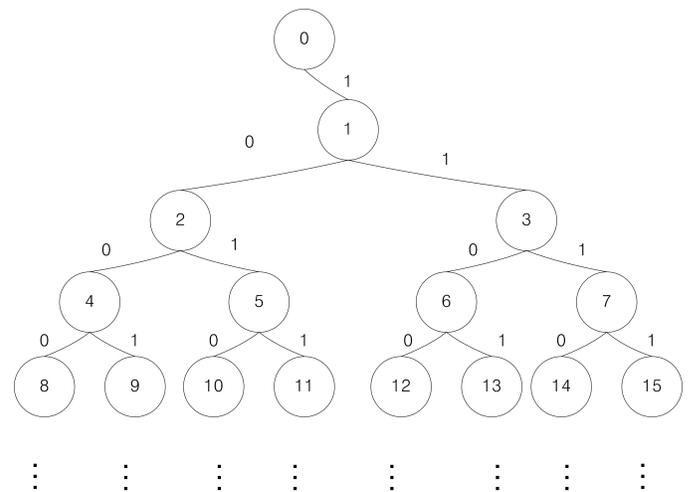


Fig. 3: The fixed hiding tree

### 3. The proposed scheme

In this section, a new data hiding scheme is proposed. Wu et al.'s scheme is not complete steganography, because stego image was not created. But stego image is created in the proposed scheme. Fig 2 shows the embedding procedure.

#### 3.1 Embedding procedure

The proposed hiding tree shown in Fig. 2 is fixed. But according to distribution of the prediction error values in the blocks, the level of the hiding tree of each block is changed. As a results, distortion of the stego image is decreased. A threshold is used to control the capacity and quality of the stego image.

Algorithm : Embedding

Input : Cover image  $I$ , secret data  $S$ , threshold  $T$  and block size  $M$

Output : Stego image  $I'$ , level ( $L$ ) of the hiding tree of each block,  $T$  and block size.

Step 1: Calculate prediction error. To calculate prediction error, the edge prediction technique mentioned in section 2.1 is used except of first row and column. First row and column's prediction value  $p_{ij}$  are different from other pixels. In the first row, prediction value is the value of previous pixel ( $I_{i-1j}$ ) represented as

$$p_{ij} = I_{i-1j}. \quad (6)$$

And  $p_{ij}$  of first column is the value of above pixel ( $I_{ij-1}$ ) represented as

$$p_{ij} = I_{ij-1}. \quad (7)$$

The  $e_{ij}$  is calculated by equation (2).

Step 2: Divide prediction errors into blocks sized  $M \times M$ .

Step 3: Calculate level of the hiding tree of current block. The each node of the hiding tree is fixed. However, level of block is calculated by a predetermined  $T$  and the distribution of prediction errors in the block. The frequencies of absolute prediction errors in the block are firstly scanned. And then the largest value of absolute prediction errors ( $le$ ) but less than  $2^T$  is chosen.  $L$  is obtained by the following equation (8).

$$L = \lfloor \log_2 le + 1 \rfloor + 1, \quad (8)$$

where  $\lfloor \cdot \rfloor$  is floor function. If there is no largest value of absolute prediction errors,  $L = -1$ .

Step 4: Modify prediction errors using the hiding tree that the level is  $L$  and secret data  $S$ . Prediction errors need to be classified into two groups. First group is embedable group which satisfy the following condition (9)

$$e_{ij} < 2^{L-1}. \quad (9)$$

Another group is not embedable which do not satisfy the condition (9). If prediction error is embedable group, prediction error is modified by the equation (3). Otherwise, prediction error is modified by the following equation (10).

$$ne_{ij} = \begin{cases} e_{ij} - 2^{L-1} & \text{if } e_{ij} < 0 \\ e_{ij} + 2^{L-1} & \text{otherwise} \end{cases} \quad (10)$$

Step 5: If it is not last block, go step3. Otherwise, go Step 6.

Step 6: Calculate stego image using the following equation (11).

$$I'_{ij} = p_{ij} + ne_{ij}. \quad (11)$$

If underflow and overflow occur, location is marked on location map instead of embedding the secret data into the pixels.

Table 1: The experimental results of Wu et al.'s scheme

Images	Underflow	Overflow	Capacity	PSNR
lena	105	68	0.998	26.56
peppers	2956	116	0.998	19.21
goldhill	478	1012	0.998	19.90
baboon	4891	126	0.980	17.04
boat	331	363	0.998	24.35
zelda	277	126	0.999	26.70
jet	123	115	0.997	27.29
zelda	277	126	0.999	26.70

In embedding procedure of the proposed scheme, additive information is generated such as location map for underflow and overflow, the levels of each block, threshold value and block size. These additional information is used to extract the secret data and recover the cover image during extraction.

### 3.2 Extraction procedure

In extraction procedure, the secret data can be found in stego image. And also cover image is found.

Algorithm : Extraction

Input : Stego image  $I'$ , level ( $L$ ) of the hiding tree of each block, and block size( $M$ ).

Output : Cover image and secret data.

Step 1: Calculate a modified prediction error using edge prediction technique.

Step 2: Extract secret data. If prediction error is less than  $2^L$ , then secret data is can be extracted by using hiding tree which constructed by  $L$ . When absolute prediction error is left or right child node of the parent node in hiding tree,  $S$  is '0' or '1', respectively.

Step 3: Recover prediction error. modified prediction errors are classified into two groups. First group satisfy the following condition (12).

$$ne_{ij} < 2^L. \quad (12)$$

Another group doesn't satisfy the condition (12). If modified prediction error is first group, prediction error is changed to the value of parent node in hiding tree. Otherwise prediction error is calculated by the following equation (13).

$$e_{ij} = \begin{cases} ne_{ij} + 2^{L-1} & \text{if } ne_{ij} < 0 \\ ne_{ij} - 2^{L-1} & \text{otherwise} \end{cases} \quad (13)$$

Step 4: Repeat the Step 1-3 until the last pixel.

And then cover image is obtained by prediction decoding.

## 4. The experimental results

In this section, eight gray-scale images sized  $512 \times 512$  are used to estimate performance of the proposed scheme as shown in Fig. 4.



Fig. 4: Eight  $512 \times 512$  gray-scale images for the experiment: (a) Baboon, (b) Boat, (c) Goldhill, (d) Jet (e) Lena (f) Peppers (g) Toy (h) Zelda

Table 2: The experimental results of the proposed scheme for test image "Baboon" under various block size when threshold value is '1'

Block size	Underflow	Overflow	Capacity	PSNR
2	22	0	0.108	48.50
3	0	0	0.106	45.93
4	28	0	0.108	44.50
6	0	0	0.106	43.19
8	29	0	0.108	42.64
10	0	0	0.106	42.48
16	28	0	0.108	42.36

Table 3: The experimental results of the proposed scheme for test image "Lena" under various threshold when block size is '2'

$T$	Underflow	Overflow	Payload	PSNR
0	0	0	0.111	48.21
1	0	0	0.296	42.84
2	0	0	0.586	38.20
3	0	0	0.818	34.92
4	0	2	0.889	33.18
5	0	4	0.923	32.13
6	3	6	0.936	31.46

The capacity and quality of the stego image are estimated. The capacity can be expressed by bpp(bits/pixels). And quality of the stego image can be expressed Peak-Signal-to-Noise Ratio(PSNR). PSNR is defined as following equation (14).

$$PSNR = 10 \log_{10} \frac{MAX^2}{MSE}, \quad (14)$$

where  $MAX$  is the maximum possible pixel value of the image, and  $MSE$ (Mean Squared Error) which for two monochrome images ,cover image and stego image sized  $m \times n$ , is defined as

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - S(i, j)]^2, \quad (15)$$

where  $I$  is cover image and  $S$  is stego image.

In Wu et al.'s scheme, there is a compressed bit string instead of stego image. Therefore PSNR can not be calculated. But suppose that an attacker can decompress the bits string and recover the image. He or she can find the fact that the image is modified because the recovered image's distortion

Table 4: The experimental results of the proposed scheme when block size is '2' and threshold value is '1'

Images	Underflow	Overflow	Capacity	PSNR
lena	0	0	0.296	45.48
peppers	138	0	0.223	46.13
goldhill	0	0	0.399	48.89
baboon	22	0	0.108	48.50
boat	4	0	0.183	46.61
zelda	10	0	0.291	45.19
jet	0	0	0.406	45.14
zelda	10	0	0.291	45.19

is large as shown in Table 1. Eventually, the secret bits can be extracted by attacker.

The experimental results of the proposed scheme according to the threshold  $T$  are shown in Table 2. When the threshold  $T$  is small, the capacity is very low. But image quality is high. Thus the capacity and distortion of stego image can be controlled by threshold  $T$ .

And the block size also control the quality of the stego image with the same capacity. When the block size is larger than 8, the quality of the stego image is similar. However, the distortion is decrease when the block size is small. But the amount of information about the levels of the tree of each block is increased. The experimental results of various block size are shown in Table 3.

When the threshold value is '1' and block size is '2', the experimental results for various image is shown Table 4. According to the characteristic of image, the underflow or overflow is can occur. When the smooth image is used as a cover image. the capacity is high than a image which has many edge areas. However, the variance of pixels between cover image and stego image in smooth image is larger than edge image.

Thus, the capacity and distortion of the stego image can be controlled by various threshold values and block sizes. Wu et al.'s scheme is not suitable to embed secret bits safely because the distortion of the stego image is significant. The proposed scheme improves the binary tree and embedding method to decrease the distortion of stego image.

## 5. conclusion

In this paper, the new data hiding scheme is proposed. The proposed scheme improve Wu et al.'s scheme. In Wu et al.'s scheme, the distortion of the stego image was significant by hiding tree. As the experimental results in section 4, the PSNR of Wu et al.'s scheme is less than 30dB. The PSNR of the proposed scheme is more than 45dB when the threshold value is 1 and block size 2. The capacity of Wu et al.'s scheme is almost 1 when the capacity is measured by bpp. The proposed scheme has low capacity. However, capacity is not more important than the quality of the stego image in the data hiding. Therefore, the proposed scheme has better performance than Wu et al.'s scheme.

## Acknowledgments

This Research was supported by Kyungpook National University Research Fund, 2009. And this work is supported by the 2nd Brain Korea 21 Project in 2009.

## References

[1] Hsien-Chu Wu, Hao-Cheng Wang, Chwei-Shyong Tsai, Chung-Ming Wang, "Reversible image steganographic scheme via predictive coding", *Displays*, Vol. 31, pp. 35-43, 2010.

[2] Hsien-Chu Wua, Chih-Chiang Lee, Chwei-Shyong Tsai, Yen-Ping Chu, Hung-Ruei Chen "A high capacity reversible data hiding scheme with edge prediction and difference expansion," *The Journal of Systems and Software* 82, pp.1966-1973, 2009.

[3] Yongjian Hu, Heung-Kyu Lee, and Jianwei Li, "DE-Based reversible data hiding with improved overflow location map," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, No. 2, 2009.

[4] Chin-Feng Lee., Hsing-Ling Chen, Hao-Kuan Tso, "Embedding capacity raising in reversible data hiding based on prediction of difference expansion," *The Journal of Systems and Software* 83, pp.1864-1872, 2010.

[5] Hsien-Wen Tseng , Chi-Pin Hsieh, "Prediction-based reversible data hiding," *Information Sciences* 179, pp.2460-2469, 2009

[6] Wien Hong, Tung-Shou Chen, Chih-Wei Shiu, "Reversible data hiding for high quality images using modification of prediction errors," *The journal of Systems and Software* 82, pp.1833-1842, 2009

[7] Ming Chen, Zhenyong Chen, Xiao Zeng, Zhang Xiong, "Reversible Data Hiding Using Additive Prediction-Error Expansion," *ACM Multimedia Security'09*, pp.19-24, 2009

[8] C.W. Honsinger, P. Jones, M. Rabbani, J.C. Stoffel, "Lossless recovery of an original image containing embedded data," *US Patent* 6 278 791, 2001

[9] Piyu Tsai, Yu-ChenHub, Hsiu-LienYeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing* 89, pp.1129-1143, 2009

[10] Kyung-Su Kima, Min-JeongLee, Hae-YeounLee, Heung-KyuLee, "Reversible data hiding exploiting spatial correlation between sub-sampled images," *Pattern Recognition* 42, pp.3083-3096, 2009

[11] Der-Chyuan Lou, Nan-I Wu, Chung-Ming Wang, Zong-Han Lin, Chwei-Shyong Tsai, "A novel adaptive steganography based on local complexity and human vision sensitivity," *The Journal of Systems and Software* 83, pp.1236-1248, 2010

[12] Xianting Zenga, "Lossless Data Hiding Scheme Using Adjacent Pixel Difference Based on Scan Path," *Journal of Multimedia*, Vol. 4, No. 3, 2009

[13] Jun Tian, "Reversible data embedding usgin a difference expansion," *IEEE Transactions on Circuits and System for Video Technology*, Vol. 13, Issue. 8, pp.890-896, 2003

[14] Zhicheng Ni, Yun-Qing Shi, Nirwan Ansari, Wei Su, "Reversible data hiding," *IEEE Transactions on Circuits and System for Video Technology*, Vol. 16, Num. 3, pp.354-362, 2006

[15] "Lossless and near-lossless coding of continuous tone still images(JPEG-LS), ISO/IEC JTC1/SC29 WG1 FCD 14495," *International Standards Organizations/ International Electrotechnical Commission*, 1997 [Online]. Available: <http://www.jpeg.org/public/fcd14495p.pdf>

# A reversible image hiding scheme using novel linear prediction coding and histogram shifting

Dae-Soo Kim<sup>1</sup>, Gil-Je Lee<sup>2</sup>, and Kee-Young Yoo<sup>3</sup>

<sup>1</sup>Department of Information Security, Kyungpook National Univ., Daegu, South Korea

<sup>1</sup>Graduate School of Electrical Engineering and Computer Science, Kyungpook National Univ., Daegu, South Korea

<sup>3</sup>School of Computer Science and Engineering, Kyungpook National Univ., Daegu, South Korea

(Corresponding author : Kee-Young Yoo)

**Abstract**—In 2009, Tsai et al. proposed reversible image hiding scheme using linear prediction coding and histogram shifting. Tsai et al.'s scheme improved the hiding capacity of Ni et al.'s scheme using the linear prediction coding and two histograms. In the linear prediction coding, however, the basic pixel is not used. If a value of basic pixel is the largest or the smallest in a block, only one histogram is generated and the hiding capacity is decreased. To solve the problems, this paper proposes the novel linear prediction coding with the inverse S-order and one histogram using two peak points. In experimental results, the hiding capacity of the proposed scheme is superior to Tsai et al.'s scheme.

**Keywords:** Reversible data hiding, Steganography, Histogram shifting, Inverse S-order

## 1. Introduction

Data hiding is a technique that embeds data into digital media to convey secret data by slightly altering the contents of the media, so that the embedded data is imperceptible [1], [2]. In image data hiding, during data embedding, distortion of image occurs since the pixel values in the cover image will be inevitably changed. The sensitive images, such as military images, medical images, or artwork preservation, are intolerable about the embedding distortion. For medical images, even slight changes are caused of the potential risk of the misdiagnosis.

Nowadays, reversible data hiding research has become quite important. Reversible data hiding techniques are designed to solve the problem of sensitive images. After the embedded secret data is extracted, the image can be completely restored to its original state before embedding occurred. Several reversible data hiding schemes have been proposed [4], [5], [6], [7]. In 2006, Ni et al. proposed a very different reversible data hiding technique based on the histogram shifting technique [8]. Ni et al.'s scheme adjusts pixel values between peak point and zero point to conceal data and to achieve reversibility. However, the capacity is limited by the most frequent pixel values in the cover image. After proposed Ni et al.'s scheme, to improve the reversible

data hiding based histogram shifting had researched [3], [9], [10], [11], [12], [13], [14], [15]. In 2009, Tai et al. proposed a pixel difference based reversible data hiding scheme [3]. Tsai et al. proposed a block-based reversible data hiding scheme using prediction coding [11]. However, Tsai et al.'s scheme had problems in prediction coding and dividing histogram into two sets. This problems cause a decrease of the hiding capacity. In this paper, the novel linear prediction coding with the inverse S-order and generating one histogram using two peak points are proposed to solve the problems of Tsai et al.'s scheme.

The rest of this paper is organized as follows: In Section 2, the reversible data hiding schemes proposed by Tai et al. and Tsai et al. are introduced. In Section 3, the embedding, extraction and recovery procedures of the proposed method are presented. Experimental results are given in Section 4, and ending with conclusions in Section 5.

## 2. Related works

### 2.1 The pixel differences in an inverse S-order

Tai et al. is proposed to a similar method in which the differences of two consecutive pixels are calculated. Data embedding is done by modifying the histogram of the absolute value of the differences using a proposed binary tree [3].

In Tai et al.'s scheme, scans the pixel value  $c_i$  of cover image  $C$  in an inverse S-order as shown in Fig. 1. The pixel difference  $e_i$  calculates between pixels  $c_{i-1}$  and  $c_i$  as follows

$$e_i = \begin{cases} c_i, & \text{if } i=0 \\ |c_{i-1} - c_i|, & \text{otherwise} \end{cases} \quad (1)$$

### 2.2 The histogram shifting

Ni et al. proposed a histogram-based reversible data hiding scheme [8]. In their scheme, all pixel values in the cover image are calculated to generate the image histogram for secret data embedding. The peak point and the zero

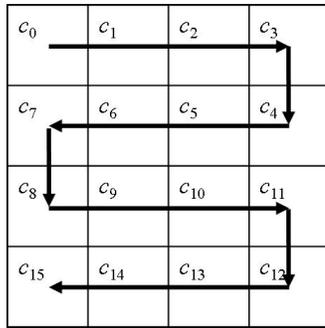


Fig. 1: Inverse S-order

point are selected as shown in Fig. 2. The peak point is the largest frequency in the histogram and the zero point is 0 or the least frequency in the histogram. The peak point is modified to embed the secret data by 1. Pixels with ranging from the peak point to zero point are modified, and pixels with outside range of values are no changed. The modified pixel values can be recovered when the embedded data is extracted.

### 2.3 Review of Tsai et al.'s scheme

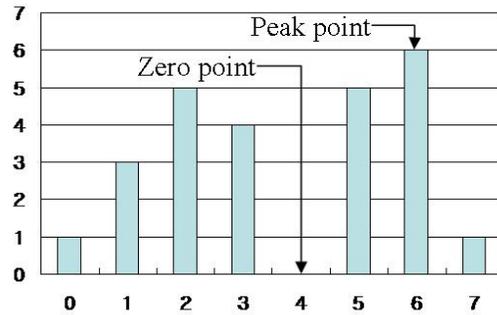
In 2009, Tsai et al.'s scheme is proposed to improve the hiding capacity of the Ni et al.'s scheme using the neighboring similarity of pixels in an image. The cover-image is divided into blocks of  $m \times m$  pixels and the prediction error value is calculated the prediction coding. It is performed using the basic pixel that is a center pixel of block shown in Fig. 3.

The prediction error values are modified based on histogram shifting technique to embed secret data. There are divided into two sets by basic pixel: non-negative histogram (NNH) and negative histogram (NH). Each set has its one peak point and zero point. Three cases are considered. If the prediction error value is peak point of NNH and NH, and secret bit is 1, no changed. Otherwise, the prediction error value is adjusted by 1 to a value closer to the zero point. Final, the prediction error value is between the peak point and zero point that is shifted by 1 closer to the zero point. In this case, no secret data is embedded in the prediction error value. After the secret data is embedded the stego-image is obtained to perform the reverse prediction coding using the modified prediction error value.

However, the basic pixel is not used the embedding procedure. If value of basic pixel is the largest or the smallest in a block, the one histogram is generated and one peak point is used. The capacity is limited by the frequency of peak point in the histogram [16]. For that reason, the capacity is decreased.

5	6	6	6	7
5	5	6	6	6
3	3	5	5	0
1	2	2	2	2
1	2	3	3	1

(a)



(b)

Fig. 2: Example of the peak point and zero point: (a) cover image and (b) image histogram

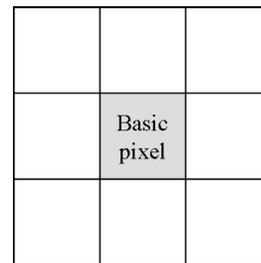


Fig. 3: The location of basic pixel in block with  $3 \times 3$  pixels

## 3. The proposed scheme

The proposed scheme is solved to the problems of Tsai et al.'s scheme [16]. The goal of the proposed scheme is provided to the higher hiding capacity than Tsai et al.'s scheme while keeping the good quality of the stego-image.

### 3.1 The main concepts

The proposed scheme is exploited to histogram shifting method and the novel linear prediction coding (NLPC) based on the inverse S-order. First, NLPC is solved to the problem of linear prediction coding of Tsai et al.'s scheme. The basic pixel is selected in the previous block and is excepted for the first block. Fig. 4 is shown that basic pixel  $r^{(n)}$  is selected to a center pixel of the nearest column or row of a previous block  $B^{(n-1)}$  by inverse S-order by cover-image, where  $n$  is sequence number of block  $B^{(n)}$ .

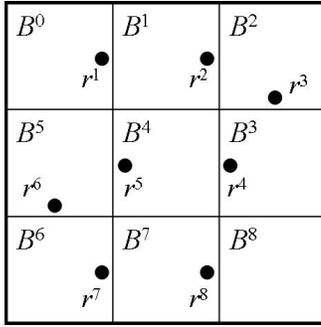


Fig. 4: The location of basic pixels

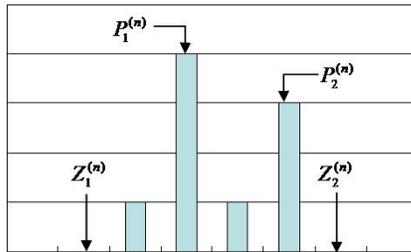


Fig. 5: The peak points and zero points

Second, the one histogram is generated. It has two peak points and two zero points. The peak points are the largest frequency and the second largest frequency are selected. Fig. 5 shows that two peak points and two zero points are searched. One of peak point  $P_1^{(n)}$  is the largest frequency of  $e_{(i,j)}^{(n)}$  where  $e_{(i,j)}^{(n)}$  is the prediction error values. Another peak point  $P_2^{(n)}$  is the second largest frequency of  $e_{(i,j)}^{(n)}$ . Two of zero point  $Z_1^{(n)}$  and  $Z_2^{(n)}$  are located beyond the range of both  $P_1^{(n)}$  and  $P_2^{(n)}$ .

### 3.2 The embedding procedure

**Input:**  $N \times N$  pixels cover-image  $C$ , secret data  $d_l$

**Output:**  $N \times N$  pixels stego-image  $S$ , Block size  $m$ , peak and zero points pair  $(P_1^{(n)}, Z_1^{(n)})$ ,  $(P_2^{(n)}, Z_2^{(n)})$  of each block

**Step 1:** Divide  $C$  into blocks of  $m \times m$  pixels.

**Step 2:** Calculate prediction error values (equation (2)) of each block that is difference between pixel  $c_{(i,j)}^{(n)}$  and basic pixel  $r^{(n)}$  while the block scans an inverse S-order.

$$e_{(i,j)}^{(n)} = c_{(i,j)}^{(n)} - r^{(n)}, \quad (2)$$

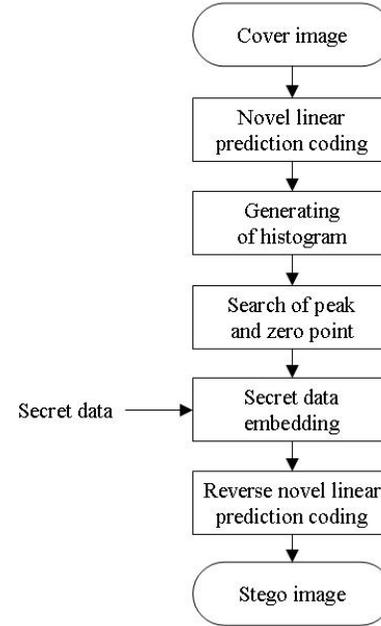


Fig. 6: Flowchart of embedding procedure

where  $n \geq 1$ ,  $0 \leq i, j \leq m - 1$

**Step 3:** Generate the histogram of the prediction error value  $e_{(i,j)}^{(n)}$  in each block  $B^{(n)}$ .

**Step 4:** Search two peak and zero points pair in the generated histogram.

**Step 5:** Embed secret data  $d_l$  in peak points by the following conditions.

If  $e_{(i,j)}^{(n)} = P_t^{(n)}$ , secret data are embedded by following equation (3)

$$e'_{(i,j)}^{(n)} = \begin{cases} e_{(i,j)}^{(n)} + d_l, & \text{if } P_t^{(n)} < Z_t^{(n)} \\ e_{(i,j)}^{(n)} - d_l, & \text{if } P_t^{(n)} > Z_t^{(n)} \end{cases}, \quad (3)$$

where  $e'_{(i,j)}^{(n)}$  is a modified prediction error value and  $t$  is 1 or 2.

If  $P_t^{(n)} < e_{(i,j)}^{(n)} < Z_t^{(n)}$  or  $P_t^{(n)} > e_{(i,j)}^{(n)} > Z_t^{(n)}$ ,  $e_{(i,j)}^{(n)}$  is modified by following equation (4)

$$e'_{(i,j)}^{(n)} = \begin{cases} e_{(i,j)}^{(n)} + 1, & \text{if } P_t^{(n)} < Z_t^{(n)} \\ e_{(i,j)}^{(n)} - 1, & \text{if } P_t^{(n)} > Z_t^{(n)} \end{cases}, \quad (4)$$

otherwise, no modification

**Step 6:** Perform the reverse prediction coding. The stego-image  $S$  is obtained by following equation (5)

$$s_{(i,j)}^{(n)} = e'_{(i,j)}^{(n)} + r^{(n)}, \quad (5)$$

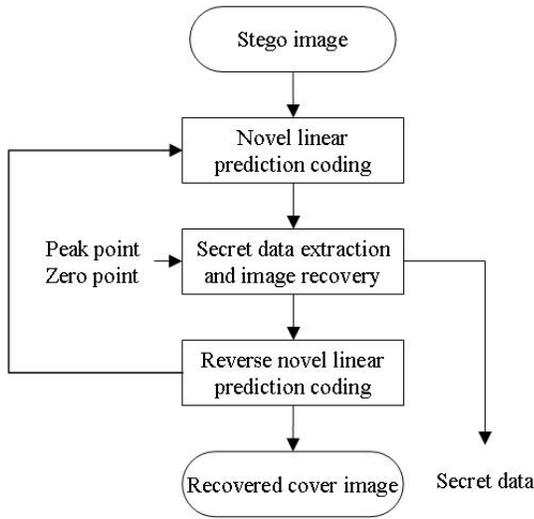


Fig. 7: Flowchart of extraction and recovery procedure

where  $n \geq 1, 0 \leq i, j \leq m - 1$

### 3.3 The extraction and recovery procedure

**Input:**  $N \times N$  pixels stego-image  $S$ , Block size  $m$ , peak and zero points pair  $(P_1^{(n)}, Z_1^{(n)})$ ,  $(P_2^{(n)}, Z_2^{(n)})$  of each block

**output:**  $N \times N$  pixels recovered cover-image  $C$ , secret data  $d_l$

**Step 1:** Divide  $S$  into blocks of  $m \times m$  pixels.

**Step 2:** Calculate prediction error values equation (6) of each block that is difference between pixels  $s_{(i,j)}^{(n)}$  and basic pixel  $r^{(n)}$  while the block scans an inverse S-order

$$e'_{(i,j)}^{(n)} = s_{(i,j)}^{(n)} - r^{(n)}, \quad (6)$$

where  $n \geq 1, 0 \leq i, j \leq m - 1$ . The basic pixel  $r^{(n)}$  is selected by the recovered pixel  $s_{(i,j)}^{(n)}$  in the block  $B^{(n)}$ .

**Step 3:** Extract the secret data (equation (7, 9)) and recover the prediction error values (equation (8, 10)).

If  $P_t^{(n)} \geq e'_{(i,j)}^{(n)} \geq Z_t^{(n)}$ ,

$$d_l = \begin{cases} 0, & \text{if } e'_{(i,j)}^{(n)} = P_t^{(n)} \\ 1, & \text{if } e'_{(i,j)}^{(n)} = P_t^{(n)} - 1 \end{cases} \quad (7)$$

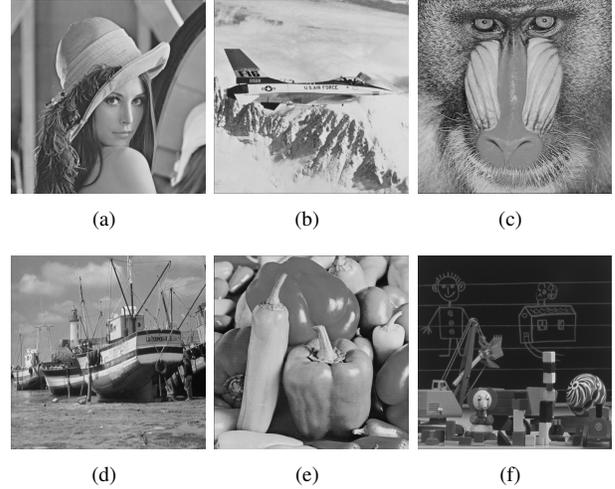


Fig. 8: Six  $512 \times 512$  size gray-scale images: (a) Lena, (b) Airplane, (c) Baboon, (d) Boat, (e) Pepper and (f) Toy

$$e_{(i,j)}^{(n)} = \begin{cases} e'_{(i,j)}^{(n)}, & \text{if } e'_{(i,j)}^{(n)} = P_t^{(n)} \\ e'_{(i,j)}^{(n)} + 1, & \text{otherwise.} \end{cases} \quad (8)$$

If  $P_t^{(n)} \leq e'_{(i,j)}^{(n)} \leq Z_t^{(n)}$ ,

$$d_l = \begin{cases} 0, & \text{if } e'_{(i,j)}^{(n)} = P_t^{(n)} \\ 1, & \text{if } e'_{(i,j)}^{(n)} = P_t^{(n)} + 1 \end{cases} \quad (9)$$

$$e_{(i,j)}^{(n)} = \begin{cases} e'_{(i,j)}^{(n)}, & \text{if } e'_{(i,j)}^{(n)} = P_t^{(n)} \\ e'_{(i,j)}^{(n)} - 1, & \text{otherwise.} \end{cases} \quad (10)$$

otherwise, no modification

**Step 4:** Recover the cover-image by following equation (11).

$$c_{(i,j)}^{(n)} = e_{(i,j)}^{(n)} - r^{(n)}, \quad (11)$$

where  $n \geq 1, 0 \leq i, j \leq m - 1$

**Step 5:** Repeat Step 2, Step 3, Step 4 until the final block.

The proposed scheme can be prevented to underflow and overflow by simple pre-processing work. In pre-processing stage, when the pixel value is 255, it is shifting to 254. Likewise, when the pixel values is 0, it is shifting to 1.

## 4. Experimental results

In this section, the proposed scheme is compared with Tsai et al.'s scheme in terms of capacity (bpp) and image

Table 1: The result of hiding capacity and image distortion

Test image	Tsai et al.'s scheme		The proposed scheme	
	Capacity (bit)	PSNR (dB)	Capacity (bit)	PSNR (dB)
Lena	93,044	54.14	114,824	53.17
Airplane	97,977	53.70	120,587	52.87
Baboon	82,995	56.31	100,655	55.49
Boat	70,470	55.14	82,371	54.11
Peppers	88,756	54.51	109,572	53.42
Toy	101,693	53.20	129,150	52.34
Average	89,156	54.55	109,527	53.57

Table 2: The result of hiding capacity (bit) and image distortion (PSNR) with several block size of Tsai et al.'s scheme

Test image	3×3		4×4		5×5	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
Lena	93,044	54.13	78,310	53.63	68,155	53.11
Airplane	97,977	53.70	83,175	53.25	73,058	52.80
Baboon	82,995	56.29	54,184	56.81	44,478	56.83
Boat	70,470	55.14	67,714	55.04	57,903	54.53
Peppers	88,756	54.50	74,155	53.97	64,500	53.31
Toy	101,693	53.20	90,495	52.50	82,522	52.10
Average	89,156	54.50	74,672	54.20	65,103	53.78

Table 3: The result of hiding capacity (bit) and image distortion (PSNR) with several block size of the proposed scheme

Test image	3×3		4×4		5×5	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
Lena	114,824	53.17	94,768	52.74	80,569	52.35
Airplane	120,587	52.87	100,723	52.47	87,267	52.11
Baboon	100,655	55.49	64,007	55.68	52,193	55.70
Boat	82,371	54.11	81,283	53.90	68,360	53.48
Peppers	109,572	53.42	89,824	52.97	76,461	52.45
Toy	129,150	52.34	111,572	51.78	99,605	51.47
Average	109,527	53.57	90,363	53.26	77,409	52.93

distortion (PSNR). The test images are the  $512 \times 512$  gray-scale image, as shown in Figure 8. The secret data used to generate the random number. When the size of block is  $3 \times 3$ , Table 1 shows the result of hiding capacity and image distortion between Tsai et al.'s and the proposed scheme. In table 1, the hiding capacity of Tsai et al.'s scheme and the proposed scheme were 93,044-bit and 114,824-bit in Lena image, respectively. The hiding capacity of proposed scheme was increased by 23%. The image distortion of proposed scheme is similar to Tsai et al.'s scheme. The image distortion is measured in PSNR [17], which is defined as follows

$$PSNR = \left( 10 \cdot \log_{10} \left( \frac{255^2}{MSE} \right) \right), \quad (12)$$

where MSE is the *meansquareerrorbetween* the original image and stego-image, and can be calculated by using the following equation (13).

$$MSE = \left( \frac{1}{MN} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I_{(i,j)} - I'_{(i,j)})^2 \right), \quad (13)$$

where  $I_{(i,j)}$  and  $I'_{(i,j)}$  indicate the pixel values of the original image and stego-image of size  $M \times N$ .

Table 2 and 3 show the result of hiding capacity and image distortion about several block sizes. It is noted that the larger block size, the lower neighboring similarity and the less the hiding capacity is even though the number of basic blocks is smaller [11]. The capacity of the  $3 \times 3$  block of Tsai et al.'s scheme and the  $4 \times 4$  block of the proposed scheme is similar. Although the block size is larger than Tsai et al.'s scheme, the hiding capacity is a similar result using the and modified histogram shifting method in proposed scheme.

## 5. Conclusion

In this paper, the problems of Tsai et al.'s scheme was solved by the novel linear prediction coding (NLPC) and modified histogram shifting technique. The NLPC that based on the invers S-order of pixel differences and modified histogram shifting technique that search two peak points in one histogram were proposed to provide the hiding capacity.

In experimental results, the hiding capacity of the proposed scheme is superior to Tsai et al.'s scheme and the image quality is similar. When the extraction and recovery procedure, two peak and zero point fair are required in each block. Although the block size of the proposed scheme is larger than that of Tsai et al.'s scheme, the hiding capacity is similar. It is shown that the communication data, peak and zero points are decreased.

## Acknowledgments

This Research was supported by Kyungpook National University Research Fund, 2010. And this work is supported by the 2nd Brain Korea 21 Project in 2011.

## References

- [1] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, "Information hiding—a survey", in: Proceedings of IEEE special issue on protection of multimedia content, Vol. 87, No. 7, pp. 1062-1078, 1999.
- [2] H. Wang and S. Wang, "Cyber warfare—Steganography vs. steganalysis", Communications of the ACM, Vol. 47, No. 10, pp. 76-82, 2004.
- [3] W.L. Tai, C.M. Yeh, and C.C. Chang, "Reversible data hiding based on histogram modification of pixel differences", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 19, No. 6, pp. 906-910, 2009.
- [4] J. Fridrich, M. Goljan, and D. Rui, "Lossless Data Embedding - New Paradigm in Digital Watermarking", In Special Issue on Emerging Applications of Multimedia Data Hiding, No. 2, pp. 185-196, 2002.

- [5] M.U. Celik, G. Sharma, A.M. Tekalp, "Reversible data hiding", in: Proceedings of IEEE International Conference on Image Processing, Rochester, NY, pp. 157-160, 2002.
- [6] J. Tian, "Reversible data embedding using a difference expansion", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 8, pp. 890-896, 2003.
- [7] C.C. Chang, W.C. Wu, "A reversible information hiding scheme based on vector quantization", Proceedings of Knowledge-Based Intelligent Information and Engineering Systems (KES 05), pp. 1101-1107, 2005.
- [8] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, "Reversible data hiding", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No. 3, pp.354-361, 2006.
- [9] G. Xuan, Y.Q. Shi, Z. Ni, P. Chai, X. Cui and X. Tong, "Reversible data hiding for jpeg images based on histogram pairs", Lecture Notes in Computer Science 4633, pp. 715-727, 2007.
- [10] M. Fallahpour and M. H. Sedaaghi, "High capacity lossless data hiding based on histogram modification", IEICE Electron. Vol. 4, No. 7, pp. 205-210, 2007.
- [11] P. Tsai, Y.C Hu, H.L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting", Signal Processing, Vol. 89, No. 6, pp. 1129-1143, 2009.
- [12] K. Kim, M. Lee, H.Y. Lee, and H.K. Lee, "Reversible data hiding exploiting spatial correlation between sub-sampled images", Pattern Recognition, Vol. 42, No. 11, pp. 3083-3096, 2009.
- [13] W. Hong, T.S. Chen, Y.P. Chang, C.W. Shiu, "A high capacity reversible data hiding scheme using orthogonal projection and prediction error modification", Signal Processing, Vol. 90, pp. 2911-2922, 2010.
- [14] W. Hong, T.S. Chen, "A local variance-controlled reversible data hiding method using prediction and histogram-shifting", The Journal of Systems and Software, Vol. 83, pp. 2653-2663, 2010.
- [15] S.S. Maniccam, N.G. Bourbakis, "Lossless image compression and encryption using SCAN", Pattern Recognition, Vol. 34, pp. 1229-1245, 2001.
- [16] M. Awrangjeb, "An overview of reversible data hiding", in: Proceedings of the Sixth International Conference on Computer and Information Technology, pp. 75-79, 2003.
- [17] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, Digital Watermarking and steganography, volume 1. Morgan Kaufmann, 2 edition, 2008.

# Robust Video Watermarking Using Image Normalization, Motion Vector and Perceptual Information

Cedillo-Hernández Antonio<sup>1</sup>, Cedillo-Hernández Manuel<sup>1</sup>,

Nakano-Miyatake Mariko<sup>1</sup>, García-Vázquez Mireya S.<sup>2</sup>

<sup>1</sup>Postgraduate Section of Mechanical and Electrical Engineering School, National Polytechnic Institute of Mexico, Mexico City, Mexico

<sup>2</sup>Research and Development of Digital Technology Center, National Polytechnic Institute of Mexico, Tijuana, Mexico

**Abstract** – This paper proposes a video watermarking algorithm robust against geometric distortions, several attacks of signal processing and intentional common attacks for video. Image normalization is used to get geometric invariant feature of video frames. Watermark embedding and detection process are carried out in the Discrete Cosine Transform (DCT) domain. Watermark energy is computed adaptively using perceptual information and motion vectors, to get major Watermark imperceptibility and robustness. Watermark Imperceptibility is evaluated by conventional PSNR and perceptual video quality measurement, taking sufficiently good visual quality. Computer simulation results show the watermark robustness to common signal distortions such as contamination by noise, JPEG compression, geometrical distortions and video common intentional attacks: frame dropping, frame swapping and frame averaging, among others.

**Keywords:** Video Watermarking, Motion Vectors, Image Normalization, Block Classification, Perceptual Sensibility

## 1 Introduction

High speed computer networks, the Internet and the World Wide Web have revolutionized the way in which digital data is distributed. The widespread and easy accesses to multimedia contents and the possibility to make unlimited copy without loss of considerable fidelity arouse the need of digital rights management. Digital watermarking is considered as a technology that can serve this purpose. A large number of watermarking schemes have been proposed to hide copyright marks and other information in digital images, video, audio and other multimedia objects [1].

Usually watermarking algorithms for still images are not efficient when these are used in video sequences, because they are not considered the temporal redundancy of video signal and common attacks to the video signals [2]. In the case of a watermarking system for copyright protection, the embedded watermark should be imperceptible and robust against common attacks such as cropping, contamination by noise, filtering and compression [3]. In addition to the

requirement of imperceptibility and robustness techniques, furthermore the video watermarking must satisfy the following requirements: a blind detection, i.e. the detection process does not require the original video signal and the conservation of file size after the insertion of the watermark. Due to the redundancy existing in the video sequences, some intentional attacks, such as frame dropping, frame swapping and frame averaging, that try to destroy the embedded watermark should be considered in design of video watermarking algorithms [3].

In this paper, we propose a video watermarking algorithm based on image normalization, in which a watermark pattern is normalized using the same geometric factors obtained in the image normalization. The proposed algorithm uses three criteria based on Human Visual System (HVS) to insert a robust watermark, preserving their imperceptibility. The first criterion is based on the sensitivity of the HVS to different basic color channels, the second one is based on spatial deficiency of HVS, which is determined using the texture and edge masking proposed by [4], and the last criterion is based on tracing deficiency of the HVS in the regions with high motion speed in video sequences, which is determined using motion vector. The computer simulation results show the watermark imperceptibility and the robustness against common signal processing, geometrical distortions and some intentional attacks to the video sequence. The watermark imperceptibility is measured using the Peak Signal Noise Ratio (PSNR) and a quality assessment based on HVS proposed by [5].

The rest of the paper is organized as follows: In Section 2, the proposed system is described in detail and the evaluation results of the proposed system are shown in Section 3. Finally Section 4 provides some conclusions.

## 2 Proposed system

The proposed video watermarking system consists of several procedures, such as image normalization, block classification using the perceptual information based on HVS, watermark embedding and extraction. In this section, each procedure will be described.

## 2.1 Image normalization

The normalization procedure of a image  $f(x,y)$  consists of the following steps [6]:

1) Center the image  $f(x,y)$ , through the Affine Transformation as given by:

$$\begin{pmatrix} x_a \\ y_a \end{pmatrix} = A \cdot \begin{pmatrix} x \\ y \end{pmatrix} - d \quad (1)$$

where matrix  $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$  and the vector  $d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$  with

$$d_1 = \frac{m_{10}}{m_{00}}, d_2 = \frac{m_{01}}{m_{00}} \quad (2)$$

where  $m_{10}, m_{01}$  and  $m_{00}$  are the moments of  $f(x,y)$ , and  $d_1, d_2$  are the density center of  $f(x,y)$  [7]. This step eliminates the translation effect by assigning the center of the normalized image at the density center of the image. The resulting centered image is denoted as  $f_1(x,y)$ .

2) Apply a shearing transform to  $f_1(x,y)$  in the  $x$ -direction using the matrix  $A_x = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}$ . The resulting image is denoted by  $f_2(x,y)$ . This step eliminates shearing effect in the  $x$ -direction.

3) Apply a shearing transform to  $f_2(x,y)$  in the  $y$ -direction using the matrix  $A_y = \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix}$ . The resulting image is denoted by  $f_3(x,y)$ . This step eliminates shearing effect in the  $y$ -direction.

4) Scale  $f_3(x,y)$  in both  $x$  and  $y$  directions with the matrix  $A_s = \begin{pmatrix} \alpha & 0 \\ 0 & \delta \end{pmatrix}$  and the resulting image is denoted by  $f_4(x,y)$ . This step eliminates the scaling effect by forcing the normalized image to a standard size.

The final image  $f_4(x,y)$  is the normalized version. It is important to denote that each step in the normalization procedure is invertible, which allows us to convert the normalized image back to its original version. The determination of transformation parameters  $\beta, \gamma, \alpha$  and  $\delta$ , associated with the transforms  $A_x, A_y$ , and  $A_s$  are shown in detail in [9]. In the figure 1, an example of the original image and the normalized version is shown.

## 2.2 Perceptual Information

The proposed system uses three criteria, employed previously in [8], to embed an imperceptible and robust watermark in a video signal.

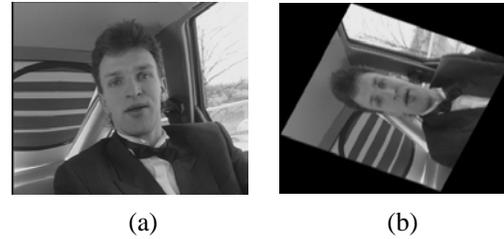


Figure 1 (a) Original image (b) Normalized version from (a).

These criteria are based on the less sensitivity of the HVS to the blue channel and the detail regions such as texture region, and also it's less ability to track a region with high speed motion. According to the human eye structure, the retina contains two types of photoreceptors, rods and cones. The last one is divided in 3, each sensitive to the three basic colors: Red, Green and Blue. The number of blue-sensitive cones is 30 times less than the number of cones sensitive to the other two colors [10]. Figure 2 shows the fraction of light absorbed by each of three types of cones, here R, G, B represents the cones sensitive to red, green and blue, respectively. This figure shows that the HVS is less sensitive to blue channel than the other two basic color channels (Red and Green).

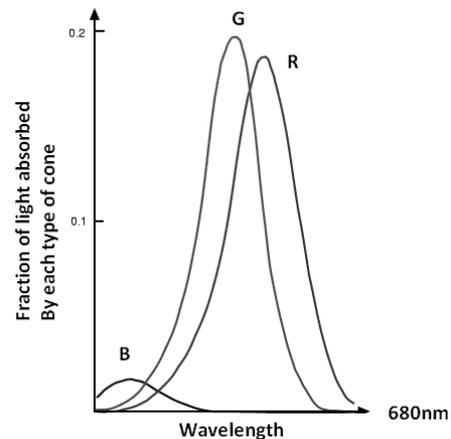


Figure 2 Sensitivity of the three types of cones R (red), G (green) and B (blue). [13]

The proposed algorithm embeds the watermark in the blue channel, taking advantage of the weakness of HVS. The blue channel of each video frame is divided into blocks with  $8 \times 8$  pixels and then the 2D Discrete Cosine Transform (DCT) is applied in each block. In the DCT domain, each block is classified into three categories: plain block, edge block and texture block according to the algorithm proposed in [4]. Figure 3 shows the result of applying the block classification algorithm to a video frame.



Figure 3 Block classification (a) video frame and (b) classification of blocks, black blocks are plain, gray blocks are textures and white blocks are edges, respectively.

The last criterion is based on deficiency of HVS to follow regions with high-speed motion. To classify regions of video frame by motion speed, the motion compensation prediction, which is a powerful tool to reduce temporal redundancy in MPEG coding, can be used. The macro-blocks (MB) with larger motion vector are classified as regions of high-speed movement, in which a watermark with greater energy can be embedded without causing degradation in the video signal. The magnitude of the motion vector is calculated using equation (3).

$$Mmv_i = \sqrt{mvh_i^2 + mvv_i^2}, \quad i = 1 \dots N_{MB} \quad (3)$$

Where  $mvh^2, mvv^2$  are the horizontal and vertical components of motion vector of the  $i$ th macro-block and  $N_{MB}$  is the total number of macro-blocks. To determine the region with a high-speed motion, we introduce a threshold value  $Th_{mv}$  and then each macro-block is classified as high-speed motion block and low-speed motion block (or without movement) as follows:

If  $Mmv_i < Th_{mv}$  then block are low-speed

If  $Mmv_i > Th_{mv}$  then block are high-speed

Where threshold  $Th_{mv}$  is calculated by equation (4)

$$Th_{mv} = \frac{1}{N_{MB}} \sum_{i=1}^{N_{MB}} Mmv_i \quad (4)$$

Figure 4 shows two consecutive video frames (Figure 4 (a) and (b)) and the motion vectors before and after classification (Figure 4 (c) and (d), respectively), the macro-blocks with arrows of some orientation are classified as high-speed blocks and blocks with black point are considered as low-speed blocks (or without movement).

### 2.3 Adequate watermark energy assignment

Combining the last two criteria: the classification of blocks (8x8) using DCT coefficients and the classification of macro-blocks (16x16) using motion vector, the watermark

embedding energy to the video frame is determined experimentally using 10 video sequences, which is shown in Table 1. In Table 1,  $B_8$  is the block of 8x8 pixels in video frames and  $MB_{MOTION}$  is the macro-block with high-speed motion, each macro-block contains 4 blocks  $B_8$ . This assignation of the watermark embedding energy is applied to the normalized video frames generated in the image normalization process. Figure 5 shows an example of the classification of blocks together with the watermark embedding energy determined using the criteria mentioned above.

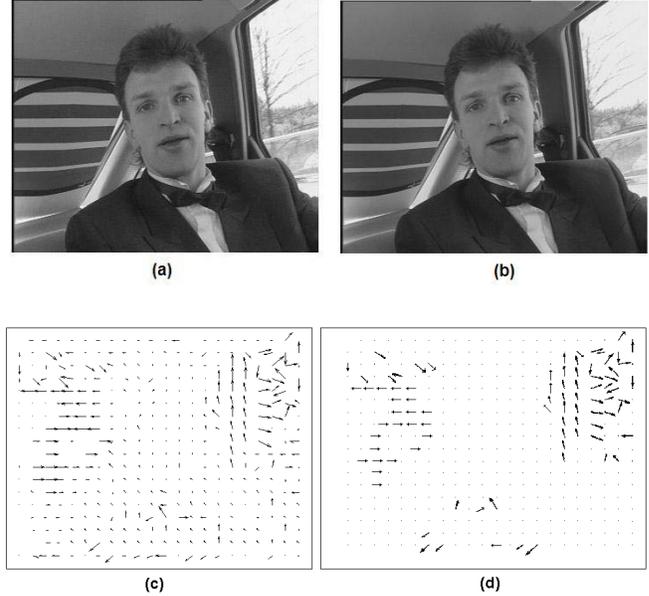


Figure 4 (a), (b) Two consecutive frames of video and (c) and (d) motion vectors before and after classification

Table 1 Watermark embedding energy

	$B_8 \in B_{plain}$	$B_8 \in B_{edge}$	$B_8 \in B_{tex}$
$B_8 \in Frame$			
$B_8 \in MB_{MOTION}$	0.9	1.2	1.8

### 2.4 Watermark embedding process

The watermark generation and embedding process are described as follows:

- 1) Apply the normalization procedure to the original image to get the normalized image.
- 2) Divide the normalized image in blocks of 8x8 pixels and get the watermark energy for every block mentioned in 2.3.

The result of this operation is a watermark embedding energy.

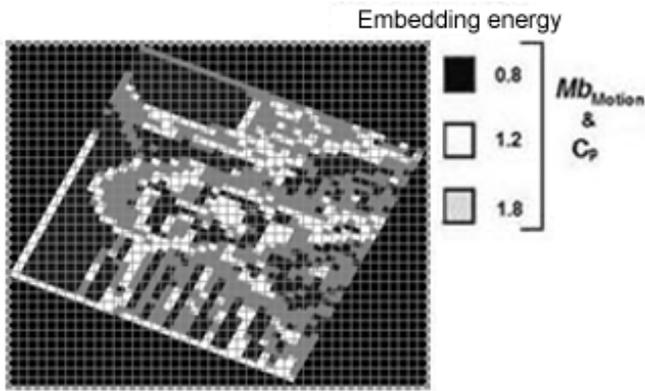


Figure 5 An example of the embedding watermark energy to each block

3) Generate the 2-D pseudo-random pattern R of the same size of original image with any key. Because this pattern is only used as the support of the watermark pattern, any key can be used and it is not necessary to save.

4) Generate a watermark vector  $W = [w_1, w_2, \dots, w_n]$  using a user's secret key, where  $w_i = \{1, -1\}$ ,  $i=1..n$ .

5) Create a mask image M, which is a binary image, taking 1s within normalized image area and 0s elsewhere, to generate masked pseudo-random pattern MR with same scale and rotation factors as these of the normalized image.

6) Watermark vector W is multiplied by a watermark energy vector determined by the previous section.

7) Divide the watermark vector W into N groups of L elements, where L must be one number from 1 to 22 (number of coefficients that compose the middle frequency range in DCT domain), thus, for example, if L = 5 and size of W is 500, the number of groups  $N = 500 / 5 = 100$ .

8) The coefficients in middle frequency range of each block are replaced with L elements of each group of watermark sequence. Then apply the IDCT to each watermarked block to get watermarked pattern  $MR_W$ .

9) Apply the inverse normalization to the watermarked pattern  $MR_W$  to get watermarked pattern WP with same size as the cover image.

10) WP is embedded into the original image additively with a gain factor  $\alpha_2$ . This produces the watermarked image.

$$I_w = I_o + (WP \cdot \alpha_2) \tag{5}$$

where  $I_o$  and  $I_w$  are original and watermarked image, and WP is watermark pattern generated by step 9.

The whole procedure is equivalent to embedding the watermark into the DCT domain of the normalized image. The adequate watermark embedding energy allows embedding strong watermark without causing any visual distortion to watermarked image. The figure 6 shows the watermark generation and embedding procedures.

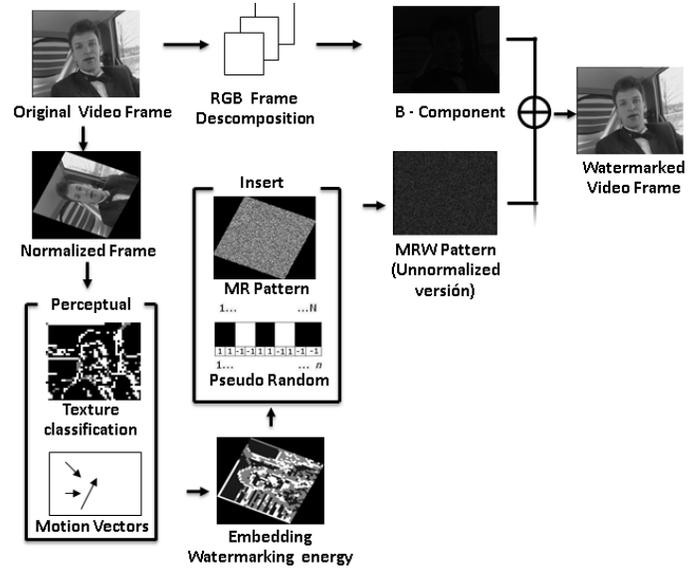


Figure 6 Watermark embedding process

## 2.5 Watermark Extraction

The process of the watermark detection is as follows:

1) Apply the image normalization procedure to watermarked image to get the normalized watermarked image.

2) Apply the DCT to each block of the normalized watermarked image, the middle range of the DCT coefficients CW are extracted. From CW the watermark vector  $\hat{W}$  is extracted by (6).

$$\hat{W} = [w_1, w_2, \dots, w_{N-1}, w_N] \tag{6}$$

$$w_k = \text{sign}(CW_k)$$

where  $w_k$  is the extracted watermark sequence (L bits) from k-th block and  $\text{sign}$  is a sign function.

### 3 Experimental Results

To evaluate the proposed system, we used 20 video sequences with YUV-CIF format at 30 FPS. All video data have at least 150 frames which are available in [11]. The proposed system is evaluated from the watermark imperceptibility and robustness points of view.

#### 3.1 Watermark imperceptibility

We evaluate watermark imperceptibility of the proposed system using numerical evaluation: Peak Signal Noise Ratio (PSNR) and perceptual objective evaluation proposed in [5]. Figure 7 shows the PSNR value calculated between original and watermarked frames in the proposed method. Perceptual quality assessment proposed in [5] evaluates video quality in perceptual manner, which gives the quality index calculated using image structure and motion vector. The quality index is in range [0.0,1.0], when video sequence under evaluation is numerically identical with its original version, its quality index is 1.0. The quality index of the watermarked video sequence generated by the proposed algorithm is approximately 0.96, which means that degradation of the watermarked video quality is minimum by HVS.

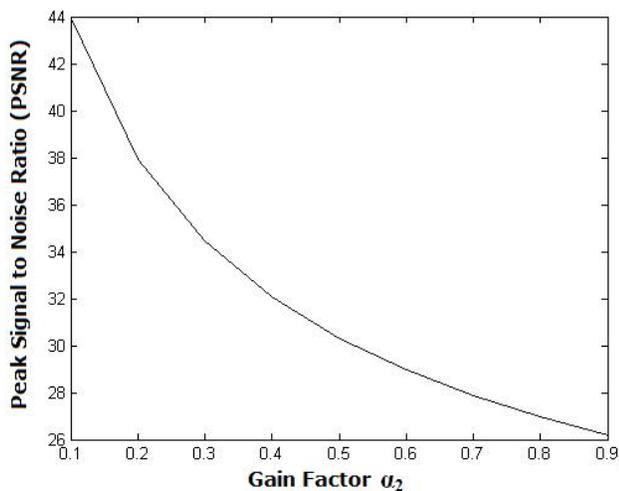


Figure 7 Peak Signal to Noise Ratio of the proposed method.

#### 3.2 Watermark Robustness

To evaluate watermark robustness of the proposed algorithm, the watermarked video sequences are attacked by some common signal and image processing, such as codification rate change, noise contamination by impulsive noise and Gaussian noise, geometrical distortions, frame dropping, frame swapping and frame averaging. The results are shown in Table 2, in which the bit-error-rate (BER) of the extracted watermark bit sequence respect to the embedded one is shown. In all cases the values correspond to the average values using the 20 video sequences with YUV-CIF format. Frame dropping, frame swapping and frame averaging are intentional attacks for video sequence. Frame

dropping is drop some frames from video sequence, frame swapping is interchange two frames and frame averaging is generate one frame taking the average of some consecutive frames. Due to that watermark sequence is embedded through temporal video frames in the proposed algorithm; embedded watermark is robust against these types of attacks. Table 3 shows the performance comparison of our proposed system and the recently reported system by Soumik which present an investigation similar to our proposal because it considers the content of video as a sequence of images and the watermark is inserted frame by frame with an invisible watermarking scheme, further extraction of the watermark is blind and is reported as a method extremely appropriate in areas such as copyright and fingerprinting [12].

Table 2. The embedded watermark signal is sufficiently robust to geometrical distortions and common signal processing. .

Attack	Watermark frame attack	BER
Geometric Rotation		0.0960
Aspect Ratio (1:2)		0.070
Gaussian noise		0.097
Impulsive noise		0.098
MPEG Codification (50% quality)		0.091

### 4 Conclusions

In this paper, we proposed a video watermarking algorithm based on image normalization, in which watermark embedding and detection process are carried out in the

Discrete Cosine Transform (DCT) domain. The watermark extraction has done blindly i.e., neither the watermark nor the original video is needed at the time of the watermark extraction. The proposed algorithm uses three criteria based on Human Visual System (HVS) to embed a robust watermark, preserving their imperceptibility. These criteria are based on the sensitivity of the HVS to different basic color channels, the texture and edge masking classification and the estimation of the motion vectors in video sequences. The computer simulation results show the watermark imperceptibility and the robustness of the scheme against common signal processing, geometrical distortions and some intentional attacks to the video sequence.

Table 3. Robustness comparison between the proposed video watermarking and Soumik method's [12]

Attack	Method proposed	Soumik method's [12]
Geometric Rotation	Detected	-
Aspect Ratio	Detected	-
Gaussian noise	Detected	-
Impulsive noise	Detected	-
MPEG Codification	Detected	-
Frame Dropping	Detected	Detected
Frame Swapping	Detected	Detected
Frame Averaging	Detected	Detected

## 5 References

- [1] Sourav Bhattacharya, T. Chattopadhyay and Arpan Pal, "A Survey on Different Video Watermarking Techniques and Comparative Analysis with Reference to H.264/AVC". IEEE 10th International Symposium on Consumer Electronics, June 2006, pp.1-6.
- [2] M. Swanson, B. Zhu y A. Tewfik, "Multi-resolution scene-based video watermarking using perceptual models", IEEE J. Select areas Communication, vol. 16, no. 4, pp. 540-550, May. 1998
- [3] Wolfgang, R.B, C.I. Podilchuk y E.J. Delp, "Perceptual watermarks for digital images and video", Proceeding IEEE, vol. 87, no. 7, pp. 1108-1126, 1999.
- [4] H.Y. Tong and A.N. Venetsanopoulos, "A Perceptual model for JPEG applications based on block classification, texture masking and luminance masking", Int. Conf. on Image Processing (ICIP), vol. 3, pp. 428-432, 1998.
- [5] Z. Wang, L. Lu and A. C. Bovik, "Video quality assessment based on structural distortion measurement", Elsevier Signal Processing: Image Communication, vol. 19, pp. 121-132, 2004
- [6] M. Cedillo-Hernández, M. Nakano-Miyatake and H. Perez-Meana, "Robust Watermarking Technique based on Image Normalization" (Spanish), Revista Facultad de Ingeniería Universidad de Antioquia, no. 52, pp. 147-160, Marzo 2010.
- [7] M. K. Hu, "Visual Pattern Recognition by Moment Invariants", IRE Trans. on Information Theory, vol. 8, pp. 179-187, 1962.
- [8] A. Cedillo, M. Nakano, H. Perez and L. Rojas, "Watermarking Technique for MPEG Video using Visual Sensibility and Motion Vector" (Spanish), Revista Información Tecnología, vol. 19, no. 2, pp. 81-92, 2008.
- [9] P. Dong, J. B. Brankov, N. P. Galatsanos, Y. Yang and F. Davoine, "Digital Watermarking robust to geometric distortions," IEEE Trans. on Image Processing, vol. 14, no. 12, pp. 2140-2150, 2005.
- [10] K. Sayood, "Introduction to Data Compression", 2nd Edition, Morgan Kaufmann Publishers, 2000.
- [11] [http://www.antonioch.com/public/?page\\_id=241](http://www.antonioch.com/public/?page_id=241)
- [12] Soumik Das, Pradosh Bandyopadhyay, Shauvik Paul, Chaudhuri Atal and Monalisa Banerjee, "An Invisible Color Watermarking Framework for Uncompressed Video Authentication", International Journal of Computer Applications, vol. 1, no. 11, pp. 22-28, February 2010.
- [13] <http://science.slashdot.org/story/08/04/08/2213222/What-Font-Color-Is-Best-For-Eyes>

# Log File Modification Detection and Location Using Fragile Watermark

Liang Xu and Huiping Guo  
Department of Computer Science  
California State University at Los Angeles  
Los Angeles, CA, USA

**Abstract-** *In this paper, a novel algorithm is proposed to protect the integrity of log files. Unlike other existing schemes, the proposed algorithm can detect and locate any malicious modifications made to the log files. Furthermore, massive deletion of continuous data can be classified and identified. Security analysis shows that the algorithm can detect modifications with high probability which is verified by the experimental results. In real application, the proposed algorithm can be built into the generation procedure of the log files, so no extra process is needed to embed the watermark for the log files.*

**Keywords:** log file; modification detection; modification location; fragile watermarking

## 1. Introduction

Nowadays logs are widely used to keep records of valuable information. Some examples are web server logs[1] and database logs[2]. Many log files contain records related to computer security, the concern of log file security has increased greatly because of the ever-increasing threats against systems and networks[3]. Lots of implementations have been created which place a greater emphasis on log file security. Most have been based on a proposed standard, RFC 3195, which was designed specifically to improve the security of syslog[4], a protocol presenting a spectrum of service options for provisioning an event-based logging service over a network. Among these implementations, transmission damage detection and message digest are aimed to protect the integrity of the log files. The transmission damage detection mainly focuses communication security. If log messages are damaged in transit, mechanisms built into the link layer as well as into the IP[5]and UDP[6] protocols will detect the damage and simply discard a damaged packet. When message digest method is used, log file integrity checking involves calculating a message digest for each file and storing the message digest securely to ensure that changes to archived logs are detected. The most commonly used message digest algorithms are MD5 and keyed Secure Hash Algorithm 1 (SHA-1).

Although the transmission damage detection and the message digest method detect the modification of the log files, neither of them is able to locate the modifications. Once authentication is detected false, the whole message packet or the whole log file will be discarded. The usability of the log file is rather low once a modification is detected. The algorithm proposed in this paper not only detects any malicious modification but also locate the modifications. Except those located modification areas, the rest of the log file is still trustable. In the case of malicious modification, the usability of the log file is greatly improved.

## 2. Related Work

In paper [7], Dr Guo and her colleagues proposed a novel algorithm of applying fragile watermarking in verifying the integrity of numerical and categorical data streams at the application layer. In their algorithm, watermarks are chained and embedded directly to the least significant digits of data groups to protect the authentication of data streams. Illegal modifications to the watermark embedded data streams can be effectively detected and located.

In this paper, we extend Dr Guo's algorithm to the application of log file generation and storage. The algorithm is alternated accordingly to the specific application of log file generation.

## 3. Proposed Algorithms

In our proposed algorithm, fragile watermarks are calculated and embedded on the fly of the generation of log files. No extra process is required for the watermark calculation and embedding. The embedded watermark can detect as well as locate any modifications made to the log file.

### 3.1 Algorithm assumptions and advantages

In the proposed security log file scheme, we assume the log files are in human readable form composed of clear text. Different from numeric data streams which are widely used in multimedia data, most log files are text documents composed of character lines. We consider each separated text line as a single log data element.

We also assume a small buffer at the log file generator side which is used to store a group of log file data. As our scheme is group based, we collect the generated log entries in the same order as they are generated into the buffer until they form a complete group. Watermarks are calculated and embedded into the buffered log data, the watermarked data are then written to the log file storage. In this process, watermarks are calculated and embedded at the same stage as log file generation. No extra round of process is required.

Our algorithm is designed to detect and locate modifications of log files using fragile watermark. It possesses the following advantages:

*Distortion free:* The embedded watermark does not change the log file content. The information of the log file remains exactly the same meaning after the watermark embedding. Data is not distorted in any means.

*Invisibility:* The embedded watermark does not degrade the perceptual quality of the log file. The watermark embedded log file remains in clear text format and conveys the exact same message as the log file generated without employing our scheme. Actually the marking is barely observable and the embedded watermark is invisible.

*Blind verification:* The original unmarked log data are not required for watermark verification.

*Location:* The algorithm can detect as well as locate the modification to a log file. Any modifications, including changing, deletion or insertion of log entries, can be narrowed and located to specific groups.

### 3.2 Watermark embedding

Table 1 gives the notations used in watermark embedding algorithm. The algorithm collects continuously the generated log data elements, i.e., log text lines, into a group and

embeds the watermark. The watermark embedding mainly consists of two parts of processes, grouping and embedding. For each collected single log data line  $e_i$ , we compute a secure hash  $h_i$ . We check whether this data element  $i$  is a synchronization point based on its hash value  $h_i$ . A log data element  $e_i$  is defined as a synchronization point if and only if

$$h_i \bmod m = 0$$

**Table 1** Watermark embedding notations

Notations	Descriptions
$e_i$	A single log data line
$h_i$	secure hash of a single log data line
$m$	Secure parameter
$L$	Lower bound of the group size
$H(i)$	Group hash value
HASH	Secure hash function
WH	Watermark hash value
$W$	extracted watermark

Note that throughout the whole algorithm, we trim the log line  $e_i$  when we compute the secure hash  $h_i$ . Since we embed the watermark bit at the end of each line as a space for watermark bit '1' and none for watermark bit '0'. And whether the last character of a log line is a space will be used to calculate the watermark.

The secure parameter  $m$  and the lower bound of the group size  $L$  decide how the log data elements are grouped. We only consider a group is formed and proceed to the embedding process if the log data element  $e_i$  is a synchronization point and the number of elements in the group is larger than  $L$ . Otherwise, the log data element is buffered until a group is formed. The parameter  $L$ , the lower bound of the group size, is set to prevent small groups from forming for security reasons.

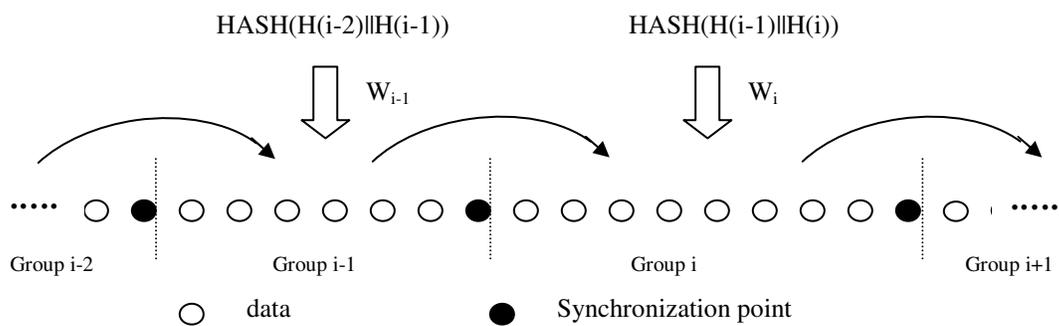


Figure 1. Watermark embedding

As shown in figure 1, the watermark embedding is group based, all data between two synchronization points (providing the group size is larger than  $L$ ), including the last synchronization point, form a group. A group hash value  $H(i)$  is computed on all the individual hash values  $h_i$  of data elements within the group. Then a watermark  $W$  is constructed based on both the previous group hash value  $H(i-1)$  and the current group hash value  $H(i)$ . We cut the

length of the watermark (number of bits of the watermark) as the same as the number of data elements in the current group. The watermark is then embedded to the current group of data by inserting a space to the end of the trimmed log element line if the watermark bit is '1'. In this way the embedded watermarks are actually chained. Even if a whole group of data is deleted, the deletion is still detectable because of the forward chain.

**Algorithm 1** Watermark embedding

---

```

1: clear buff
2: fillBuff(buff) //See Algorithm 2
3: H0 = getGoupHashe(buff) //See Algorithm 3
4: while true do
5:     fillBuff(buff)
6:     H1 = getGoupHash(buff)
7:     WatermarkEmbed(buff, H0 , H1 ) //See Algorithm 4
8:     Write data in buff into log file
9:     Clear buff
10:    H0 = H1
11: end while

```

---

**Algorithm 2** fillBuffer(buff)

---

```

1: k ← 0
2: while receive an incoming data element ei do
3:     buff(k) = ei // collect and buff log element line ei
4:     k++
5:     if (ei is a synchronization point) and (k >= L) then
6:         return
7:     end if
8: end while

```

---

**Algorithm 3** getGroupHash(buff,K)

---

```

1: k ← number of log data elements in buff
2: for each data element ei in buff do
3:     hi = HASH( ei ) //log data element line ei is trimmed.
4: end for
5: H = HASH2( h1, h2, . . . , hk )
6: return H

```

---

**Algorithm 4** watermarkEmbed(buff, H0, H1)

---

```

1: WH = HASH2(H0 , H1 )
2: k ← number of data element in buff
3: W = extractBits(WH) //See Algorithm 5
4: for i = 1 to k do
5:     if w(i) == 1 then
6:         Insert a space at the end of log data element line buff(i)
7:     else if w(i) == 0 then
8:         Do nothing.
9: end for

```

---

**Algorithm 5** extractBits(WH, k)

---

```

1: if length(WH) >= k then
2:     W = concatenation of first k selected bits from WH
3: else
4:     m = k - length(WH)
5:     W = concatenation of WH and extractBits(WH, m)
6: end if
7: return W

```

---

The above algorithms 1 – 5 illustrate how the watermark embedding process works. We will need one buffer in the process to collect and temporarily store a group of log data. As shown in algorithm 2, the buffer is filled until the log element line  $e_i$  is a synchronization point and the current number of element lines in the buffer is not less than  $L$ , a group is then formed. Algorithm 3 calculates the hash value  $h_i$  for each individual element line, and calculates the group hash value based on the hash value  $h_i$  of all the individual

elements in the group. Algorithm 5 extracts the same number of bits as the number of data elements in the group from a given watermark hash value  $WH$ , the extracted bits form the watermark to be embedded to the group data. Algorithm 4 computes the watermark hash value based on the group hash value of the previous data group,  $H_0$ , and the group hash value of the current group,  $H_1$ , watermark is extracted from the resulting watermark hash value and embedded to the current group of data. Algorithm 1 repeats the watermark embedding process infinitely.

**3.3 Watermark verification**

As in the watermark embedding, watermark verification uses synchronization points to group log data elements into a group. A watermark which is constructed from the group hash value of the previous and the current group is checked against the extracted watermark from the current group. If the two watermarks match, no modification will be detected; the preliminary verification value of the current group is true.

**Table 2** Watermark verification notations

Notations	Descriptions
G	A group
pV1	Preliminary verification value of the current group G1
pV0	Preliminary verification value of the previous group G0
V1	Final verification value of the current group G1
V0	Final verification value of the previous group G0
V(-1)	Final verification value of the group before the previous group, i.e., G(-1)

Table 2 gives the notations used in watermark verification algorithm. To verify the integrity of the log file, we need one buffer to collect and temporarily store a current group of data  $G1$ ; we also need two pointers to point the beginning position and ending position of the previous group  $G0$ . The latter is used to locate the modifications made to the previous group. The watermark verification algorithms use the same initial group hash value  $H_0$ , secret parameter  $m$ , and the group size lower bound  $L$  as the watermark embedding algorithms. We also define two verification values here, the preliminary verification value and the final verification value. Notation  $pV0$  indicates the preliminary verification value for the previous group and  $pV1$  indicates the preliminary verification value for the current group; Similarly,  $V0$  denotes the final verification value for the previous group and  $V1$  denotes the final verification value for the current group.  $V(-1)$  denotes the final verification value we finalized at an earlier time for the group before the previous group. The preliminary verification value  $pV$  for a group, which results from the matching check between the watermark constructed and the watermark extracted, may be different from the final verification result  $V$  of that group. The final verification result for a group  $V$  indicates whether the group is authentic. It is decided on the following watermark verification rational table.

**Table 3** Watermark verification rationale

Cases	Precondition			Rational Results			Note
	pV0	pV1	V(-1)		V0	V1	
1	true	true	true		true	true	No modification detected
2	true	false	true		true	false	
3	false	true	false		true	true	
4	false	true	true	missing group(s)	true	true	One or more groups may be missing before G0
5	false	false	true		false	?	V1 to be finalized at next stage using next group G2
6	false	false	false		false	?	V1 to be finalized at next stage using next group G2

Table 3 shows the rational of watermark verification. Since the watermark embedded in the current group G1 is a chained watermark computed from the value of the previous group G0 and the current group G1, if the watermark constructed from G0 and G1 matches the watermark extracted from G1, a positive preliminary verification result for the current group (pV1 = true) indicates the originality of the previous group data G0 and the current group data G1 (V0 = V1 = true). The situation is more complex if the two watermarks do not match. In this case, we can only say the preliminary verification for the current group is false (pV1 = false). We will still need to investigate the integrity of the next group before ascertaining the final verification result of the current group (forward check). Similarly, the final verification result for the previous group (V0) depends not only on the preliminary value of its own (pV0) but also on the preliminary verification result of the current group (pV1). If the preliminary verification for the current group is positive (pV1 = true), which also indicates a positive verification for the previous group (V0 = true), the false preliminary verification pV0 means either one or more group are missing between the G(-1) and G0, or the false verification result is brought forward from the false verification of G(-1).

There are some interesting cases where preliminary verification is false. This may be due to modifications in the previous group or those in the current group, or missing groups. In Table 3, one or more groups are missing before G0 in the case 4. Since in this case the verifications of both the previous group V0 and the group before previous group V(-1) are true, which excludes the possibility of modifications in both G(-1) and G0, the only explanation for the false pV0 is that one or more entire groups between G(-1) and G0 have been deleted. It is these deletions that cause the preliminary verification of the previous group pV0 to be false. Because watermarks are chained together, no matter how many groups are deleted, the deletions can be correctly detected.

---

#### Algorithm 6 Watermark verification

---

```

1: clear buff
2: pV0,V0, V(-1) ← true
3: fillBuff(buff) //See Algorithm 2
4: H0 = getGoupHash(buff) //See Algorithm 3
5: while true do
6:     fillBuff(buff)
7:     H1 = getGoupHash(buff) //See Algorithm 3

```

```

8:     WatermarkVerify(buff, H0, H1, pV0, V0, V(-1)) // See
Algorithm 7
9:     Clear buff
10:    H0 ← H1
11:    pV0 ← pV1
12:    V(-1) ← V0
13:    V0 ← V1
14: end while

```

---

#### Algorithm 7 watermarkVerify(buff, H0, H1, pV0, V0, V(-1))

---

```

1: WH = HASH(H0, H1)
2: k ← number of data elements in buff
3: W1 = extractBits(WH, k) //See Algorithm 5
4: for i = 1 to k do
5:     if end of log element line buff(i) is a space
6:         W2(i) ← 1
7:     else
8:         W2(i) ← 0
9: end for
10: IF (W1 == W2) then
11:     V1 = pV1 = V0 = true
12: else
13:     pV1 = false
14:     V1 = V0 & pV1
15: end if
16: if V0 == false then
17:     The previous group may be tampered
18: end if
19: if pV0 == false && V0 == true && V(-1) == true then
20:     One or more groups between the previous group and the
        current group may be missing
21: end if

```

The algorithms 6 and 7 describe the watermark verification process. A group of log data lines are filled into the buffer, the group hash value H1 is computed. Algorithm 7 verifies the watermark extracted from a group of log data through watermark matching. Watermark hash value WH is computed based on the current group hash value H1 and the previous group hash value H0. (For the first group of data, the initial watermark hash value WH0 is introduced.) Watermark W1 is constructed from WH. Meanwhile, watermark W2 is extracted from the group of data lines. If W1 equals to W2, the watermark verification for the current group succeeds and the verification result is set to true; otherwise, the verification fails and a false preliminary verification result is set to the current group. Algorithm 6 repeats the above process for an infinite log file check.

## 4. Implementations

To directly create the watermarked security log file, we integrate our watermark embedding mechanism with the logging service. Figure 2 displays the integration structure of integrating watermark embedding mechanism into the log4j logging. Based on this structure, the printing methods of a logger instance in log4j will output the log data to a socket. The watermark embedding mechanism collects the log data element lines from the socket and invokes the watermark embedding method. Thus the watermark embedding process is triggered and the watermarked log data are produced. The watermarked log data stream will finally be written to the predefined destination through the appending method of an appender and the write method of the output stream in log4j.

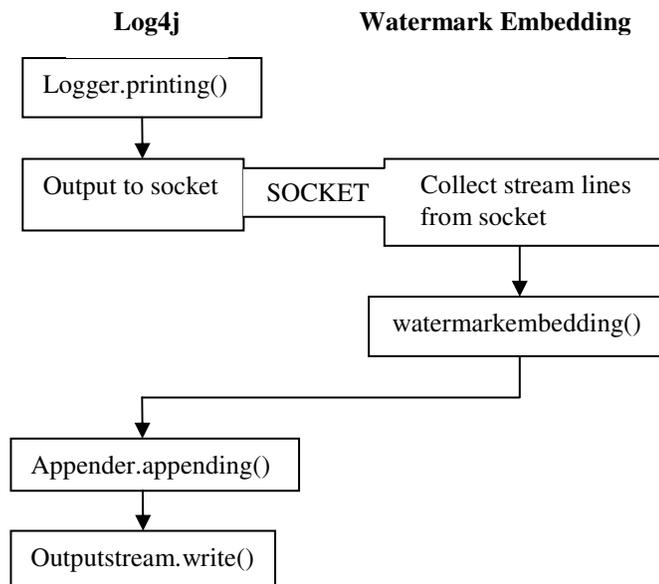


Figure 2. Integration with log4j

## 5. Experimental Results and Conclusion

### 5.1 Experiment scenarios and results

We test the validity of the mechanisms under various scenarios. We choose log files of different types and different sizes and embed watermarks into the log files using the watermark embedding mechanism. Then we modify the watermarked log file in various ways manually or using the log file modification program. The watermark verification mechanism is applied to the manipulated log files to detect and locate the modifications made to the log files.

#### 5.1.1 Successful verification without modification

First, we verify the original watermarked log file without any modification. No modification is detected in this scenario. The verification result turns out to be successful.

#### 5.1.2 Single data element modification

We change, delete or insert a single data element line in the watermarked log file and then run the verification process. The results show that if a regular data line is modified, the modification is easily detected and located to the modified group. If the modified line happens to be a synchronization point (this may happen at a less possibility rate), the grouping of the data is affected, the detection will locate the modification to the group before the synchronization point (including the synchronization point) and the group after the synchronization point.

#### 5.1.3 Multiple data elements modification

We change, delete or insert multiple data element lines in the watermark embedded log file and then run the verification process. The same as single data element modification, if regular data lines (not synchronization points) are modified, as long as the grouping of data is not affected, the modifications are easily detected and located to the modified groups. The detection becomes more complex if the grouping of data is changed because of the modification or change of synchronization points. If the modification is related to synchronization points, the detection mechanism locates the modification to the group before the modified synchronization point (including the synchronization point) and the group after the synchronization point.

The experimental results also show that the mechanism is able to detect and locate the modifications when one or more entire groups are deleted or inserted although these cases rarely happen in real world. If the inserted two or more groups are continuous groups with well-embedded watermarks (this can be made by repeating continuous groups of data), the classification for the insertion fails which is mistakenly classified as missing groups. But the mechanism is still able to locate the modification before the inserted group and before the group after.

### 5.2 Security analysis

There is a probability that the log file watermark mechanism fails to detect a modification made to the log file. In the case that the extracted watermark from the modified group happens to match the watermark constructed from the modified group and the group before, the preliminary verification of the modified group will succeed, the modification will not be detected. Assume the size of a group is  $l$ , after a modification is made to the group, the probability that the preliminary verification of the group will succeed (i.e., the false negative rate) is as

$$1/2^l \leq 1/2^L$$

$L$  is defined as the lower bound of group size. The group false negative rate monotonically decreases with the value of  $L$ .

With group size fixed, the false negative rate of any affected group in an attack remains the same no matter how many

elements in the group are changed. Therefore, we consider the attack at group level and assume that  $g$  groups are affected in attacks. The overall false negative rate, which is the probability that at least one affected group is verified successfully, can be computed as

$$1 - (1 - 1/2^L)^g$$

The overall false negative rate is monotonic increasing with  $g$  and decreasing with  $L$ . Providing a relative large  $L$  and small  $g$ , which is as in most cases in the real world, we can limit the overall false negative rate to a rather small rate.

During the experiment, we noticed that the lower bound value of group size (i.e.,  $L$ ) has an impact on the number of affected groups (i.e.,  $g$ ) in simulated attacks. With a smaller  $L$  value, there are generally fewer data lines in an average group, the modification of single element and especially multiple data elements is more likely to affect the grouping of data. Thus, with a smaller size for each group, the modification location accuracy for a group increases while more groups may be affected and be located in overall modification locations.

### 5.3 Performance analysis

**Table 4** Watermark embedding time

Log file size	Generation time without watermarking	Generation time with watermarking	Time difference (watermarking time)
1 KB	2 ms	2 ms	0
5.03 KB	5 ms	6 ms	1 ms
12.1 KB	9 ms	11 ms	2 ms
133 KB	43 ms	48 ms	5 ms
367 KB	67 ms	74 ms	7 ms
463 KB	74 ms	84 ms	10 ms
1,285 KB	160ms	177 ms	17 ms
6,545 KB	400 ms	450 ms	50 ms
10,241 KB	955 ms	1105 ms	150 ms
27,179 KB	1510 ms	1676 ms	166 ms

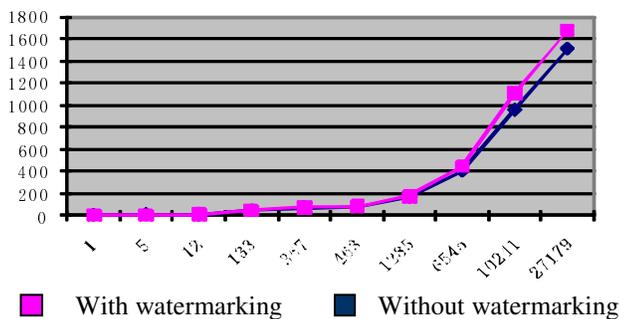


Figure 3. Watermark embedding time

Table 4 gives approximate time to watermark log files in different sizes using our watermark embedding mechanism. Figure 3 illustrates the time increase chart for the log generation without watermarking and with watermarking as the log file size increases. Both the table and the figure show

that the watermarking process doesn't cost much extra time compared to the original log generation. It costs about one tenth more time to generate a watermarked log file than to produce the original log file, which proves the feasibility of the proposed scheme.

### 5.4 Conclusion

This paper gives a naive way to protect the integrity of the log files. According to our scheme, modifications, including change, insertion and deletion, made to a log file can be detected and located to a small group area. The algorithm is described in details. Implementations and experimental results are discussed. We could embed the scheme into the generation procedure of the log files thus no extra process is needed to embed the watermark for the log files.

The algorithm can be applied to any text files. For any text files, fragile watermarks can be calculated and embedded, thus any modifications made to the text files can be detected and located to small areas using the watermark verification mechanism described in this paper.

### 6. References

- [1] M. Bruce, (1999, Jul.). A Brief Introduction to Server Logs. [Online]. Available: <http://evolt.org/node/233/>
- [2] Wikimedia Foundation, Inc., (2011, Feb.). Transaction log. [Online]. Available: [http://en.wikipedia.org/wiki/Database\\_log](http://en.wikipedia.org/wiki/Database_log)
- [3] K. Kent, M. Souppaya, "Guide to Computer Security Log Management", National Institute of Standards and Technology, Technology Administration, U.S. Department of Commerce, Special Publication 800-92
- [4] C. Lonvick, "The BSD Syslog Protocol", RFC 3164, August 2001.
- [5] J. Postel, "Internet Protocol", STD 5, RFC 791, September 1981.
- [6] F. Baker, "Requirements for IP Version 4 Routers", RFC 1812, June 1995.
- [7] H. Guo, Y. Li, A. Liu and S. Jajodia, "A fragile watermarking scheme for detecting malicious modifications of database relations," *Information Sciences*, vol. 176, pp. 1350-1378, May 2006

# On Energy Efficiency of Elliptic Curve Cryptography for Wireless Sensor Networks

Tinara Hendrix<sup>1</sup>, Michael Bimberg<sup>2</sup>, and Dulal Kar<sup>1</sup>

<sup>1</sup>Department of Computing Sciences, Texas A&M University-Corpus Christi, Corpus Christi, Texas, USA

<sup>2</sup>Department of Computer Science, Southeast Missouri State University, Cape Girardeau, Missouri, USA

**Abstract** - Energy efficiency is a primary concern in Wireless Sensor Networks (WSN). This is due to the fact that WSNs are powered battery, and hence the life of WSNs becomes limited by the battery life. Efforts to replace depleted batteries are not feasible if WSNs are often deployed with thousands of sensor nodes, possibly in inaccessible, hostile, hazardous, or remote territories. Though WSNs are energy-constrained, adequate level of security is often desired in many applications of WSNs that guarantees data integrity and confidentiality. However, security protocols introduce extra energy overhead to a sensor node due to additional processing and communications associated with the protocols. Thus it is important to seek security protocols and solutions for WSNs that are energy-efficient but also effective. Security is commonly implemented through symmetric key cryptography which requires a key exchange mechanism to establish a key between the communicating parties. One well known key exchange protocol that has been found to be suitable for WSNs is the Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol. In this work, we determine energy requirements of ECDH for various key sizes in order to aid the selection of a security level for the key exchange protocol while maintaining the best possible energy efficiency for WSNs. Particularly, we study energy requirements of TinyECC software package for the ECDH key exchange protocol used in Imote2 sensor nodes for key sizes of 128 bits, 160 bits, and 192 bits as recommended by NIST (National Institute of Standards and Technology). Our study shows very favorable results for the 160-bit key ECDH exchange algorithm.

**Keywords:** Wireless sensor networks; Elliptic Curve Cryptography; Elliptic Curve Diffie-Hellman; TinyECC

## 1 Introduction

Wireless sensor networks have become more widely used for information gathering. One of the main concerns when using WSNs is the finite power supply of the motes that make up the network [1].

Often, these networks are deployed in areas that are unreachable, making it impossible to replenish the power supply physically. It may also be infeasible or economically unattractive in some situations to do so. Essentially the life of

a WSN becomes limited by the life of its power supply. Security protocols running in a WSN consume energy due to communication as well as computation [2]. It is important to reduce energy requirement to support security needs in a WSN as much as possible. For key-exchange, many energy-efficient protocols and algorithms have been proposed in literature. Several of these protocols involve a modified version of the Elliptic Curve Diffie-Hellman (ECDH) which uses Elliptic Curve Cryptography (ECC) [3]. These protocols maintain a sufficient security level by using keys that meet the recommended minimum sizes suggested by National Institute of Standards and Technology (NIST). In cases where data collected from a WSN is not highly sensitive, relatively a key of smaller size for ECDH can be used to reduce computation time. In turn, the reduction of computation time should lower energy consumption. Our work involves determining energy consumption when using different key sizes in conjunction with ECC for key sharing using Diffie-Hellman key exchange protocol. Particularly, we study energy requirements of TinyECC software for various key sizes used in the ECDH key exchange protocol for imote2 sensor nodes. Our results can be used to decide on the tradeoff between the key size and the energy requirement for the ECDH protocol. Specifically among the key sizes recommended by NIST, we find that the 160-bit implementation of ECC in TinyECC software is very energy efficient for imote2 sensor nodes.

In section 2, we provide brief reviews of the principles of Elliptic Curve Cryptography (ECC), ECDH, TinyECC software, and the imote2 sensor platform. Subsequently in section 3, we describe the methodology to analyze the energy requirements of various ECDH key sizes on imote2 sensor nodes. Finally in section 4, we present the results of our study on energy efficiencies of various key sizes used in the ECDH key exchange protocol.

## 2 Overview

### 2.1 Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography is a public key cryptography based on the algebraic structure of elliptic curves over finite fields that exploits the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP) for security. On a binary field  $\mathbf{F}_2^m$ , ECC uses the relation  $y^2 + xy = x^3 + ax^2 + b$ , where  $b \neq 0$ .

Most protocols for WSNs that involve ECC use the prime field  $\mathbf{F}_p$  because micro-controllers do not sufficiently support arithmetic operations over  $\mathbf{F}_2^m$  [4]. ECC, on a prime field  $\mathbf{F}_p$ , uses the elliptic curve defined as  $y^2 \bmod p = x^3 + ax + b \bmod p$  where  $4a^3 + 27b^2 \bmod p \neq 0$  and  $p$  is a prime number chosen to allow for a large enough number of points on the curve to maintain security. The size of  $p$  in bits determines the security level of the cryptosystem [3]. In both of the above equations for binary and prime fields,  $a$  and  $b$  are the parameters of the curve and changing the values of either or both will result in a different curve. The point  $P$  is the generator point on the predetermined curve. The private key is a random number  $r$ . The public key,  $Q$ , is generated by  $rP$ . Both the parameters and points  $P$  and  $Q$  are public. Due to the difficulty of the ECDLP, even with points  $P$  and  $Q$  made public, it is virtually impossible to find  $r$  when  $r$  is sufficiently large [6].

A hybrid cryptosystem that utilizes public key cryptography as well as symmetric key cryptography should choose key sizes of same or similar strength for all of its cryptographic components. This is due to the fact that the weakest component determines the overall strength of the cryptosystem. As recommended by NIST, Table 1 shows key sizes needed to consistently maintain a certain level of security in the uses of symmetric key, RSA-based, and ECC-based cryptographic components in the same cryptosystem. As shown in the table, a particular key size is no longer safe to use to secure sensitive data, once its recommended lifetime is over. Compared to Integer Factorization Cryptography (IFC) such as RSA, ECC requires considerably a fewer number of bits for the same level of security. As shown in the table, the key size of 224 bits for ECDH can guarantee equivalent security that can achieved using RSA key size of 2048 bits. Compared to IFC, ECC has been identified as more energy-efficient and suitable for WSNs, since it involves computation with relatively smaller numbers.

Table 1: Lifetime and Minimum Key Sizes Recommended for IFC and ECC Algorithms [5]

Symmetric Key Cryptography	Integer Factorization Cryptography (e.g., RSA)	Elliptic Curve Cryptography (e.g., ECDH)	Key Usage Life
80 bits	1024 bits	160 bits	Through 2010
112 bits	2048 bits	224 bits	Through 2030
128 bits	3072 bits	256 bits	Beyond 2030

## 2.2 Elliptic Curve Diffie-Hellman (ECDH) Protocol

The Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol implements ECC and has been determined as a suitable method for WSNs in which energy efficiency is a concern [7]. The ECDH protocol allows two parties, Alice and Bob, to securely share a secret key over an insecure channel. Prior to the exchange, the parties predetermine the specific elliptic curve,  $E$ , and the domain parameters used to generate a set of public/private keys. The domain parameters  $(a,b)$  specify the curve on the finite field  $\mathbf{F}_p$ . The base point  $P$  is located on  $E(\mathbf{F}_p)$  and is also agreed upon before any encryption or sharing is done. By using the domain parameters and  $P$ , a cyclic subgroup of order  $n$  is generated. To exchange the secret, Alice generates the random number  $r_A$  which is within the interval  $(1, n-1)$ . She then combines  $r_A$  and  $P$  using scalar multiplication to create the public key  $Q_A$  where  $Q_A = r_A \cdot P$ . Alice then sends  $Q_A$  to Bob. Bob does the same operation with  $Q_B = r_B \cdot P$  and sends  $Q_B$  to Alice. In order for Alice to obtain the secret  $S$ , once she receives  $Q_B$ , she uses scalar multiplication to compute  $S$  because  $S = r_A \cdot Q_B$ . Bob determines  $S$  in the same manner using  $S = r_B \cdot Q_A$ . Both parties will have the same  $S$  because  $E(\mathbf{F}_p)$  is a communicative group and  $r_A \cdot Q_B = r_A \cdot r_B \cdot P$  and  $r_B \cdot Q_A = r_B \cdot r_A \cdot P$ . In this protocol, even if an outside attacker were to capture  $Q_A$  and  $Q_B$ , due to reason explained in the overview of ECC, privacy of  $S$  is still maintained.

Rapid computation and small key sizes and signatures make ECC a popular selection in WSNs [4]. As indicated in Table 1, the equivalent level of security of 1024-bit RSA-based schemes can be achieved by using only 160-bit ECC-based schemes. The smaller key size makes ECC, and by association the ECDH protocol, more energy efficient than RSA for communication purposes as well.

Having prestored keys in sensor nodes, as suggested in many recent works, can be problematic if nodes are in danger of being captured by an adversary. The ECDH protocol can be effective against such node capture as the ECDH protocol allows dynamic key re-establishment. Another reason for popularity of the ECDH protocol is its high scalability, which accommodates large WSNs and allows the size of a network to be changed easily

## 2.3 TinyECC

For our work, we use the TinyECC implementation to run the ECDH protocol on an Imote2 sensor node. TinyECC is a library created for sensor platforms that runs in TinyOS environment. It is implemented in nesC, which is based on C. TinyECC supports the ECDH key exchange protocol, Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Integrated Encryption Scheme (ECIES) [4]. Imote2, TelosB, Tmote Sky, and MICAz motes all successfully use the TinyECC library making it very portable. TinyECC also

has several optimization switches, which make it highly configurable. By using optimization switches, prime fields for ECC operations, and inline assembly code, TinyECC can significantly decrease computation time resulting in reduced energy consumption [4]. Because scalar multiplication is the most expensive operation when computing ECC keys, many of the optimizations supported by TinyECC address this issue. Below is a brief description of some of the optimization options available in TinyECC:

- Fast Modular Reduction is accomplished by using Barrett Reduction method, which achieves results more quickly than simple division. However, it requires more ROM and higher RAM usage [8].
- Fast Modular Inversion can be achieved by using projective coordinate representation as opposed to the affine coordinate system, which requires very expensive modular inversion operations. The projective system in TinyECC replaces the inversion operations with modular multiplication and squares. This results in quicker computation of critical point addition and point doubling operations, which are the basis for scalar multiplication in ECC [4, 8].
- Hybrid Multiplication is used in TinyECC by the implementation of a hybrid multiplication algorithm proposed by Gura et al [9]. In high level languages similar to nesC, the implementation of large integer multiplication algorithms causes the binary code of the microprocessor to frequently load the operands from the memory into the registers [9]. This algorithm decreases the amount of memory operations and optimizes the use of registers [4].
- Hybrid Squaring uses a modified version of the above hybrid multiplication algorithm and makes computation faster but increases code size.
- Curve-Specific Optimization uses elliptic curves over a pseudo-Mersenne prime field. The curves used have been specified by NIST and SECG [10, 11]. This allows for the reduction modulo to be done by a few modular multiplications and additions while avoiding division. The result is faster computation and higher performance.
- The Sliding Window Method directly addresses scalar multiplication by scanning a set number of bits  $w$  at a time instead of scanning each bit individually from least to most significant. Every time a group of  $w$  bits is scanned, the algorithm must then perform  $w$  point doublings. The Sliding Window Method pre-computes the points within the group of size  $w$ . Thus, point addition only needs to be done only once for every  $w$  bits. However, the method requires additional memory due to increase in code size.

- Shamir's Trick reduces the cost of two scalar multiplications to be near the equivalent of one. However, it increases both ROM and RAM requirement due to increase in code size.

All of the above optimization options may be turned on or off depending on resource constraints such as memory or processing capacity of a sensor node. Besides configurable optimization options, TinyECC incorporates all 128-bit, 160-bit and 192-bit ECC parameters suggested by the Standards for Efficient Cryptography Group (SECG) that allows application-specific customization of TinyECC [4,11].

## 2.4 Imote2

For our work on measuring power consumption of the ECDH protocol, we use Imote2 sensor nodes. As stated above, the Imote2 platform is suitable for the complete TinyECC library. According to the MEMSIC specification datasheet for the Imote2, "*The Imote2 is aimed at applications involving data-rich computations, where there is a need for both high performance and high bandwidth, which require greater processing capability and low-power operation with a low duty cycle to achieve longer battery-life*" [12]. The Imote2 platform supports the ITS400 sensor board for collecting data on acceleration, humidity, temperature, and light; the IMB400 sensor board for capturing images, video, and audio; and the IMS400 sensor board for collecting vibration health data and structural health monitoring. As our main concern is power consumption, accordingly in Table 1, we provide some specification details of the Imote2 platform for power consumptions at various operational modes [13].

Table 2: Imote2 Specifications

<b>PROCESSOR BOARD IPR2400</b>	
CPU	Marvell PXA271 XScale Processor
Memory	256 KB SRAM, 32 MB SDRAM, 32 MB FLASH
<b>POWER</b>	
Battery	3xAAA
Battery Voltage	3.2 – 4.5 V
Current Draw in Deep Sleep Mode	387 $\mu$ A
Current Draw in Active Mode (13 MHz, radio off)	31 mA
Current Draw in Active Mode (13 MHz, radio Tx/Rx)	44 mA
Current Draw in Active Mode (104 MHz, radio Tx/Rx)	66 mA

### 3 Methodology

**Optimization Options:** For our tests, we use the curves defined by the verifiably random SECG parameters over the field  $\mathbf{F}_p$  and accordingly build our test program on the TinyECC foundation. We select these standard elliptic curves because of their widespread acceptance and use for elliptic curve cryptography. Particularly we conduct our tests for key sizes of 128 bits, 160 bits, and 192 bits using SECG parameters for elliptic curves over the field  $\mathbf{F}_p$ . For the sake of comparative analysis, we study three test cases: 1) We test all the three key sizes for power consumption with no optimization options for ECC operations in order to provide a base case for each key size; 2) Next we study the projective coordinate optimization as a separate case because of its fast modular inversion; and 3) Finally for our “all optimization” case, we select the case of the projective coordinate optimization again but in conjunction with other optimization options and operations such as hybrid multiplication, hybrid squaring, and SECG curve specific optimization options. The hybrid multiplication and squaring optimizations are selected for their efficient use of registers that reduces the number of needless memory movement operations. We do not use the Barret Reduction and Sliding Window optimization methods because of the large memory overhead involved with both of them, even though the Imote2 sensor node has enough memory and processing power to handle them. When using ECC on sensor nodes in more than just a test fashion, the consumption of resources must be kept to a minimum in order to facilitate their primary node functions for the intended application.

**Measurement of Completion Time:** The calculation of the completion time for each case of optimization option for a key size is done directly in the code using the standard system get time functions available in the C language. We view the results of these calculations using the “trace debug messages” option available on the Imote2 platform that outputs the messages on a USB port console. The time value is displayed in terms of number of clock cycles. Hence we divide this number by the clock speed to get the runtime in seconds [14]. For each of our time measurement, the average of five trials is computed.

**Measurement of Power Consumption:** We calculate power consumption for each test case by measuring the battery voltage and the current drawn by the Imote2 setup. To measure the current, we power the Imote2 using the battery board and use a standard ammeter in series with the batteries to complete the electrical circuit. A standard voltmeter is used in parallel with the battery board to measure the voltage. The LEDs we use for debugging purposes in our program appear to cause a slight fluctuation in power consumption. As LEDs do not have anything associated with ECC, we record only the minimum in power consumption in order to nullify or minimize the effect of power consumption by LEDs on the overall power consumption for ECC. For each test case, we

measure power consumption three times in this way and record only the median of the three measurements whenever there is any variation among them.

Table 3. Execution Time (in seconds)

	SECP128R1	SECP160R1	SECP192R1
No Optimization	25.536	41.720	66.118
Projective Coordinate Optimization	12.691	13.345	19.677
All Optimization*	10.821	7.218	10.640

Table 4. Current (in mA)

	SECP128R1	SECP160R1	SECP192R1
No Optimization	57.5000	58.3250	58.3250
Projective Coordinate Optimization	58.0000	58.7500	58.7500
All Optimization*	59.3750	59.3750	59.6875

Table 5. Total Energy Consumption (in Joules)

	SECP128R1	SECP160R1	SECP192R1
No Optimization	6.607	10.950	17.353
Projective Coordinate Optimization	3.312	3.528	5.202
All Optimization*	2.891	1.929	2.858

\*All optimization includes hybrid multiplication, hybrid squaring, projective coordinate, and SECG curve optimizations

Tables 3, 4, and 5 show the results of our measurements on execution time for ECC, current drawn by the Imote2 setup, and corresponding energy consumption for ECC. All the

results are obtained by running all test programs 30 times with the processor clock set to 104 MHz.

In each table, a label SECPXXXXR1, the XXX is the key size and the 'P' indicates that the prime number field  $F_p$  is used. The 'R' in the key name indicates that these are verifiably random keys as specified by SECG [11].

## 4 Conclusions

As our results show, smaller SECG parameters with less number of bits in size do not necessarily mean less overall power consumption. While the execution time does steadily decrease with smaller key sizes when no optimization is implemented, however, the optimizations seem to heavily favor the larger key sizes. The 160 bit ECDH computations as well as the 192 bit ECDH computations show major reductions when all optimizations are included while the 128 bit ECDH computations show minor reductions. With the current optimizations available to elliptic curve cryptography on small mobile platforms like the Imote2, it appears that there is no reason to sacrifice security by using less than a 160 bit key for ECDH. As our tests show, the ECDH computations using a 128 bit key are actually less energy efficient when all optimizations are applied, and a key any smaller would leave the wireless sensor network too vulnerable. Even though the NIST standards are beginning to recommend 192 bit ECC keys for high level security, the 160 bit keys may still be adequately secure for a while and offer power savings that are not insignificant on these small, battery-powered sensor nodes. For the vast majority of applications, which do not require any top secret level security, 160 bit ECC keys seem to offer a balance of security and power savings in WSNs.

Future works may include efficiency testing of a multitude of other key sizes as well as on other types of wireless sensor platforms. Future tests could also utilize different combinations of optimizations to determine which key size each optimization favors. RAM usage could also be included and compared to total power consumption and execution time to find a better balance between security and resource consumption. Realistically, there is little use in testing key sizes larger than 256 bit for a while, but including RAM usage into the comparison seems like the next logical step.

## 5 Acknowledgements

We would like to thank National Science Foundation for funding this work and making all of this possible through a grant (CNS #1004902). We would also like to thank Clifton Mulkey and Jon Gonzalez of Texas A&M University-Corpus Christi for their technical assistance and An Liu of North Carolina State University for providing technical insights of the TinyECC library.

## 6 References

- [1] J. Grobschadl, A. Szekely, and S. Tillich, "The Energy Cost of Cryptographic Key Establishment in Wireless Sensor Networks," Second ACM Symposium on Information, Computer and Communications Security, 20-22 March 2007.
- [2] "D4.4 Power Estimation Methodology for Secure Algorithms and Protocols in Embedded Networks," Information Society Technologies (IST) Programme, Secure Middleware for Embedded Peer-to-Peer Systems, 12 September 2008.
- [3] Anoop MS, "Elliptic Curve Cryptography: An Implementation Tutorial," Tata Elxsi Ltd, 2007, Available at: [http://www.infosecwriters.com/text\\_resources/pdf/Elliptic\\_Curve\\_AnnopMS.pdf](http://www.infosecwriters.com/text_resources/pdf/Elliptic_Curve_AnnopMS.pdf), Last accessed: 03 August 2010.
- [4] A. Liu and P. Ning, "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks," Seventh International Conference on Information Processing in Sensor Networks, pp. 245-256, April 2008.
- [5] E. Barker, et al., "Recommendation for Key Management – Part 1: General Guidelines," NIST Special publication 800-57, National Institute of Standards and Technology, March 2007.
- [6] W. Wang, Y. Lin, and T. Chen, "The study and application of elliptic curve cryptography library on wireless sensor network," Eleventh IEEE International Conference on Communication Technology, pp.785-788, 10-12 November 2008.
- [7] C. Lederer, R. Mader, M. Koschuch, J. Grobschadl, A. Szekely, and S. Tillich, "Energy-Efficient Implementation of ECDH Key Exchange for Wireless Sensor Networks", 3rd Workshop on Information Security Theory and Practice, Springer Verlag, pp.112-127, 2009.
- [8] D. C. Kar, H. L. Ngo, and G. Sanapala, "Applied Cryptography for Security and Privacy in Wireless Sensor Networks," International Journal of Information Security and Privacy, Vol. 3, Issue 3, 2009.
- [9] N. Gura, A. Patel, A. Wander, H. Eberle, and S.C. Shantz, "Comparing Elliptic Curve Cryptography and RSA on 8-Bit CPU's," Cryptographic Hardware and Embedded Systems (CHES), Springer Berlin/Heidelberg, 2004.
- [10] "Recommended Elliptic Curves for Federal Government

Use,” National Institute of Standards and Technology, July 1999, Available at: [csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf](http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf), Last Accessed: 18 May 2011.

- [11] “SEC 2: Recommended Elliptic Curve Domain Parameters,” Standards for Efficient Cryptography, Certicom Corporation, Version 1.0, 20 September 2000.
- [12] “Imote2 Hardware Bundle for Wireless Sensor Networks,” Memsic Corporation, Available at: <http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=139%3Aimote2-multimedia>, Last accessed: 18 May 2011.
- [13] “Imote2 High-Performance Wireless Sensor Network Node,” Memsic Corporation, Available at: <http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=134%3Aimote2>, Last accessed: 18 May 2011.
- [14] J. Rice and B.F. Spencer, “Structural Health Monitoring Sensor Development for the Imote2 Platform,” Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, SPIE, 2008.

# Symmetric key Cryptography using modified DJSSA symmetric key algorithm

**Dripto Chatterjee<sup>1</sup>, Joyshree Nath<sup>2</sup>, Sankar Das<sup>3</sup>, Shalabh Agarwal<sup>4</sup> and Asoke Nath<sup>5</sup>**

<sup>1,3,4,5</sup>Department of Computer Science, St. Xavier's College(Autonomous),Kolkata, India

<sup>2</sup>A.K.Chaudhuri School of I.T, Raja Bazar Science College, Kolkata,India

**e-mail:dripto18@gmail.com<sup>1</sup>, joyshreenath@gmail.com<sup>2</sup>, dassankar16@yahoo.co.in<sup>3</sup>, shalabh@sxccal.edu<sup>4</sup>, asokejoy@gmail.com<sup>5</sup>**

**Abstract** -The present work deals with modified advanced symmetric key cryptographic method i.e. modified DJSSA algorithm[1] for multiple encryption and decryption of any file. Recently Nath et al[1] developed an algorithm called DJSSA for encryption and decryption of any file using a random key matrix 256x65536 containing all possible 3 lettered words. Nath et.al [2] also proposed a method using a randomized key of size 256x256 containing all possible 2 lettered words. The above method was an extension of MSA algorithm proposed by Nath et al[3]. In the present work the authors have extended the DJSSA algorithm one step further. The present work proposes a key matrix of size 65536x256 which contains all possible 3-lettered words. The present method uses a simple randomization technique[1] to make this key matrix random. So the complexity of finding the actual key matrix will be 16777216! trial runs and which is intractable. In the current modified DJSSA method the authors added one additional module to perform bit interchange between two consecutive bytes. This bit interchange will take place before DJSSA method. The bit interchange will continue during each encryption to make the encryption system more secured. The authors claim that it may be taken as the ultimate symmetric key method which can not be broken by using any brute force method. This method will be suitable in any business house, government sectors, communication network; defense network system provided the file size is small. The user has to enter some secret text-key. The maximum length of the text key should be 16 characters long. To calculate the randomization number and the number of encryption to be done is calculated from the text-key using a method proposed by Nath et.al(3). The present method will be most suitable for encryption of a small file such as digital signature or watermarking etc. The present method can be used to encrypt a large file by splitting the entire file into

reasonable numbers and then run encryption program parallel in different machines and after that one has to append those encrypted file to get the ultimate encrypted file. To decrypt the file one has to follow the same trick i.e. first split into same number of files and then apply the decryption algorithm in parallel from different machines and finally append all decrypted files to get back the original file.

## 1. Introduction

Due to enormous development in internet technology the security of data has now become a very important challenge in data communication network. One can not send any confidential data in raw form from one machine to another machine as any hacker can intercept the confidential message. This will be more prominent when someone is sending some confidential matter over the mail such as question paper or Bank statement or any other confidential matter. There is no guarantee that the message will not be intercepted by anyone. This may be further worse during e-banking or e-commerce where the real data should not be intercepted by any hacker. When a client is sending some confidential matter from client machine to another client machine or from client machine to server then that data should not be intercepted by someone. The data must be protected from any unwanted intruder otherwise any massive disaster may happen at any time. The disaster may happen if we send plain text or clear text from one computer to another computer. To get rid of this problem one has to send the encrypted text or cipher text from client to server or to another client. Due to this problem network security and cryptography is now an emerging research area where the people are trying to develop some good encryption algorithm so that no intruder can intercept the encrypted message. The classical cryptographic

algorithm can be classified into two categories: (i) symmetric key cryptography where the single key is used for encryption and for decryption purpose. (ii) public key cryptography where two different keys are used one for encryption and the other for decryption purpose. Depending on the problem sometimes symmetric key algorithms are applied and sometimes public key cryptography algorithm is applied. The merit of symmetric key cryptography is that the key management is very simple as one key is used for both encryptions as well as for decryption purpose. In case of symmetric key cryptography the key should be maintained as secret key. In public key cryptography the encryption key remains as public but the decryption key should be kept as secret key. The public key methods have got both merits as well as demerits. The problem of public key cryptosystem is that one has to do massive computation for encrypting any plain text. Moreover in some public key cryptography the size of encrypted message may increase. Due to massive computation the public key crypto system may not be suitable in security of data in sensor networks. So the security problem in sensor node is a real problem. However, there is quite a number of encryption methods have came up in the recent past appropriate for the sensor nodes. In the present work we are proposing a symmetric key method which is an updated DJSSA algorithm(1) where we have used a random key generator for generating the initial key and that key is used for encrypting the given source file. In the present method the authors have added one module with DJSSA(1) algorithm. Before we apply DJSSA method we read 6 characters from the original file and then change the bit pattern as shown below:

We interchange the bits as follows: (bit-8 of byte-1,bit-7 of byte-2), (bit-7 of byte-1, bit-8 of byte-2), (bit-6 of byte-1, bit-5 of byte-2), (bit-4 of byte-1, bit-3 of byte-2), (bit-3 of byte-1, bit-4 of byte-2), (bit-2 of byte-1, bit-1 of byte-2), (bit-1 of byte-1, bit-2 of byte-2). It shows the original 6 characters will be encrypted or rather modified after bit interchange. After this bit interchange divide the 6 characters into 2 blocks each containing 3 characters and then search the corresponding blocks in the random key matrix file to get the corresponding encrypted pattern and then we write the encrypted message in another file. For searching characters from the random key matrix we have used MSA method which was proposed by Nath et.al(2). In the present work there is a provision for encrypting message multiple times. Before we apply actual encryption method each time we first interchange the bits as shown in table-1 to make the entire process more secured. The key matrix contains all possible words comprising of 3 characters each generated from all characters whose ASCII code is from 0 to 255 in a random order. The pattern of the key matrix will depend on text\_key entered by the user. To make the key matrix random we have used our own randomization algorithm which we generate from initial text\_key. Nath et.al(2) proposed a simple algorithm to obtain the randomization number and encryption number from the text\_key. To decrypt any file one has to know exactly what is the key matrix and to find the random matrix theoretically one has to apply 16777216! trial runs and it is almost intractable. We apply our method on possible files such as executable file, Microsoft word file, excel file, access database, foxpro file, text file, image file, pdf file, video file, audio file, oracle database and we found in all cases it giving 100% correct solution while encrypting a file and decrypting a file. The present method can be used for encrypting digital signature, watermark before embedding in some cover file to make the entire system full secured.

TABLE-1: INTERCHANGING BIT PATTERNS

1 <sup>st</sup> character	bit-8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1
2 <sup>nd</sup> character	bit-8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1
3 <sup>rd</sup> character	bit-8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1
4-th character	bit-8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1
5-th character	bit-8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1
6-th character	bit-8	bit-7	bit-6	bit-5	bit-4	bit-3	bit-2	bit-1

## 2. Generation of 65536X256 Random Key Matrix:

To create Random key matrix of size (65536x256) we have to enter a text-key which is a secret key. The size of text-key must be less than or equal to 16 characters long. These 16 characters can be any of the 256 characters (ASCII code 0 to 255). From the text-key we calculate (i) randomization number and (ii) encryption number using some simple algorithm which we will be showing below. This method is very much sensitive on the relative position of each character. Now we will show how we can calculate (i) Randomization number and (ii) encryption number from a given text-key :

We choose the following table for calculating the place value and the power of characters of the incoming key:

TABLE-2: LENGTH OF TEXT-KEY AND THE CORRESPONDING VALUE OF BASE

Length of key(n)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Base value(b)	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2

Suppose key entered by the user is "AB". The length of the text-key is 2. To calculate the randomization number and encryption number we follow the following steps:

$$m= n$$

$$\text{Step-1: Take Sum} = \sum (\text{ASCII Code}) * b^m$$

$$m=1 \quad \text{-----(1)}$$

So if our text-key is "AB" then from equation (1) we get  
Sum=65\*161 + 66 \* 162 =17936

Now we have to calculate 2 parameters from this sum (i) Randomization number(n1) and (ii) Encryption number(n2) using the following method:

### 2.1. To calculate Randomization number(n1):

Calculate sum of product of each digit in Sum and its place value in that sum as follows:

$$\text{num1} = 1*1 + 7*2 + 9*3 + 3*4 + 6*5 = 84$$

$$\text{Now } n1 = \text{Mod}(\text{sum}, \text{num1}) = \text{Mod}(17936, 84) = 44$$

Note: if n1=0 then we set n1=num1 and if n1>8 then n1=n1/8

### 2.2. To calculate Encryption number(n2):

Calculate the product of each digit in the Sum by its position in the sum in reverse order as follows:

$$\text{num2} = 6*1 + 3*2 + 9*3 + 7*4 + 1*5 = 72$$

$$\text{Now calculate } n2 = \text{Mod}(\text{sum}, \text{num2}) = \text{Mod}(17936, 72) = 8$$

Note: if n2=0 then we set n2=num2 and if n2>15 then we set n2=n2/15

Now we explain how we have made the random key of size 65536x256 which is used for encryption as well as for decryption purpose. We create the random key file in a step by step manner as follows. It is difficult to store 50331648 (=65536x256x3 characters) elements in some array hence we store the entire key in a file. We divide the entire key into 4096 blocks where each block contains 64x64 words and each word contains 3 characters. We create each block separately in computer in random order and then we apply randomization methods one by one on

it and then write on to an external file again in some random order. The basic idea of randomization process is to make the key

matrix totally random so that no one can guess it in advance.

Now we show the original key matrix(256x256x3x256) which contains 1024x4 blocks each of size 64x64x3 characters:

TABLE -2: THE ORIGINAL KEY MATRIX:

Block-1(64X64X3)	Block-2(64X64X3)	→	Block-1024(64X64X3)
Block-1025(64X64X3)	Block-1026(64X64X3)	→	Block-2048(64X64X3)
Block-2049(64X64X3)	Block-2050(64X64X3)	→.	Block-3072(64X64X3)
Block-3073(64X64X3)	Block-3074(64X64X3)	→	Block-4096(64X64X3)

We generate each block in a 3-dimensional array of size(64x64x3) where we store 3 lettered words starting from a word 000 to final word 255255255 in some random order. The words in each block we generate in computer internal memory and then apply 5 randomization methods one after another in a random order and then write onto key file again in random order.

The following randomization process we apply serially on each block internally. We apply the below methods in random to make the elements in each block as random as possible.

The following are the operations we execute serially one after another.

TABLE-3: RANDOMIZATION STEPS

Step-1: Function cycling()
Step-2: Function upshift()
Step-3: Function rightshift()
Step-4: Function downshift()
Step-5: Function leftshift()
Step-6: Repeat Function downshift()
Step-7: Repeat Function rightshift()
Step-8: Repeat Function upshift()
Step-9: Repeat Function cycling()

Now we describe the meaning of 5 above functions(Step-1 to step-5) when we apply on a 4x4 matrix as shown below:

TABLE-4 : ORIGINAL TABLE

AAA	ABA	ACA	ADA
BAA	BBA	BCA	BDA
CAA	CBA	CCA	CDA
DAA	DBA	DCA	DDA



TABLE-5: CALLED CYCLING()

BAA	AAA	ABA	ACA
CAA	BCA	CCA	ADA
DAA	BBA	CBA	BDA
DBA	DCA	DDA	CDA



TABLE-6: CALLED UPSHIFT()

DAA	BAA	BBA	AAA
DDA	CCA	CDA	ADA
DBA	CAA	DCA	BCA
ACA	CBA	ABA	BDA



TABLE-7: CALLED RIGHTSHIFT()

BDA	DAA	DBA	BAA
CAA	BBA	DCA	AAA
BCA	DDA	ACA	CCA
CBA	CDA	ABA	ADA



TABLE-8: CALLED DOWNSHIFT()

ADA	DCA	ABA	AAA
BDA	BCA	DAA	DDA
DBA	ACA	BAA	CCA
CAA	CBA	BBA	CDA



TABLE-9: CALLED LEFTSHIFT()

ABA	CCA	AAA	CAA
ADA	ACA	DCA	BAA
DAA	CDA	DDA	DBA
BDA	CBA	BCA	BBA

The above randomization process we apply for n1 times and in each time we change the sequence of operations to make the system more random. Once the randomization is complete we write one complete block in the output key file. Now we show how we apply encryption process on a particular file. For this we choose our last randomized 4x4 matrix(table-9). We apply the following encryption methods:

Case-I : Suppose we want to encrypt DAA and CDA which appears on the same row. Then the encrypted message will be DDA and DBA.

Case -II : Suppose we want to encrypt CCA and ACA where CCA and ACA appears on the same column in the key matrix. The encrypted message will be CDA and CBA.

Case-III: Suppose we want to encrypt ACA and DDA which appears in two different rows and different columns then the encrypted message will be ADA and DBA.

The decryption process will be just the opposite path of encryption process. Our method supports both multiple encryption and multiple decryption process.

### 3. Results and Discussion:

Here we are giving result which we obtain after we apply encryption method on a text file and also the decryption method on the decrypted file to get back original file.

- (i) Text-key used=12
- (ii) Randomization number created by our method : 1
- (iii) Encryption number generated by our method : 2

The above values were used for encryption and decryption on a given text file(jesuits.txt):

**Original File Name: Jesuits.txt**  
 JESUITS AND EDUCATION IN INDIA

The Society of Jesus, a Christian Religious Order founded by Saint Ignasius of Loyola in 1540, has been active in the field of education throughout the world since its origin. In the world, the Society of Jesus is responsible for over 1865 Educational Institutions in over 65 countries. These Jesuit Educational Institutions engage the efforts of approximately 98,000 teachers. They educate approximately 17,92,000 students. In India, the Society of Jesus is responsible for 150 High Schools, 38 University Colleges, 14 Technical Institutes with 8095 teachers educating 2,34,338 students, belonging to every social class and linguistic group. These Institutions are part of the Catholic Church's effort to share in the country's educational undertaking.

The Jesuit College aims at the integral, personal formation of youth. To accomplish this, special efforts are made:

- To help students to become mature, spiritually-oriented men and women of character;
- To encourage them continuously to strive after excellence in every field;
- To value and judiciously use their freedom;
- To be clear and firm on principles and courageous in action;
- To be unselfish in the service of their fellowmen; and
- To become agents of needed social change in their country.

The Jesuit College aims at making its own contribution towards a transformation of the present-day social condition so that principles of social justice, equality of opportunity, genuine freedom, and respect for religious and moral values, enshrined in the Constitution of India, may prevail, and the possibility of living as fully human existence may be open before all.

**Size of the original file and Decrypted file:2KB**

**Encrypted file(output.txt):**

**Size of encrypted file:2KB**

**Encrypted File Name: out.txt(size=2KB)**

```

à A( à UÅX%VN A „CÁW- nU
[] Mi%eMDðF%eCé| • -L kotTlt:k| 8"Ja"%%X
óU
"iüt*³v áll'giŠm| -wrl Lr ðg A(Rà o¹ UOİbll c
Ycn a "mêp?hrAx ©mlésÿ XdâeDâcafw
§g ÈyO-¹ nll □ ðgy
h àSt¹¹ àKsY'ô¶| ©wfn¹ àLlùd¹
"lî"Xð0hm,Xll p b( à
nYhNcboðk@0eŠkhYmt,hT©un a
àtfâ"©goâ"¹!"c¹t- jv r hI'jgÿs¹ il ô+Uà •A(µUl
ân
•efâo¹ ©xn a- àtsâ"¹- îrecc!! ?hhaYke o| -neµo©
Ot-
T3drljRYknºoNDosévS?hiŠm- ©- ěf
jjo ~²nt),¹ 3d ;% à A(
cbäÿ dTlt:k| 8"Ja"%%xió+ W*p...y¹ - mlèa$"
ýtFlw11!Tø6 « ¹ à A( à •bi"6Fh
ÆfO!isOhr ?he n¹ Yfsð#r - h
,oİñle>¹¹ àkhal- Yk x...ft"l r jl
,oN%omtôj• Gfn{d- W*oi"- ¶lg5 ?Gàctôl- qj
Î.Eà A( à nŵ
'w
l
tÊTee}àJiô~v!côu&jva r h□ ¼&tþu=èviév - mn
m §ctâ"!! 3d ;% à
ègtŠgtôk • àCeigšgmákMW*mô"ll h \alèv
w f¹ ¾&tÿh| qj%ot- meF¹ E.c
†~OlgpI±haŠj¹ ¹g98#Uø, ±8† ¹geâj¹ aphó_FY{
A( à A(
¼hdi}!! W*r±x!! jmræl r NknÉe- ?hhal"sòg¹ Yma
n¹ 8"iîa- jneİ E³weñr- cbosz-
,keñr□ ¼¶ ,j3830d< àseµd¼ q ç'Sà A( à A(
dlt-a- jloAz r àt "vHºqîî rmo1!!"iîa- jnsúsS¹k
Å&C-¹ I., ça à-I'jôAZ£k ô{Elgs)nŠµp éx
sgnðm "c ücB(l A(Rà uŠtUjñ
ä¶ llgp...qP©qiôm ¹{ePBI©w
igI¶uo=c-.d'IDİreÛs

```

```

év5C28"„heàSoèm-x8A:¶ àUelz
ap '~Ifdgin
Yx b(à A( à
fA(F@qlâbL)isMs- cbmä"¹ @qv@#| jn,...yYàer
ó j ©mig,¾& øcT-¹ T7!eYhcib
jnsÂİS keöv◀ W*hev9 "1+ ¹geâj¹ ap s( à A( à
nÂdN³vi.p 'koî" "iif j,yŽkYàpaG~!! ©nn aB²"
ègT-¹ cg¶ - jv îeIÁ 3¶( 8=uürUðg,
İBİbnÂm-ot÷&- "rög¹ 8" A( à sðdS"cién
¹{ id ¼ogùh◀ cbu f- -nu hNkx e r ©xcâe¹ $"
á{MCos-
S¹ h ,oİfn òqA;mi l'iti} r ©hoçqOYZA(à A( à
A( à A( -rbj&• Yē òfO;ñ©
f¹j "İE%omtôj• Gfn{d- W* ògAºcoò" "İ
ègT,fclzj¹ ©hC-
c3t¹ ÈbàW*oæd²v ôm hm
)ba©meðkEactôl- qzİYêmað¹ | ¹k-
¹¹ àurYg vbg+oİeÄnoàJiô~v!!
e/ºiicg-x Áw jñYh
¹g¹'kieŠphapx¹m¹ àfa!j'koî" "i'iw'- 3e
ll ¹piif!! ¼otø&• "k-
~Èsi...i¹ jnfâdF@qaø#¹ sgdŽkD¹ 1Té•Y- "l;İLP"
d-p¹ Ymtó#- "o...i
-gtŽkTus
□ ,hÊ rðk r èvlán¹ ¼eÒçqm--
□ YmdámDâwníf r ?hh cppeñw¹ 4<TŠTM Y- "oíau
s fA¹jc &- -muÈe r -w i{S¹4hrü{R©ufçm¹ ¹g
xYzIÁFÿ Ymi$¾&ræc¹ 8"l;h²OTé•Y- "uan¹ $"
îfA»piÚf
¹w i{Šµpháp
Ykr o äg;io-¹ | % )- ll •

```

The present method can be used to encrypt password, question paper, bank transaction report, small database, in government office, railways etc. This method may be further extended by introducing bit exchange and DJSSA algorithm in serial order to make the entire system hard to decrypt by any intruder.

**4. Conclusion**

In the present work we use the maximum encryption number=15 and maximum randomization number=8. We have used the key matrix of size 65536x256x3 . This key may be generated in 16777216! ways. So in principle it is not possible for any one to decrypt the encrypted text without knowing the exact key. In the present work we have used two stages of encryption one by exchanging bits and then by changing pattern according to random key matrix. Our method is essentially block cipher method and it will take more time if the files size is large and the encryption number is also large. The merit of this method is that it is almost impossible to break the encryption algorithm without knowing the exact key matrix and the bit exchange method. We propose that this encryption method can be applied for data encryption and decryption in banks, in defense for sending confidential data. This work may be further extended by introducing random bit exchange before we begin encryption method using DJSSA algorithm.

## 5. Acknowledgement

We are very much grateful to Department of Computer Science to give us opportunity to work on symmetric key Cryptography. One of the authors (AN) sincerely expresses his gratitude to Mr. Totan Karmakar of Dept. of Computer Science for giving constant inspiration to complete this work. AN is grateful to University Grants Commission for giving financial assistance through Minor Research Project. JN expresses her gratitude to A.K.School of IT for allowing to work in research project at St. Xavier's College.. DC, AN,SA,SD are also thankful to all 3<sup>rd</sup> year Computer Science Hons. Students (2010-2011 batch) for their encouragement to finish this work.

## References:

- [1] Advanced Symmetric key Cryptography using extended MSA method: DJSSA symmetric key algorithm, Dripto Chatterjee, Joyshree Nath, Soumitra Mondal, Suvadeep Dasgupta and Asoke Nath, Journal Computing, Vol.3, issue 2, Page 66-71, Feb 2011.
- [2] A new Symmetric key Cryptography Algorithm using extended MSA method :DJSA symmetric key algorithm, Dripto Chatterjee, Joyshree Nath, Suvadeep Dasgupta and Asoke Nath : accepted for publication in IEEE CSNT-2011 to be held at SMVDU(Jammu) 03-06 June 2011.
- [3] Symmetric key cryptography using random key generator, A.Nath, S.Ghosh, M.A.Mallik, Proceedings of International conference on SAM-2010 held at Las Vegas(USA) 12-15 July,2010, Vol-2,P-239-244.
- [4] Data Hiding and Retrieval, A.Nath, S.Das, A.Chakrabarti, Proceedings of IEEE International conference on Computer Intelligence and Computer Network held at Bhopal from 26-28 Nov, 2010.
- [5] Advanced Steganographic Approach for Hiding Encrypted Secret Message in LSB,LSB+1,LSB+2 and LSB+3 Bits in Non standard Cover Files, Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath, International
- [6] Cryptography and Network , William Stallings , Prectice Hall of India
- [7] Modified Version of Playfair Cipher using Linear Feedback Shift Register, P. Murali and Gandhidoss Senthilkumar, UCSNS International journal of Computer Science and Network Security, Vol-8 No.12, Dec 2008.
- [8] New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJSSAA symmetric key algorithm :Neeraj Khanna, Joel James, Joyshree Nath, Sayantan Chakraborty, Amlan Chakrabarti and Asoke Nath : accepted for publication in IEEE CSNT-2011 to be held at SMVDU (Jammu) 03-06 June 2011.
- [9] Advanced steganographic approach for hiding encrypted secret message in LSB, LSB+1, LSB+2 and LSB+3 bits in non standard cover files: Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath, International Journal of Computer Applications, Vol14-No.7,Page-31-35, Feb(2011).
- [10] Advanced Steganography Algorithm using encrypted secret message : Joyshree Nath and Asoke Nath, International Journal of Advanced Computer Science and Applications, Vol-2, No-3, Page-19-24, March(2011).
- [11] A Challenge in hiding encrypted message in LSB and LSB+1 bit positions in any cover files : executable files, Microsoft office files and database files, image files, audio files and video files : Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath, to be published in Journal of Global Research in Computer Science, Vol-2, May issue, 2011.
- [12] New Data Hiding Algorithm in MATLAB using Encrypted secret message : Agniswar Dutta, Abhirup Kumar Sen, Sankar Das, Shalabh Agarwal and Asoke Nath , accepted for publication in IEEE CSNT-2011 to be held at SMVDU(Jammu) 03-06 June 2011

# An efficient data hiding method using encrypted secret message obtained by MSA algorithm

Joyshree Nath<sup>1</sup>, Meheboob Alam Mallik<sup>2</sup>, Saima Ghosh<sup>3</sup> and Asoke Nath<sup>4</sup>

<sup>1</sup>A.K.Chaudhuri School of IT,Raja Bazar Science College,Calcutta University

<sup>2</sup>Department of Computer Science,Raja Bazar Science College,Calcutta University

<sup>3</sup>Cognizant Technology Services, Kolkata

<sup>4</sup>Department of Computer Science, St. Xavier's College(Autonomous),Kolkata

**e-mail:** <sup>1</sup> joyshreenath@gmail.com, <sup>2</sup> subho66@gmail.com,  
<sup>3</sup> saima.ghosh@gmail.com and <sup>4</sup> asokejoy@gmail.com

**Abstract** - In the present work the authors are proposing a new method for hiding any encrypted secret message inside a cover file by substituting in the 4-th bit position from LSB. For encrypting secret message the authors have used new algorithm called MSA proposed by Nath et al(1). For hiding secret message the method used was proposed by Nath et al(2). In MSA(1) method the authors have introduced a new randomization method for generating the randomized key matrix which contains all 256 characters in 16 X 16 matrix to encrypt plain text file and to decrypt cipher text file. To make the encryption and the decryption process totally secured the authors have introduced multiple encryption and multiple decryption methods. MSA method is totally dependent on the random text\_key which is to be supplied by the user. The maximum length of the text\_key can be of 16 characters long and it may contain any character (ASCII code 0 to 255). From the given text-key one can calculate the randomization number and the encryption number using an algorithm proposed by Nath et. al(1). The different key matrices can be formed is 256! and which is quite large and hence if someone applies the brute force method then he/she has to give trail for 256! times which is quite absurd. Moreover the multiple encryption method makes the system further secured. To hide encrypted secret message in the cover file the authors have inserted in the 4-th bit of each character of encrypted message file in 8 consecutive bytes of the cover file. The authors have introduced password for hiding data in the cover file. This method may be the most secured method in water marking, in defense for sending some confidential message.

## 1. Introduction

In the present work we have used two(2) distinct algorithms (i) to encrypt secret message(SM) using MSA(Meheboob,Saima and Asoke) proposed by Nath et al.(1). (ii) We insert the encrypted secret message inside the cover file(CF) by changing the 4-th bit from the least significant bit(LSB). Nath et al(2) already proposed different methods for embedding SM into CF but there the SF was inserted as it is in the CF and hence the security of steganography was not very high. In the present work we have basically tried to make the steganography method more secured. One can extract SM from CF but cannot be decrypted as one has to execute the exact decryption method. In our present work we try to embed almost any type of file inside some standard cover file(CF) such as image file(.JPEG or .BMP) or any image file inside another image file. Here first we will describe our steganography method for embedding any type of file inside any type of file and then we will describe the encryption method which we have used to encrypt the secret message and to decrypt the extracted data from the embedded cover file.

(i) 4-th Bit insertion method: Here we substitute the bits of the secret message in to 4-th bit position of every byte of the cover file. It was shown by a group of researchers(10) that this way the result which we get better than substitution in the LSB of the cover files. Although Nath et al(2) shown that it is not true in all cases. Now we choose some bit pattern where we want to embed some secret text:

```
11000100 00001100 11010010 10101101
00101101 00011100 11011100 10100110
```

Suppose we want to embed a number 224 in the above bit pattern. Now the binary representation of 224 is 11100000. To embed this information we need at least 8 bytes in cover file. We have taken 8 bytes in the cover file. Now we modify 4-th bit from LSB of each byte of the cover file by each of the bit of secret text 11100000. Now we want to show what happens to cover file text after we embed 11100000 in the 4-th bit from LSB of all 8 bytes:

TABLE 1 CHANGING 4-TH BIT FROM LSB

Before Replacement	After Replacement	Bit inserted	Remarks
00101101	0010 <b>1</b> 101	1	No change in bit pattern
00011100	0001 <b>1</b> 100	1	No change in bit pattern
11011100	1101 <b>1</b> 100	1	No change in bit pattern
10100110	1010 <b>0</b> 110	0	No change in bit pattern
11000100	1100 <b>0</b> 101	0	No change in bit pattern
00001100	0000 <b>0</b> 100	0	Change in bit pattern (1)
11010010	1101 <b>0</b> 010	0	No change in bit pattern
10101101	1010 <b>0</b> 100	0	Change in bit pattern(2)

Here we can see that out of 8 bytes only 2 bytes get changed only at the 4-th bit from LSB position. As human eye is not very sensitive so therefore after embedding a secret message in a cover file our eye may not be able to find the difference between the original message and the message after inserting some secret text or message on to it. To embed secret message we first skip 5000 bytes from the last byte of the cover file. After that according to size of the secret message (say n bytes) we skip 8\*n bytes. After that we start to insert the bits of the secret file into the cover file. The size of the cover file should be more than 10\* sizeof(secret message). For extracting embedded file from the cover file we have to perform the following:

We have to enter the password while embedding a secret message file. Once we get the file size we follow simply the reverse process of embedding a file in the cover file. We read 4-th bit from LSB of each byte and accumulate 8 bits to form a character and we immediately write that character on to a file.

We made an exhaustive experiment on different types of host files and also the secret messages and found the following combinations are most appropriate:

Table-II COVER FILE TYPE AND SECRET MESSAGE FILE TYPE

Sl. No.	Cover file type	Secret file type used
1.	.BMP	.BMP,.DOC,.TXT,.WAV,.MP3,.XLS,.PPT,.AVI,.JPG,.EXE,.COM
2.	.JPG	Any file type provided the size of the secret message file is very small in compare to cover file
3.	.DOC	Any small file
4.	.WAV	.BMP,.JPG,.TXT,.DOC
5.	.AVI	.TXT,.WAV,.JPEG
6.	.PDF	Any small file

After doing exhaustive study on all possible type of files we conclude that the .BMP file is the most appropriate cover file which can be used for embedding any type of file.

(ii) Meheboob, Saima and Asoke(MSA) Symmetric key Cryptographic method:

Symmetric key cryptography is well known concept in modern cryptography. The plus point of symmetric key cryptography is that we need one key to encrypt a plain text and the same key can be used to decrypt the cipher text and the minus point is that the same key is used for encryption as well as decryption process. Hence the key should not be public. Because of this problem the public key cryptography was introduced such as RSA public key method. RSA method has got both merits as well as demerits. The problem of Public key cryptosystem is that we have to do massive computation for encrypting any plain text. Some times these methods may not be suitable such as in sensor networks.. Nath et al.(1) proposed a symmetric key method where they have used a random key generator for generating the initial key and that key is used for encrypting the given source file. MSA method is basically a substitution method where we take 2 characters from any input file and then search the corresponding characters from the random key matrix and store the encrypted data in another file. In our work we have the provision for encrypting message multiple times. The key matrix contains all possible characters (ASCII code 0 to 255) in a random order. The pattern of the key matrix will depend on text\_key entered by the user. Nath et al.(1) proposed algorithm to obtain randomization number, encryption number and the shift parameter from the initial text\_key. We have given a exhaustive trial run on text\_key and we found that it is very difficult to match the three above parameters for 2 different Text\_key which means if someone wants to break our encryption method then he/she has to know the

exact pattern of the text\_key otherwise it will not be possible to obtain two sets of identical parameters from two different text\_keys. For pure text file we can apply brute force method to decrypt small text but for any other file such any binary file we can not apply any brute force method and it does not work.

## 2. Random Key Generation and MSA Encryption Algorithm:

Before we embed the secret message in a cover file we first encrypt the secret message using MSA method. The detail method is given in our previous publication(1). Here we will describe the MSA algorithm in brief:

To create Random key Matrix of size(16x16) we have to take any key. The size of key must be less than or equal to 16 characters long. These 16 characters can be any of the 256 characters (ASCII code 0 to 255). The relative position and the character itself is very important in our method to calculate the randomization number, the encryption number and the relative shift of characters in the starting key matrix. We take an example how to calculate randomization number, the encryption number and relative shift from a given key. Here we are demonstrating our method: Suppose key entered=AB. We choose table-3 for calculating the place value and the power of characters of the incoming key:

TABLE-III SIZE OF KEY

Length of key(n)	1	2	3	4	5	6	7	8
Base value(b)	17	16	15	14	13	12	11	10
Length of key(n)	9	10	11	12	13	14	15	16
Base value(b)	9	8	7	6	5	4	3	2

$$\text{Sum} = \sum_{m=1}^n \text{ASCII Code} * b^m \text{ ----(1)}$$

Now we calculate the sum for key="AB" using equation(1)

$$\text{Sum} = 65 * 16^1 + 66 * 16^2 = 17936$$

Now we have to calculate 3 parameters from this sum (i) Randomization number(n1), (ii)

(i)Randomization number(n1):  
 $\text{num1} = 1 * 1 + 7 * 2 + 9 * 3 + 3 * 4 + 6 * 5 = 84$   
 $\text{n1} = \text{sum mod num1} = 17936 \text{ mod } 84 = 44$

**Note: if n1=0 then n1=num1 and n1<=128**

(ii)Encryption number(n2):  
 $\text{num2} = 6 * 1 + 3 * 2 + 9 * 3 + 7 * 4 + 1 * 5 = 72$   
 $\text{n2} = \text{sum mod num2} = 17936 \text{ mod } 72 = 8$   
**Note: if n2=0 then n2=num2 and n2<=64**  
 (iii)Relative shift(n3):  
 $\text{n3} = \sum \text{all digits in sum} = 1 + 7 + 9 + 3 + 6 = 26$

We first create 16 X 16 (total 256 characters) key matrix which contains all characters (ASCII code 0-255) and then we give a relative shift(n3) to all elements in the matrix.

After that we apply the following randomization methods one after another in a serial manner:

- Step-1: Function cycling()
- Step-2: Function upshift()
- Step-3: Function downshift()
- Step-4: Function leftshift()
- Step-5: Function rightshift()
- Step-6: Function random()
- Step-7: Function random\_diagonal\_right()
- Step-8: Function random\_diagonal\_left()

For detail randomization methods we refer to our previous work(1).

After finishing above shifting process we perform

- (i)column randomization
- (ii)row randomization and
- (iii)diagonal rotation and
- (iv)reverse diagonal rotation.

Each operation will continue for n3 number of times.

Now we apply encryption process on any text file. Our encryption process is as follows: We choose a 4X4 simple key matrix:

TABLE-IV

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Case-I : Suppose we want to encrypt **FF** then it will taken as **GG** which is just one character after F in the same row.

Case -II : Suppose we want to encrypt **FK** where **F** and **K** appears in two different rows and two different columns. **FK** will be encrypted to **KH (FK→GJ→HK→KH)**.

Case-III: Suppose we want to encrypt **EF** where **EF** occurs in the same row. Here **EF** will be converted to **HG**.

### Changing 4-th bit from LSB Bit of Cover File using encrypted secret message file:

In the present work the last 5000 bytes of the TABLE -V CHANGING 4-TH BIT FROM LSB

cover file we reserved for storing the password and the size of the secret message file. After that we subtract  $n \times (\text{size of the secret message file})$  from the size of the cover file. Here  $n=8$  depending on how many bytes we have used to embed one byte of the secret message file in the cover file. For strong password we have used a simple algorithm as follows: We take XOR operation with each byte of the password with 255 and insert it into the cover file. To retrieve the password we read the byte from the cover file and apply XOR operation with 255 to get back original password. To embed any secret message we have to enter the password and to extract message we have to enter the same password. The size of the secret message file we convert into 32 bits binary and then convert it into 4 characters and write onto cover file. When we want to extract encrypted secret message from a cover file then we first extract the file size from the cover file and extract the same amount of bytes from cover file. Now we will describe the algorithms which we have used in our present study:

We read one byte at a time from the encrypted secret message file (ESMF) and then we extract 8 bits from that byte. After that we read 8 consecutive bytes from the cover file (CF). We check the 4-th bit from the LSB of each byte of that 8 byte chunk whether it is different from the bits of ESMF. If it is different then we replace that bit by the bit we obtain from the ESMF. Our program also counts how many bits we change and how many bytes we change and then we also calculate the percentage of bits changed and percentage of bytes changed in the CF. Now we will demonstrate in a simple case :

Suppose we want to embed "A" in the cover text "BBCDEFGH". Now we will show how this cover text will be modified after we insert "A" within it.

TABLE -V CHANGING 4-TH BIT FROM LSB

Original Text	Bit string	Bit to be inserted in 4-th position	Changed Bit string	Changed Text
B	01000010	0	01000010	B
B	01000010	1	01001010	J
C	01000011	0	01000010	C
D	01000100	0	01000100	D
E	01000101	0	01000100	E
F	01000110	0	01000110	F
G	01000111	0	01000110	G
H	01001000	1	01001001	H

Here we can see that to embed "A" we modify 5 bits out of 64 bits. After embedding "A" in cover text "BBCDEFGH" the cover text converts to "BJCDEFGH". We can see that the change in cover text is not prominent. Only one character is been modified. For text file this change is noticeable but when we do it in some image or audio file then it will not be so prominent. To extract byte from the cover file we follow the reverse process which we apply in case of encoding the message. We simply extract serially one by one from the cover file and then we club 8 bits and convert it to a character and then we write it to another file. But this extracted file is now in encrypted form and hence we apply decryption process which will be the reverse of encryption process to get back original secret message file.

### 3. Results and Discussion:

- Case-1: Cover File type=.jpg      Secret File type=.jpg



Fig\_1:Cover file name: image3.jpg      Fig\_2:Secret message File:joy1.jpg      Fig\_3: Embedded Cover file (Size=3779687B)      ( Size=1870 B )      File name: image3.jpg (Size=3779687B)  
(secret message encrypted before embedding)

- Case-2:      Cover File type=.AVI      Secret message file =.jpg



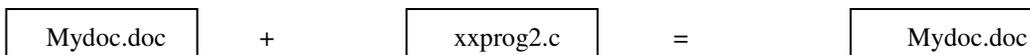
Fig\_4: Cover File name : rhinos.avi      Fig\_5:Secret message File : joy1.jpg      Fig\_6:Embedded Cover File name :rhinos.avi (Size=768000 B)      ( Size=1870 B )      (size=768000 B)  
(secret message encrypted before embedding)

- Case-3: Cover File type=.BMP      secret message file =.jpg



Fig\_7: Cover file name = tvshow1.bmp      Fig\_8: Secret message file= tuktuk1.jpg(size=50880B)      Fig\_9: Embedded cover file name=tvshow1.bmp (size=688856B)      (The secret message file was Encrypted while embedding)

- Case-4: Cover File type=..DOC(MS-Word File)      secret message file =.C



Fig\_10: Cover File Name= mydoc.doc      Fig\_11: Secret message File name =xxprog2.c      Fig\_12: Embedded Cover File name= mydoc.doc (Size=22528 B)      (Size=136 B)      (Size=22528B)  
(The encrypted secret message file is embedded)

**Cover File Name:Mydoc.doc**

To,  
Dr. Amlan Chakrabarti  
A.K.Chaudhuri School of I.T.  
Raja Bazar Science College

Date:

05/07/2010

Dear Dr. Chakrabarti,

How are you? I hope you will be now very busy with your work. I was trying to go and meet you in Science College but because of tremendous work pressure I am unable to meet you. I am leaving Kolkata on 10-th July at 8:00PM. My lecture on 13-th July at 12:20PM-12:40PM(Las Vegas Standard Time). On 14-th July I have to chair one full 2hrs session. My return ticket on 16-th of July and I will return back to Kolkata on 18-th July at 10:20PM. If everything goes right direction then I will again join college on 19-th July. As soon as I return back I will contact you at Science College. I have to finalize the workshop on Matlab.

Amlan I have some queries. If possible you answer it :

- (i) When the result of 6-th of MCA will be declared?
- (ii) When Joyshree and the students of MCA final year can join M.tech in your Dept as I will be out from 10/07/2010 to 18/07/2010.
- (iii) Any idea about the amount of fees to be deposited at time of admission in M.Tech?

**4. Conclusion:**

In the present work we try to embed some secret message inside any cover file in encrypted form so that no one will be able to extract actual secret message. Here we change 4-th bit from LSB bit of the cover file. Our encryption method can use maximum encryption number=64 and maximum randomization number=128. The key matrix may be generated in 256! Ways. So in principle it will be very difficult task for any one to decrypt the encrypted message without knowing the exact key

I am now totally tied up with our second work on Cryptography. It is almost at the finishing stage. I want to send it before I leave from Kolkata. I to also prepare the slides of my lecture to be delivered at Las Vegas.

With kind regards.

Yours sincerely

Asoke Nath  
Nataraj Housing  
381 & 382A M.G.Road  
Kolkata-700 082  
Phone: 24020909

**Secret Message File: xxprog2.c**

```
#include<stdio.h>
main()
{
int c,s;
clrscr();
s=0;
c=1;
while(c<=10)
{
s=s+c;
c=c+1;
}
printf("Sum=%d\n",s);
getch();
}
```

matrix. Our method is essentially stream cipher method and it may take huge amount of time if the files size is large and the encryption number is also large. The merit of this method is that if we change the key\_text little bit then the whole encryption and decryption process will change. This method maybe most suitable for water marking. The steganography method may be further secured if we compress the secret message first and then encrypt it and then finally embed inside the cover file. We have also developed an innovative algorithm to hide any secret message inside an ASCII file or any type of word document

## 5. Acknowledgement:

AN sincerely expresses his gratitude to Department of Computer Science for providing necessary help and assistance. AN is also extremely grateful to University Grants Commission for providing fund for continuing minor research project on Data encryption using symmetric key and public key crypto system. JN is grateful to A.K. Chaudhury School of I.T. for giving inspiration and permission for doing research work.

## 6. References:

- [1] Symmetric key cryptography using random key generator, A.Nath, S.Ghosh, M.A.Mallik, Proceedings of International conference on SAM-2010 held at Las Vegas(USA) 12-15 July,2010, Vol-2,P-239-244
- [2] Data Hiding and Retrieval, A.Nath, S.Das, A.Chakrabarti, Proceedings of IEEE International conference on Computer Intelligence and Computer Network held at Bhopal from 26-28 Nov, 2010.
- [3] Advanced Steganography Algorithm using encrypted secret message : Joyshree Nath and Asoke Nath, International Journal of Advanced Computer Science and Applications, Vol-2, No-3, Page-19-24, March(2011).
- [4] A Challenge in hiding encrypted message in LSB and LSB+1 bit positions in any cover files : executable files, Microsoft office files and database files, image files, audio files and video files : Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath : Journal of Global Research in Computer Science, to be published in May issue,2011
- [5] . New Data Hiding Algorithm in MATLAB using Encrypted secret message :Agniswar Dutta, Abhirup Kumar Sen, Sankar Das,Shalabh Agarwal and Asoke Nath : CSNT-2011 to be held at SMVDU(Jammu) 03-06 June 2011
- [6] Cryptography and Network , William Stallings , Prectice Hall of India
- [7] Modified Version of Playfair Cipher using Linear Feedback Shift Register, P. Murali and Gandhidoss Senthilkumar, UCSNS International journal of Computer Science and Network Security, Vol-8 No.12, Dec 2008.
- [8] Jpeg20000 Standard for Image Compression Concepts algorithms and VLSI Architectures by Tinku Acharya and Ping-Sing Tsai, Wiley Interscience.
- [9] Steganography and Seganalysis by Moerland, T , Leiden Institute of Advanced Computing Science.
- [10] SSB-4 System of Steganography using bit 4 by J.M.Rodrigues Et. Al.
- [11] An Overview of Image Steganography by T.Morkel, J.H.P. Eloff and M.S.Oliver.
- [12] An Overview of Steganography by Shawn D. Dickman

## **SESSION**

# **MALICIOUS CODE + ATTACKS DETECTION**

**Chair(s)**

**TBA**



# Feasibility of Attacks: What is Possible in the Real World – A Framework for Threat Modeling

Ameya M Sanzgiri and Shambhu J. Upadhyaya

Computer Science and Engineering, University at Buffalo, Buffalo, NY, U.S.A

**Abstract**—*In this paper we present a new method to assess risks of attacks faced by a network. Our methodology approaches these risks from the perspective of an attacker in order to bridge the gap created by traditional security schemes which approach from the defender's perspective. These dual perspectives of risk analysis can lead to more effective solutions to security. We describe the various parameters that affect an attack in the real world and use these parameters to analyze the risks of an attack. We also create a model for formally analyzing the risk of an attack using the above parameters. We finally use a case study of jamming attacks on the MAC Layer of the OSI Stack as an illustration and assess the risks for different MAC protocols.*

**Keywords:** Jamming attacks, Perspectives of attack, Risk analysis, Threat modeling

## 1. Introduction

Current security schemes are designed to protect against attacks as seen by the defender based on the limitations and vulnerabilities of his system. From a defender's perspective, the *entire* system is vulnerable to attacks and needs to be secured. Thus, the goal of a defender is to secure the complete system against all possible attacks. However, an attacker's perspective which is orthogonal to the defender's perspective, is to focus on a part of the system and attack. This difference in perspective is further highlighted in their individual goals where an attacker tries to find *one* flaw in the system and leverage it while the defender tries to defend his *entire* system by designing a security scheme. Currently the process of designing a security scheme relies heavily on Attack Graphs [1] and Attack Surfaces [2], [3] which are two methods for formal assessment of risks. Attack surfaces is a conceptual tool used to increase the security of a software during development. Attack graph is an abstraction that divulges the ways by which an attacker can leverage the vulnerability of a system to violate a security policy. It must be noticed that in order to use the attack surface concept on a system, one has to know of all possible vulnerabilities and then optimize the available resources to cover the attack surface.

However, the inherent problem with the design of such schemes is that firstly, the defender does not have enough resources to completely secure his network. The countermeasures usually consider a single attack and are rarely

feasible in terms of implementation complexity or cost to the network. Also, the defender is already at a disadvantage due to the fact that his perspective remains wide and vulnerable, while the attacker's perspective is more focused and specific. This methodology of designing security schemes has resulted in a performance as well as a feasibility gap of schemes in theory and practice which causes them to be reactive in nature. Thus, a paradigm shift is necessary to primarily reduce this gap and to minimize or eliminate the disadvantage of a defender. We propose that such a shift can be obtained by creating a new risk model which would include the attacker's perspective along with the traditional defender's perspective. Such a risk model would culminate in a new classification of attacks (from both perspectives) while providing insights on the kind of information needed by an adversary for an attack. Using the concepts of attack surfaces, one can visualize the objectives of the defender and the attacker as a *game* where the defender tries to minimize the attack surface of the system (securing the system) while the attacker tries to maximize it. This is different from the traditional approach as incorporating the attacker's perspective means the re-examination of some common assumptions with the goal of providing an effective yet practical outlook of the security of a system. The contributions of the paper are 1) A new risk model, 2) Classification of attacks, and 3) A means for proactive security schemes.

The rest of the paper is organized as follows. After discussing the related work and background in Section 2, we present our risk model including the assumptions of the model in Section 3. In Section 4 we discuss the steps of the attacker leading to an attack. Section 5 describes the different factors that need to be considered in an attack. Section 6 presents a qualitative analysis of our model, using jamming attacks as a case study. Section 7 discusses the future work and implications of the model, and then concludes the paper.

## 2. Background and Related Work

Currently, risk analysis enables the separation of the critical or major threats from the minor ones [4]. In understanding the risks, knowledge of the real threats helps place in context the complex landscape of security mechanisms. The evaluation in [4] is conducted according to three criteria: likelihood, impact and risk. The likelihood criterion ranks the possibility that a threat materializes as an attack. The impact

criterion ranks the consequences of an attack materializing a threat. The likelihood and impact criteria receive numerical values from one to three and for a given threat, the risk is defined as the product of the likelihood and impact. Depending on the numerical values received the risk is classified as minor, major and critical. While the approach is relatively simple the likelihood of an attack is based from the system administrator's point of view and does not consider the absence of *a priori* knowledge of the system that an attacker is likely to have. Secondly the evaluation requires the administrator to have expert knowledge of target systems or existing exploits [5]. Further, most risk analyses do not consider network characteristics and their effects. The aforementioned reasons contribute to the inadequacy of such evaluation techniques to correctly analyze risks. The authors of [6] state that an attack graph can provide a methodology for documenting the risks of a system when it is designed. However generation of the graph also requires analyzing the system's purpose and attacker goals which are seldom easy. They also describe how one can utilize the concept of attack graphs in assessing how a multistage attack occurs, where an attacker tries to utilize the intrusion into a system as launching point for other attacks, provided his intrusion is undetected. However, incorporation of network characteristics in traditional risk analysis can prove beneficial and provide the system administrator with some information. Duan et al. [7] present a theoretical analysis of minimum cost blocking attacks on multi-path routing protocols in Wireless Mesh Networks and prove that such an attack is completely infeasible in WMNs. Their evaluation considers the effect of the attack, the characteristics of the target network such as traffic generation patterns and the size of the network on the attack. However, they too make certain assumptions such as the attacker having a way to implement the attack and *a priori* knowledge of the network. Traditional risk models and their assumptions illustrate the extent of the gap between the theoretical and practical risk analysis. We propose to use the parameters that affect an attacker in his attack to analyze the risks of attacks in order to bridge this gap.

### 3. Risk Model

Existing security schemes are reactive due to the inability of the defender to foresee the domain of all possible attacks. Researchers make theoretical assumptions and develop complex security solutions yet systems can be compromised by an attacker through a simple, low cost and practical means that was not foreseen by the defender. This problem is exacerbated by the widening gap between the theoretical and practical aspects of security. Some of the attacks theorized by researchers, although wishful, may never occur in practice due to the high cost of attack on the part of the adversary or due to the practical limitations of hardware devices. Further, most formal tools like the ones discussed above require a thorough knowledge of the individual system components

and their interaction with each other, the lack of which leads to inaccurate or ineffective security solutions. Hence one needs a new risk model that can classify the attacks from a more practical perspective that is not only feasible but also effective. To achieve this we re-examine the assumptions that are made in the literature and include both the attacker's and the defender's perspectives on an attack. Including the attacker's perspective on attacks however requires one to analyze and enumerate the factors that an attacker would consider in his attack.

#### 3.1 Assumptions of the Model

The assumptions made by a model have a direct effect on the analysis of risks and can cause unreliable assessments. This can lead to a false sense of security or cause inefficient resource allocation by a system administrator. We assume that the attacker has no or very little *a priori* information about the target network. This includes knowledge about network components, its purpose or its usage. However, the attacker does have the resources and technical knowledge of implementing an attack and can gain the knowledge of the system he intends to attack. This is a valid assumption as we shall discuss in Section 4. We also apply the same constraints on the hardware the attacker possesses as in the real world. This however does not imply that the network is physically isolated, in the sense that an attacker is quite capable of both performing active and passive attacks on the target network. The scenarios of insider attacks and attacks resulting due to the mistake of a target network's user is not considered and is beyond the scope of this paper.

### 4. Modus Operandi of an Attack

Before we present the steps of an attack, we need to clearly define an attack. An attack is a series of intentional steps taken to gain some unauthorized result. Since the steps of any attacker are intentional and methodical, it should be generally quantifiable and can be represented as a process, which in turn would help in creating a proactive defence strategy. An attack generally follows a sequence of steps, viz. Reconnaissance, planning, collection, analysis and execution while targeting a system [8]. The goal of these steps is to first obtain the Information Content necessary for the attack in order to execute an attack. Thus, the procedure to gain information about the network, is the precursor to an attack. From an attacker's point of view, this would include gaining as much information of the system as one can so as to develop one's strategy for attack. What we can broadly classify as information content are the features of the target network such as the data in the network, components, protocols of the target network, etc. It is important to understand that from an attacker's perspective this information content comprises of all the factors that has to considered for staging an attack. In Section 5 we present a detailed analysis and motivation of these factors. While the exact amount of information

required for an attack depends on the skills of the attacker it can be fairly assumed that most of this information is essential for an attacker. In the current literature so far, it is usually assumed that the attacker already has the required information content. However, we believe that if a defender has to regain his advantage security schemes need to increase the cost of the process of collecting information content for the attacker.

## 5. Motivational Factors of an Attack

The goal of any risk model is to assess the risk of an attack and classify the threat it poses to a network. However, from the defender's perspective the risk of an attack should relate closely to a real world scenario so as to be able to efficiently allocate his resources. In most cases the risk analysis of an attack takes into account only the defender's perspective and knowledge, and presents a rather pessimistic scenario. However, if we were to take in factors from the attacker's perspective as well, the parameters that affect the analysis of a risk change. When we consider both these perspectives, the risk of an attack depends on – i) motivation of attacker, ii) probability of attack, iii) easier alternative, iv) target network characteristics and v) cost of attack. This is described next.

### 5.1 Motivation of an Attacker

This parameter directly affects the risk assessment of an attack and asymptotically either elevates or depreciates the risk of an attack. It is scientifically difficult to quantify this parameter as it depends on an attacker's behavior. However, one can try to quantify it by observing other factors such as the type, target and the purpose/effect of the attack. In [4], the authors state that an attacker's motivation can be categorized to be High, Medium and Low. Thus, both the purpose of attack and the motivation contribute to the overall risk of an attack. For example, a highly motivated attacker attacking out of inquisitiveness is likely to be less dangerous than one for financial gain.

### 5.2 Probability of Attack

This parameter denotes if an attack is desirable based on two factors – *cost of an attack* and the *severity factor* of the attack. We define *cost of attack* as a combination of time, the hardware needed and the general strategy required for an attack. Severity factor is defined as the effect an attack has on a network. It is evident that the probability of an attack is likely to increase as the cost decreases and the severity increases. Thus we can quantify the probability of attack as

$$\Pr(\text{Attack}) = f(\text{Severity}_{\text{Attack}}, \text{Cost}_{\text{Attack}}) \quad (1)$$

### 5.3 Easier Alternative

This parameter relates the risk of an attack to another attack which is at a higher probability due to either increased severity or lower cost for a given network.

### 5.4 Target Network Characteristics

This parameter describes the features and characteristics of the target network. It encompasses other features such as system level misconfiguration [9], the unexpected side effect of operations [10] and platform specific attacks which can be exploited. Another factor that would be considered by an attacker is the type of traffic flowing through the network.

### 5.5 Cost of Attack

This parameter quantifies what it would cost an attacker to launch an attack. The three factors that make up this parameter are *Time*, *Strategy* and *Hardware*. It is evident that the first two factors are directly dependent on each other and it is the prerogative of an attacker to decide which factor is more important to him. These two factors affect the third factor – as the attacker has to invest in the appropriate hardware depending on which of the above two factors he gives more importance to.

#### 5.5.1 Time

This parameter denotes the time taken for an attack which includes the time for gathering information and implementation.

#### 5.5.2 Hardware Constraints

This parameter specifies the constraints that an attacker has to both work with or face when launching an attack. Suppose an attacker takes over a node in a Wireless Sensor Network. The energy constraint as well as the memory constraint would be a factor that would prevent him from making more complex attacks. On the other hand the same constraints (as the characteristic of the target network) also allow him in implementing a denial of service attack. Similarly the uncertainty of radio ranges [11] and radio hardware could affect the severity of his attack.

#### 5.5.3 Strategy

This parameter features in the cost of an attack and is an important parameter. We further subcategorize it into:

- 1) Practical Difficulties: This factor considers the remaining aspect of difficulties while dealing with network hardware such as synchronization [12] and basic cryptography in networks. We also use this factor to represent the unpredictable behavior of the wireless medium which equally affects the attacker as the target network such as radio ranges.
- 2) Implementation: This refers to implementation difficulties of attacks due to built-in defenses in the target network or hardware constraints.
- 3) Identification of Network Protocols: The correct functioning of a network protocol relies on specifications and implementations [13]. However implementations are inherently more complicated and could introduce discrepancies and vulnerabilities, even though the analysis for soundness

validation may not discuss it [14]. It has been shown that most Internet protocols such as ICMP, TCP are subject to these discrepancies [15]. The universal presence of these discrepancies is due to the fact that network protocols cannot be completely and deterministically specified; instead opportunities are provided for implementations to distinguish itself [16]. The author of [17] states that the identifying protocols employs the following two methods:

**Network Protocol Fingerprinting:** This is the process of identifying a protocol by analyzing its output characteristics and traces based on the user input using tools like NMAP [18] or TBIT [19]. This method is called active fingerprinting since one can change the input to get different outputs. However it is also prone to alerting system administrators. Passive fingerprinting, where one does not provide inputs but only observes the output is a time intensive process. Further, it is extremely difficult to conduct rigorous proof about the validity of fingerprinting experiments [20]. It has been shown that the complexity and time required for fingerprinting make it infeasible in practice [16].

**Network Protocol Fuzz Testing:** This is the process of mutating the normal traffic to reveal unwanted behavior such as crashing or confidentiality violation [21]. However the authors also states that due to various factors this method is also mostly infeasible and inaccurate.

4) Selection: This denotes the methodology of the attacker including factors such as gaining information content by gathering and storing data, analyzing it to obtain target network characteristics, and verifying the results. Too aggressive methods of gathering data, could unintentionally alert a system administrator about the attacker's intention. The information content includes operating system, hardware, type of data, network protocols, purpose of network, size of network, topology, etc. We are specifically interested in identifying a network protocol which contrary to intuition, is much more complex. For instance, the author of [22] suggests that the difference among NewReno and Reno (TCP) can be discovered only when multiple packets are dropped within the same congestion window. This suggests that the time and resources required by an attacker to accurately assess a network protocol are important.

Figure 1 summarizes our risk model along with the underlying factors and their relationships. In the following section we use a qualitative approach to validate our risk model and highlight the novelty of our approach by evaluating the risk of a jamming attack against a network. We first present the risk analysis of the attack by evaluating it only from the defender's perspective. We then show how our model, by incorporating the attacker's perspective, evaluates the same "highly" probable attack as a "low" probability attack.

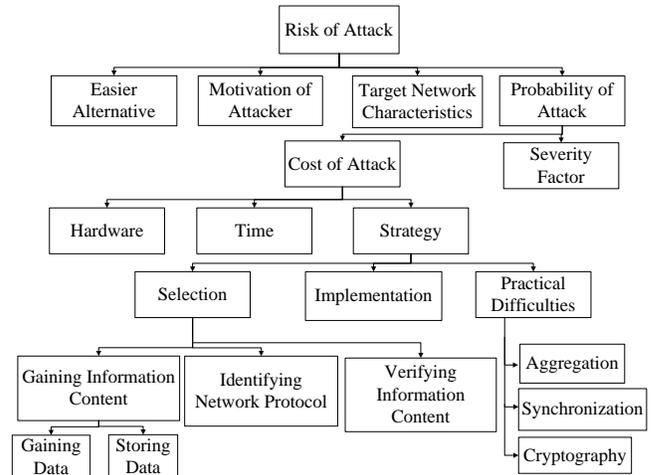


Fig. 1

RELATION BETWEEN FACTORS AFFECTING RISK ASSESSMENT IN OUR MODEL

## 6. Case Study-Jamming attacks

### 6.1 Overview

Jamming attacks target the Medium Access Control Layer (MAC) or the Physical (PHY) Layer of the OSI stack. This attack involves a jammer causing interference by emitting a RF signal continuously, disrupting the operations of a target network. However, the authors of [23], [24] state that a broader range of behaviors can be adopted by a jammer and a common characteristic of jamming attacks is that their communications are not compliant with the MAC protocols. They define a jammer as any entity interfering with the transmission or reception of wireless communications by either preventing a source from sending out a packet or reception of legitimate packets, leveraging mostly on the shortcomings of the MAC or PHY protocols. Any attack based on this idea is classified as a jamming attack.

#### 6.1.1 Profiles of a Jammer

The success of a jamming attack like most attacks is dependent on the strategy chosen by the jammer. It must be noted that the strategy in this kind of attack includes both the layer of choice, i.e., either PHY or MAC and the model used to *jam* it. There are four different models or profiles of jammers – Reactive, Constant, Random and Deceptive [24].

#### 6.1.2 Severity of Jamming Attack

Jamming attacks at the MAC level are effective due to the simple strategy and the difficulties in detection [23], [25]. Further since these attacks specifically target the protocols there are no effective means of circumventing the problem. Particularly, the problem lies in the inability of the

network devices to distinguish between *malicious* jamming and *unintentional* interference. The only effective solutions are changes to the MAC protocol or using expensive radio level technologies at the PHY level such as Direct-Sequence Spread Spectrum (DSSS) techniques [26].

## 6.2 Effectiveness of Jamming Attack

From a network perspective the effectiveness of jamming attacks is dependent on the following two necessary features of the network.

- 1) Target Network Characteristics: WSNs or Ad-Hoc Networks are attractive targets due to their resource constrained nature since jamming attacks aim at depleting the energy of the devices by reducing their sleep times, increasing either the number or time of re-transmissions. Another characteristic of jamming is that it directly affects the data flow in a network making it effective against networks where data freshness is critical.
- 2) Hiding in Plain Sight: The success and effectiveness of the attack also depends on the jammer's ability to remain unidentifiable in the network. While a part lies in the implementation of the attack, a major part is the network's inability to differentiate between jamming and congestion. In addition to this it is also necessary that the network cannot identify the misbehaving devices. This implies that any kind of scheduled access to the medium is ruled out, as in such cases the jammer(s) can be easily identified and the network can differentiate if it is under attack.

## 6.3 Consideration of Jammer's Perspective

As explained in Section 6.1.2, the effectiveness and strategy of a jamming attack makes it hard for a network administrator to defend without investing in expensive countermeasures. Further, current countermeasures require an elaborate protocol of secret sharing for the scheme to be viable and effective. Since the defense strategies against these attacks are expensive, they are unlikely to be widely deployed. Considering this one would assume that such attacks would be nearly impossible to prevent or protect and should be widespread. However, the lack of evidence of such attacks in real-world [27] implies that while theoretically plausible there are some caveats that make them unpopular. This indicates that traditional threat modeling which considers only the defender's perspective does not encapsulate the risk convincingly. Further, it has to be noted that these attacks are unpopular from an *attacker's* perspective which means that one has to consider an attacker's perspective. A reasonable explanation as to why such an attack is unattractive to an attacker could be that the effort required for successful initiation of the attack is large with diminishing returns or that the attack does not comply with the motivations of most attackers. For an attacker, the effort required for initiation is the effort (time and cost) to gain the information content that convinces him that the attack can be successful. Further, in

DoS attacks an attacker's motivation is likely to be low since there is nothing tangible to gain. Since we are incorporating the attacker's perspective we have to also present some of the concerns in planning such an attack. In the following subsection we first present these concerns and try to analyze if one of the two factors mentioned above or a combination of them is the reason for the unpopularity of such attacks.

## 6.4 Attacker's Perspective and Concerns

To begin with an attacker has to spend considerable resources to ascertain that the network complies to the two necessary conditions described in Section 6.2. This includes finding the answers to the following questions:

- 1) What is the type of network? This critical question has to be addressed for the attacker to know what target network he is attacking.
- 2) Is the concern of the network energy or data freshness? This question would tell an attacker if a jamming attack is going to be effective or not.
- 3) What is the type of data flow in the network – Periodic, Query based or Event driven?
- 4) If the concern is data freshness, what are the standard packet sizes that flow in the network? Are there other features in the network such as aggregation or network coding? Answers to the above questions help in choosing the kind of jammer profile. Methods such as aggregation/network coding will reduce the effectiveness of the attack or require deploying/taking over more resources.
- 5) Identifying the *exact* protocol of the network. This is another critical dependency for an attacker. A motivating example for this is that the implementation of the attack is completely different in case of a CSMA MAC protocol from a preamble based MAC protocol. If the target network is running a schedule based MAC protocol, the attack will be ineffective.
- 6) Identifying physical access to the channel. What is the power required to jam? For example, if the devices transmit using BPSK or AM [28], due to the robustness of the signals the jamming attack may not be viable.

These are some of the concerns an attacker has to address to guarantee success to even an extent. However the following are also some additional practical concerns which an attacker needs to address.

- 1) What is the size of the network? What is its topology?
- 2) How to implement his attack? Does an attacker have physical access to the network? Where to place the jamming nodes?

Figure 2 presents the cookbook steps of an attacker's preparation for a jamming attack based on our analysis. The figure shows that there are 3 main steps for an attacker, namely, *Identifying Network Characteristics*, *Identifying Exact Protocols* and *Implementation Concerns*.

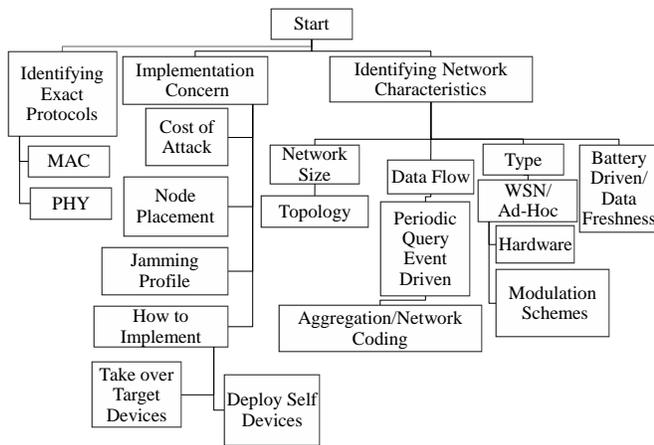


Fig. 2

STEPS AN ATTACKER HAS TO TAKE FOR A JAMMING ATTACK

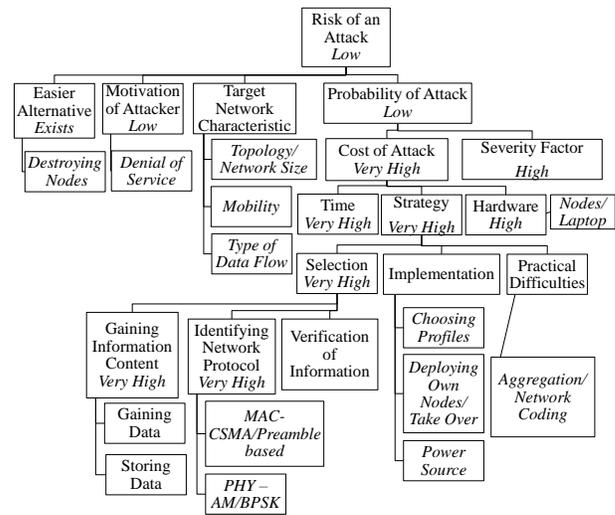


Fig. 3

RISK MODEL APPLIED TO JAMMING ATTACKS

## 6.5 Attack Implementation Concerns

Section 5.5.3 describes the concerns and analysis of identifying network characteristics and exact protocols. We now focus on the implementation concerns for the practical aspects of the attack. The implementation of the attack requires us to consider two scenarios as shown in Figure 2 – Takeover Target Devices or Deploy Own Devices. We present an analysis below:

1) Takeover Target Devices: In this scenario, the attacker has to take over the nodes of the target devices and use them in his attack. Since we do not consider human interaction, an attacker has to get within transmission range or have physical access to the devices. In cases of WSN or Ad-Hoc networks tamper proof (TPD) devices [29] could easily circumvent this problem. Further, if physical access is possible, then the attacker has easier options such as destroying them - a feasible alternative in a DoS attack.

2) Deploy Own Devices: Here, the attacker deploys his own devices. While this scenario is feasible and is likely to improve the success rate, the cost of attack also increases. The attacker has to invest in the devices just for denying service or interfering with the performance. Again easier alternatives such as destroying devices exists. The scenario of a more powerful device (such as a laptop) against sensors does exist, however the effect of jamming would be localized to a small region. Further, even in such cases the attacker too is restricted with the same energy constraints. Deploying more than one laptop will increase his cost of attack manifold.

The next aspect of implementation is choosing an optimum jammer profile since all the profiles are orthogonal to each other in terms of effect, cost and target networks.

1) Constant: This profile is effective on all kinds of protocols. However, the type of data flow also directly affects its efficiency. If the data flow is periodic, event driven or query based, constant jamming is going to be wasteful and will also affect the life of the jammer nodes as they need to transmit all the time.

2) Deceptive: This profile is very effective on a very small subset of preamble based protocol. However, it requires the jammer to be able to exactly ascertain the protocol as it has to send the exact preamble or the packet.

3) Random: This profile is the most efficient profile, provided that the jammer is able to configure the exact time/distribution of sleeping and jamming. Its efficiency reduces significantly in Event Driven networks and would not be effective at all in query based networks. It is again important to note that this profile attacks data freshness more than energy consumption.

4) Reactive: This profile is the most effective but also the least efficient since the jammer node has to be "ON" all the time. While it circumvents the amount of information content required by an attacker, networks with aggregation or small packet sizes would not be really affected. Further, considering that the energy consumption for reception is nearly equal to transmission, this profile would lead to wastage of energy.

The most important factor in this attack after observing the steps of a jamming attack is the cost of attack. This attack aims at a small subset of networks and requires too many necessary conditions for the attack to be successful. Simply put, this kind of attack extracts a huge cost in terms

of time and resources from the attacker, due to the amount of reconnaissance required. The description above leads to a risk model for jamming attacks as shown by Figure 3. This is an instance of the generic model from Figure 1 where the boxes represent the factors we have identified, with their respective values shown in italics.

## 7. Discussions

We have presented a new risk model that incorporates factors from the attacker's perspective. We believe this is a new approach that can be used for creating new proactive security and privacy schemes by including features such as obfuscation/confusion to provide efficient and practical measures. We have demonstrated the effectiveness of our model using jamming attacks as an illustration and also highlighted the novelty of our approach as compared to the generic approach in the literature that takes into account only the defender's perspective. Specifically we show how some of the assumptions made in the literature are questionable. While we have studied the attacks using a systematic qualitative analysis, our next step will be a more formal mathematical analysis for deeper insights into the complexity of attacks.

## Acknowledgments

This research is supported in part by NSF Grant No. IIS-0916612. Usual disclaimers apply.

## References

- [1] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," in *Computer Security Foundations Workshop, 2002. Proceedings. 15th IEEE*, 2002, pp. 49 – 63.
- [2] P. Manadhata and J. Wing, "An attack surface metric," *Software Engineering, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2010.
- [3] M. Howard, J. Pincus, and J. Wing, "Measuring relative attack surfaces," *Computer Security in the 21st Century*, 2003.
- [4] M. Barbeau, J. Hall, and E. Kranakis, "Detecting impersonation attacks in future wireless and mobile networks," in *In Proceedings of MADNES 2005 Workshop on Secure Mobile Ad-hoc Networks and Sensors - Held in conjunction with ISC2005*. SVLNCS, 2005.
- [5] D. Geer and J. Harthorne, "Penetration testing: A duet," in *ACSAC*, 2002, pp. 185–198.
- [6] S. Gupta and J. Winstead, "Using attack graphs to design systems," *IEEE Security & Privacy*, vol. 5, no. 4, pp. 80–83, 2007.
- [7] Q. Duan, M. Virendra, and S. J. Upadhyaya, "On the hardness of minimum cost blocking attacks on multi-path wireless routing protocols," in *ICC*, 2007, pp. 4925–4930.
- [8] C. Peikari and S. Fogie, *Maximum Wireless Security*. Indianapolis, IN, USA: Sams, 2002.
- [9] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," *Security and Privacy, IEEE Symposium on*, vol. 0, p. 273, 2002.
- [10] S. Chen, Z. Kalbarczyk, J. Xu, and R. K. Iyer, "A data-driven finite state machine model for analyzing security vulnerabilities," in *In IEEE International Conference on Dependable Systems and Networks*, 2003, pp. 605–614.
- [11] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*. New York, NY, USA: ACM, 2004, pp. 125–138.
- [12] S. Dolev, S. Gilbert, R. Guerraoui, F. Kuhn, and C. Newport, "The Wireless Synchronization Problem," in *Proceedings of the 28th Annual Symposium on Principles of Distributed Computing*, 2009.
- [13] D. Lee, D. Chen, R. Hao, R. E. Miller, J. Wu, and X. Yin, "A formal approach for passive testing of protocol data portions," in *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*. Washington, DC, USA: IEEE Computer Society, 2002, pp. 122–131.
- [14] G. Lowe and B. Roscoe, "Using csp to detect errors in the tmn protocol," *IEEE Trans. Softw. Eng.*, vol. 23, no. 10, pp. 659–669, 1997.
- [15] R. Beverly, "A robust classifier for passive tcp/ip fingerprinting," in *PAM*, 2004, pp. 158–167.
- [16] G. Shu and D. Lee, "Network protocol system fingerprinting – a formal approach," in *Proceedings of IEEE Infocom*, 2006.
- [17] G. Shu, "Formal methods and tools for testing communication protocol system security," Ph.D. dissertation, Ohio State University, 2008.
- [18] F. Yarochkin., "Remote os detection via tcp/ip stack fingerprinting." 1998, <http://www.insecure.org>.
- [19] J. Padhye and S. Floyd, "On inferring tcp behavior," in *In the proceeding of SIGCOMM*, 2001, pp. 287–298.
- [20] D. Lee and K. Sabnani, "Reverse-engineering of communication protocols," in *Network Protocols, 1993. Proceedings., 1993 International Conference on*, 1993, pp. 208–216.
- [21] O. Arkin and F. Yarochkin, "Xprobe2 - a 'fuzzy' approach to remote active operating system fingerprinting." 2002, <http://www.sys-security.com>.
- [22] K. Fall and S. Floyd, "Simulation-based comparisons of tahoe, reno and sack tcp," *SIGCOMM Comput. Commun. Rev.*, vol. 26, no. 3, pp. 5–21, 1996.
- [23] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *Network, IEEE*, vol. 20, no. 3, pp. 41–47, 2006.
- [24] Y. Chen, W. Xu, W. Trappe, and Y. Zhang, *Securing Emerging Wireless Systems: Lower-layer Approaches*, 1st ed. Springer Publishing Company, Incorporated, 2008.
- [25] A. Wood and J. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.
- [26] R. A. Poisel, *Modern Communications Jamming Principles and Techniques*. Artech House Publishers, 2006.
- [27] S. Peters, *2010 CSI/FBI Computer Crime and Security Survey*. Computer Security Institute, December 2009.
- [28] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing (4th Edition)*. Prentice Hall, Mar. 2006.
- [29] P. Ning and W. Du, "Journal of computer security," January 2007.

# Denial of Service (DoS) Attack Detection by Using Fuzzy Logic over Network Flows

S. Farzaneh Tabatabaei<sup>1</sup>, Mazleena Salleh<sup>2</sup>, MohammadReza Abbasy<sup>3</sup> and MohammadReza NajafTorkaman<sup>4</sup>

Faculty of Computer Science and Information System,

University of Technology Malaysia(UTM) , Kuala Lumpur, Malaysia

<sup>1</sup>(farzanehtabatabaei@gmail.com), <sup>2</sup>(mazleena@utm.my), <sup>3</sup>(ramohammad2@live.utm.my),

<sup>4</sup>(rntmohammad2@live.utm.my)

**Abstract** - *Intrusion Detection System (IDS) is the tool that is able to detect occurrences of intrusion at host, network, as well as application. One of the most common network attacks is Denial of Service (DoS) attack. In DoS attack, a single host will send huge number of packets to one machine and thus make the operating of the network and host slow. There are several algorithms that have been proposed to detect DOS attacks and most of these solutions are based on detection mechanisms that have the potential of producing high number of false alarms. In addition, most of the solutions are monitoring and analyzing packets inside the network instead of network flow. In this paper, signature of selected attacks such as Smurf, Mail-Bomb and Ping-of-Death which are based on network flow is considered. The proposed engine monitors the network flows to detect attacks and the results show less false negative error during monitoring. In addition signature based IDS which use fuzzy decision tree for monitoring network flow proves that there are improvements on speed of detection and also performance of system.*

**Keywords:** DOS Detection, Fuzzy Logic, IDS

## 1 Introduction

Reports of the internet usage showed that the number of internet users is increasing and unfortunately this phenomenon has attracted attacks on the network. Consequently this has raised the concern of the network security especially by the services providers and they are always looking for solutions to monitor and check packets being received from clients to avoid any kind of attacks.

Security mechanism used in a network is to prevent the system from any kind of attack and to stray away from any unsecured state. As prevention mechanism could not capable to impede the attacks entirely, so new level of security will be needed and the goal is to detect and stop the attack as soon as possible [1].

An intrusion-detection system (IDS) dynamically monitors the actions taken in a given environment such as host or traffic of network and decides whether these actions are symptomatic of an attack or constitute a legitimate use of the environment [2]. The two most common detection

techniques which could be applied in IDS are signature based detection and anomaly based detection [3].

Signature based detection technique in IDS is looking for characteristics of known attacks and IDS try to find the similarity between previous behavior of the system or network with characteristics of known attack in signature database but in this technique IDS cannot detect novel attacks [4] [5] [6] [7].

Anomaly detection technique adopts the normal condition of the network traffic or behavior of host as criteria of anomaly; by this approach it can detect unknown attacks. But this approach create a percentage of detection errors because of the difficulty to define the normal state of the network traffic precisely [4] [5] [6] [7].

Denial of Service (DoS) attack uses up the resources of host, network or both in the way that normal user as a client could not access to the Server [8].

Some researches use artificial intelligence [9] and data mining [10] and fuzzy [11] in IDS to detect intrusion. Recently fuzzy based intrusion detection systems have proved robustness to noise, self-learning capability, and the ability to build initial rules without the need for a priori knowledge [12].

Although a variety of approaches proposed to detect intrusion like DoS but still the accuracy and efficiency of detection needs more improvement. So information security experts are still trying to improve the mechanism for detection of DoS attacks by several algorithms.

## 2 Problem Statement

According to Table1, the number of incidents increased rapidly and thus pushes researchers to give more and better effective way to stop those incidents. One of the solutions is to build IDS which can detect more intrusion with small false positive rate. According to [12] the rate of detection and false positive do not satisfy users especially in anomaly based IDS.

This paper is about introducing an engine for IDS which detect some types of DoS attacks with better rate of detection

and less false positive errors. This mechanism will be signature based and the engine use fuzzy algorithm and it monitor the network flows for better performance. Number of security incident from CERT [13] website is shown in Table1. *“Given the widespread use of automated attack tools, attacks against Internet-connected systems have become so commonplace that counts of the number of incidents reported provide little information with regard to assessing the scope and impact of attacks”* [13]. Therefore, CERT stopped providing this statistic at the end of 2003.

Table1: Number of security incident reports received  
by CERT

Year	Number of Incident
2003	137,529
2002	82,094
2001	52,658
2000	21,756
1999	9,859
1998	3,734
1997	2,134
1996	2,573
1995	2,412

### 3 Solution and Methodology

The progress of providing a solution for the stated problem is divided to two phases; first, design and second, analyzing. After defining the objectives previous researches and methods which are used by different researchers are studied and then a system was designed which is based on these studies and the purpose is to reach the objectives. In second phase which is called analyzing, the consequence of the design and its effects on the system improvement is determined.

This paper lead to design an application which use Fuzzy algorithm to detect DoS attacks in TCP and ICMP protocols, in this algorithm Fuzzy logic will process the data which was extracted from Network Flow Header to find the intrusion.

Data collection after identifying the problem provided the idea as paper topic. Basic and general information about IDS is gathered and then a discussion about DoS attacks and their behaviors is conducted. These studies show two important problems in IDS, they are low speed and low detection rate for detecting DoS.

The solution which is proposed is monitoring the network flow using fuzzy system to increase the speed and rate of DoS attack detection.

### 3.1 Design

In this step, the study on related works is done and the mechanism of similar systems in details to find out what mechanism should be used to detect DoS attacks, is analyzed. There was an analysis between anomaly based detection and signature based detection in IDS, and finally signature based detection selected, because most of DOS attacks have their own signature and rate of detection is high in signature based detection.

A review of previous researches was done to get a complete view about the proposed mechanisms for intrusion detection system for detecting DOS attacks and finally fuzzy decision tree was chosen to be the engine of IDS for analyzing the traffic and find DOS attacks.

The system designed in a way to reach the objectives which declared in identifying problem phase and also considering knowledge from Gathering data.

In this phase proposed architecture of the system is described in detail. Enough consideration should be taken to design the component to have a correct output from each module and whole system (attack report).

### 3.2 Analyzing

The designed system monitors the network traffic and put all of the packets into the network flows but in the same time fuzzy engine finds suspicious packet and save those flows in the array. At the end, fuzzy decision tree will check headers of suspicious flow and in case of attack, the system generates the error.

By applying fuzzy logic on traffic sample of Defense Advanced Research Projects Agency (DARPA) from Lincoln Laboratory of Massachusetts Institute of Technology (MIT) the system is tested and detection rates and performance of application is shown. The output of this phase is to achieve designing the algorithm to detect DoS attacks by Fuzzy Engine.

## 4 System Design

Based on the findings from related researches and works the design of the system is introduced. There is an overview of the solution to show all the processes and how it can improve the performance or accuracy of the system. The system architecture is the first thing which is mentioned here, and then an overview of Fuzzy and Network Flow is mentioned. The full description about how to apply Fuzzy and Network Flow on IDS is followed by how these two improve the detection and speed.

### 4.1 Architecture

Figure 1 shows the whole process of the system with some details. In this Design the IDS collects all of the packets from Traffic Sample and put them inside of the flows to save inside of the memory. Meanwhile, the fuzzy engine is collecting any suspicious packet, and put them inside of suspicious flow. Whenever the suspicious flow is finished, the fuzzy engine will check it for the final attack report.

## 4.2 Preprocess Data

TCP and ICMP packets from network are gauge and the network flows are constructed by the Network Flow Engine (NFE). Identification of flows for TCP packets are based on the numbers of packets from same Source, Destination, Source Port and Destination Port. It starts with SYN packet and it will finish when the FIN packet arrive. On the other hand for ICMP protocol, NFE could be defining two types of packets. First packet contains a request from one machine to another and second packet is the answer of request. The NFE will check the network flows for any anomaly behavior. Network Flow contains numbers of packet in one communication in the network.

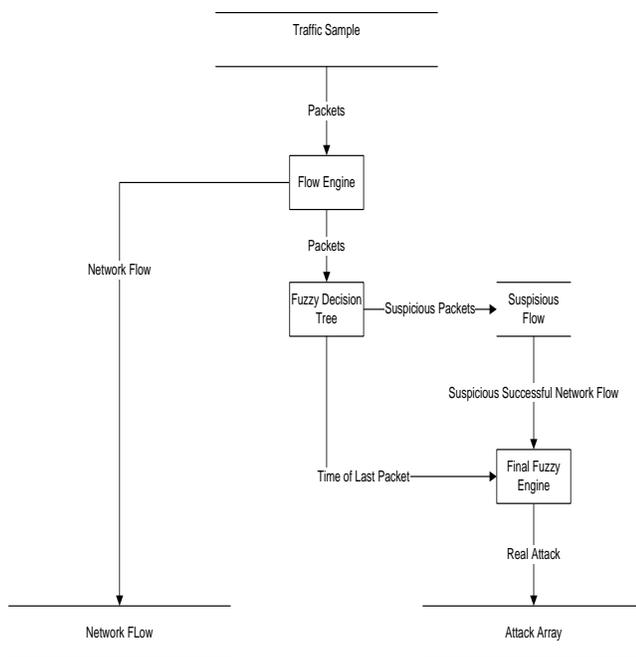


Figure 1: DFD System Architecture Context Diagram

## 4.3 Design Issue

Some of the issues in IDS are false positive error, false negative error, rate of detection, performance and speed. By using Network Flow as input and applying fuzzy decision tree as an engine for intrusion detection the result could have less False Positive error and better rate of detection and also better performance and speed.

## 4.4 Signature Based

As mentioned before the objective of this paper was to detect 4 types of DOS attacks in DARPA traffic sample, so there is an explanation about each attack.

### 4.4.1 Land Attack

If protocol of incoming packet is TCP and Source IP and Destination IP are same as each other and Source Port is equal to Destination Port the Land attack will happen.

### 4.4.2 Mail-Bomb Attack

There will be a TCP flow in this attack after establishment of one TCP connection between two computers. In this flow SMTP port will be used to send email but the number of packets in one Flow is about 10,000 packets and size of each packet is about 1,000 byte. So size of flow will be about 10 MByte.

### 4.4.3 Smurf Attack

There will be several ICMP flows in this attack, the number of packets in one flow is low but size of each packet is approximately 1,000 byte. However the number of flow will be high because several computers send a large packet to single computer. The packet contains Reply message but Request message never sent from victim.

### 4.4.4 Ping of Death Attack

There will be large number of oversize IP packets in one flow from one computer to another. Each packet is about 1,000 byte and size of attack flow if high, approximately 64,000 Bytes and it is under ICMP protocol which causes rebooting, freezing and crashing the victim machine.

## 4.5 Fuzzy

Fuzzy sets just include 0 and 1, so there could be only two options, but in fuzzy logic by combination of several fuzzy set there could be several answers. Table 2 explains how fuzzy set and fuzzy logic combined in this system and made the fuzzy decision tree for detecting the 4 types of DOS attacks.

Meanwhile the fuzzy engine looks for suspicious packets to change the status of flow from Normal to Suspicious to speed up the detection. This sub-process will be suspicious to packets which have following attribute (pseudo code form):

### For Land attack it is using these rules

IF flowprtc equal to TCP

IF flowsrc equal to flowdest

Record to Land attack array

### For MailBomb attack the rule which applied is

IF flowprtc equal to TCP

IF flowdestPort equal to SMTP

IF flowsize >10 MB

Record to MainBomb attack array

**For Smurf attack the rule applied is**

IF flowprtc equal to ICMP

IF info contain Reply

FOR Packet from last minute, to this Packet, go one by one

IF info not contain Request from same machine

Record to Smurf attack array

**For Ping of Death attack rule applied is**

IF flowprtc equal to IP

IF info contain ICMP

Record to Ping of Death array

Table 2: Combination of fuzzy set and fuzzy logic

Attack	If Prtc = TCP	If Src = Dest	Flow Size	Packet count	No Flow	If Prtc = ICMP	If Prtc = IP	Packet Size
Land	1	1	-	-	-	0	0	-
Mail Bomb	1	-	More than 10Mb	More than 1000 packet	-	0	0	-
Ping of Death	0	-	64.000b<M<10 or H>10Mb	More than 60 packet	-	1	1	More than 1000b
Smurf	0	-	-	-	10	1	0	-

Figure 2 shows the full Fuzzy decision tree for detecting four types of DOS attacks. All of the rules in the Fuzzy decision tree are based on the attack signature which comes from DARPA website.

### 5 Analysis and Conclusion

The design of fuzzy decision tree which can detect four types of DOS attacks by analyzing network flow is described. The proposed architecture is a guideline for implementation of the system. Experiments are conducted with the used of dataset from DARPA.

Previous solutions on IDS were based on detection method which used packets data and resulted with high false errors. In this study the IDS design focused on solving the problem by applying fuzzy decision tree as processor and network flow as input of system.

In this system, all of the packets are initially preprocessed and the subsequently the network flows are constructed. During this process, fuzzy engine will put all suspicious packets in to the memory. Finally, the flow header will be generated, the suspicious flow will be checked again by fuzzy engine and detected attacks will be printed.

Using network flow as input of the proposed IDS was a method to increase detection rate of four types of DOS attacks, for example Land attack start with a flow which contain same source and destination IP, or in mail bomb attack the system must save the size of SMTP flow, attack like Smurf must be detected by counting number of flow to

one machine and finally in ping of death attack number of packet in one flow must be high.

Another method was using fuzzy decision tree inside of IDS. One of the main focuses in this project was to use simplest rules to detect four types of DOS attack; simple rules make time of process less so the speed of detection will be fast. Also for improvement of rate of detection the fuzzy decision tree applied rules from DARPA website, so in that case all of the signatures are 100% true and reliable. Table 3 shows the performance of fuzzy decision tree to detect DoS.

Table 3: Performance of Fuzzy decision tree to detect DoS

Name of attack	False Negative	False Positive	Rate of Detection
Land Attack	0%	0%	100%
MailBomb Attack	0%	0%	100%
Ping-of- Death	0%	0%	100%
Smurf	0%	0%	100%

In this solution the Land attack will be detected when one TCP packet which contain SYN come to the network with same source IP and destination IP and same source port and destination port, at that moment the system will report the alarm.

Mail-Bomb attack will be detected when size of one SMTP flow exceeds a determined critical point and the system will generate alarm.

This system will generate alarm of Ping of Death attack when flow size in ICMP protocol exceeds certain number of bytes when packet size in one flow is high (more that 1000 bytes).

protocol happen to one machine in short period of time when there is only Reply packet (no Request packet in last minute).

Finally in Smurf attack the system will generate error when first N (certain number) network flows in ICMP

These rules which mentioned above make this system fast enough to detect those DoS attacks.

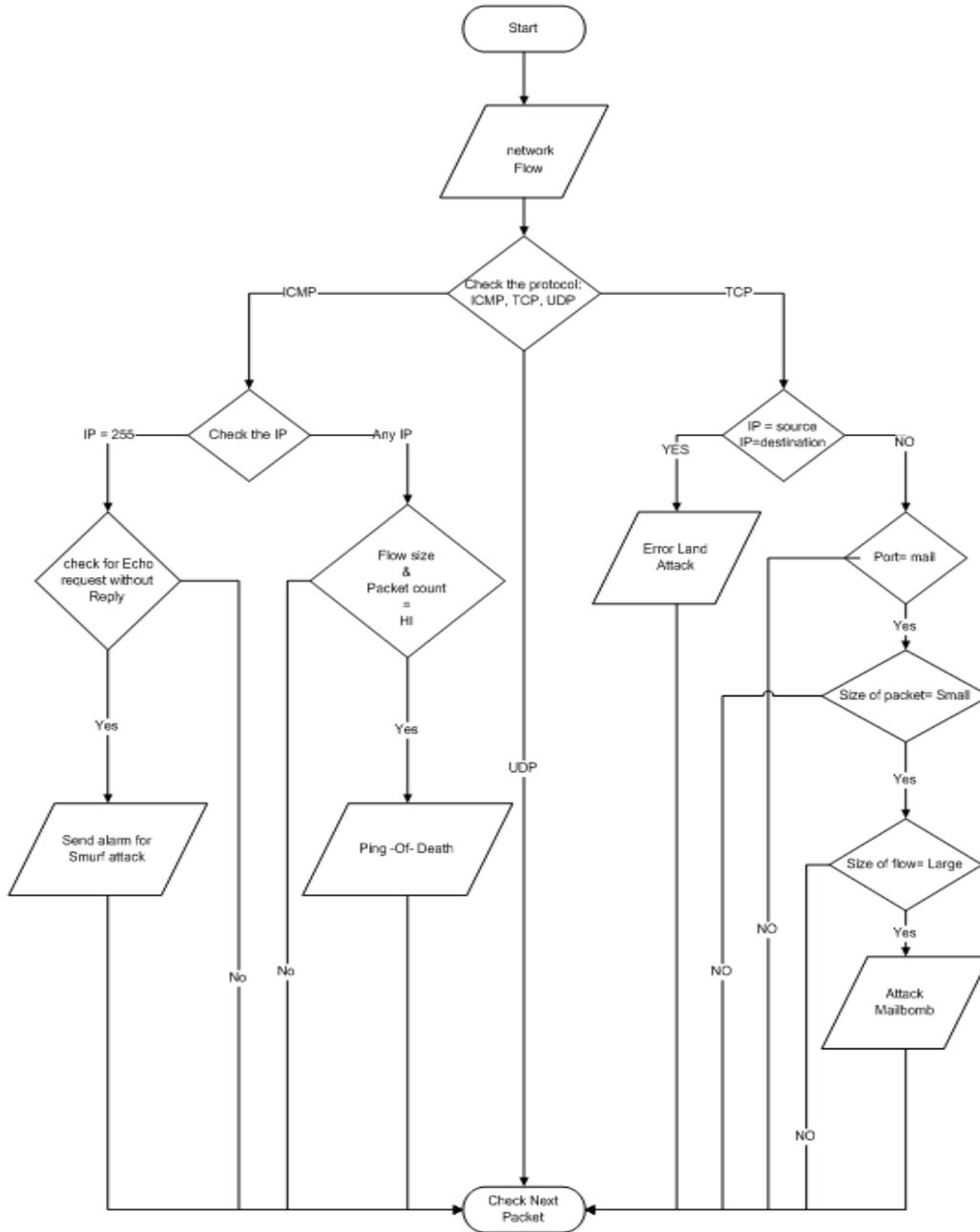


Figure 2 : Fuzzy Decision Tree

## 6 References

[1] Molina, J., and Cukier, M. (2009). Evaluating Attack Resiliency for Host Intrusion Detection Systems. Journal of

Information Assurance and Security, volume 4, no 1, 001-009.

[2] Debar, H., Dacier, M., Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. ACM Computer

Networks: The International Journal of Computer and Telecommunications Networking - Special issue on computer network security, Volume 31, Issue 8, 805–822.

[3] Sundaram, A. (1996). An Introduction to Intrusion Detection. ACM Crossroads - Special issue on computer security, Volume 2, Issue 4.

[4] Anderson, D., Lunt, T. F., Javitz, H., Tamaru, A., Valdes, A. (1995). Detecting unusual program behavior using the statistical component of the next-generation intrusion detection expert system (NIDES), In SRI International Computer Science Laboratory Technical Report SRI-CSL-95-06.

[5] SPADE, Silicon Defense, <http://www.silicondefense.com/software/spice/>.

[6] Mahoney, M. V., and Chan, P. K.,(2001) Detecting Novel Attacks by Identifying Anomalous Network Packet Headers. Florida Institute of Technology, Technical Report, CS-2001-2.

[7] Waizumi, Y., Kudo, D., Kato, N., Nemoto, Y. (2005). A New Network Anomaly Detection Technique Based on Per-Flow and Per-Service Statistics, In Proceedings CIS IEEE, 252–259.

[8] Moore, D., Shannon, C., Brown, D. J., Voelker, G. M., Savage, S. (2006) Inferring Internet Denial-of-Service Activity. ACM Transactions on Computer Systems in 2006, Volume 24, No 2, 115–139.

[9] Frank, J., (2004). artificial intelligence and intrusion detection: current and future directions. In proceedings of the 17th national computer security conference. Volume 10.

[10] Lee, W., Nimbalkar, R. A., Yee, K. Y., Patil, S. B., Desai, P. H., Tran, T. T., Stolfo, S. J.(2000). a data mining and CIDF based approach for detecting novel and distributed intrusions. In proceeding of 3rd international workshop on the recent advances in intrusion detection, Toulouse, France, Volume 1907, 46-65.

[11] Sap, M.N.M., Abdullah, A.H., Srinoy, S., Chimphe, S., Chimphe, W.,(2006). Anomaly Intrusion Detection Using Fuzzy Clustering Methods, Jurnal Teknologi Maklumat, FSKSM, UTM, Jurnal Teknologi Maklumat, Jld. Volume 18, 25-32.

[12] Fries, T. P. (2008). A Fuzzy-Genetic Approach to Network Intrusion Detection. Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation, Atlanta, GA, USA, 2141-2146

[13] CERT Coordination Center, CERT/CC Statistics (1988-2008); <http://www.cert.org/stats/>.

# A Witness Based Approach to Combat Malicious Packets in Wireless Sensor Network

Usman Tariq<sup>a</sup>, Yasir Malik<sup>b</sup>, ManPyo Hong<sup>c</sup> and Bessam Abdulrazak<sup>b</sup>

<sup>a</sup>Department of Information Systems, Al-Imam Mohammed Ibn Saud Islamic University, Riyadh, Saudi Arabia

<sup>b</sup>Department of Informatique, University of Sherbrooke, Sherbrooke, Quebec, Canada

<sup>c</sup>Graduate School of Information and Communication, Ajou University, Suwon, South Korea

**Abstract**—*Limitation in resources like processing power, energy and storage capacity has raised security issues for small embedded devices in ubiquitous environments. Moreover, deployment of tiny devices like sensor nodes in these environment, make it easy for intruder to plant attack node or control over the legitimate node for launching the network attack. Such threat raises the issue to design light weight cryptography algorithms to secure such networks. In this paper, we analyzed the basic threat models in wireless sensor network, and proposed a secure witness based approach to combat malicious packets in wireless sensor network. Our model authenticate legitimate broadcast of control packet like HELLO Packet. and identify the adversary communication between nodes. Once the malicious node is identified the localization algorithm can identify the vulnerable node location and can take appropriate actions. Simulation results show the effectiveness of the proposed model.*

**Keywords:** Sensor networks, encryption, analysis, control packet, authentication, localization

## 1. Introduction

Recently wireless sensor networks (WSN) and its applications got tremendous attention in industry and research community. A WSN is a self-configuring network of small sensor nodes communicating among themselves using radio signals, and deployed in quantity to sense, monitor and understand the physical world [1] The density of sensor network varies from ten to thousands of sensor nodes depending on the application and network requirement. The deployments can vary from global scale for environmental monitoring, habitat to study emergent environments for search and rescue, in factories for condition based maintenance, in buildings for infrastructure health monitoring, in homes to realize smart homes, or even in bodies for patient monitoring.

Sensor nodes are usually less resourceful in term of energy and computational capabilities, therefore the protocols and application in sensor networks should be designed considering these constraints. In addition to functions, procedures and algorithms which involve various OSI layers two most important elements which can affect the performance of WSN includes security and location determination. Both

functions are vital to proper functioning of WSN; without security, data readings can be compromised; without location determination, data readings cannot be spatially associated. The limitation presented by WSN makes them vulnerable and easy to be attacked. In IEEE 802.15.4 standard<sup>1</sup>, sensor nodes are requiring to broadcast HELLO packet to inform their presence to their neighboring nodes. The node which receive HELLO packet may assume that the sender is within its communication radius and can respond to the query. Failure to receiving the HELLO packet after a timeout indicates that either the node is no longer a neighbor or it is no longer available. An adversary with high communication radius for example laptop can make this assumption false by broadcasting high transmission routing information to all or a part of the network or by sending false data which would be important for application for decision making.

This paper focused on spoofed HELLO flooding attacks and data authentication and accusation location identification of nodes. We intended to guarantee the secure and smooth communication i.e. (every legitimate network node should receive relevant and updated packet) and defending adversary nodes inside and outside the network. To do so our scheme will identify the malicious node behavior and its location and later appropriate actions can be taken according to network policy.

This paper is organized as followed, In section 2 we summarize the state of art, section 3 presents proposed witness based control packet authentication scheme, in section 4, we shows the performance of and results. Finally, we conclude the paper in section 5..

## 2. Related Work

Deployments of large number of energy decisive sensor nodes, which will resourcefully enable the information sharing among each other via sharing control and data packets, are defenseless to many kind of routing attacks like HELLO flood. In [1] authors illustrated that every packet delivered by the node is encrypted with a private key and any two sensors share the same common secret. Each time the communication is established between to sensors, algorithm will create a unique key on the fly. It was assumed

<sup>1</sup>IEEE 802.15.4 Standard [www.ieee802.org/15/pub/TG4.html](http://www.ieee802.org/15/pub/TG4.html)

that only reachable neighbor node with decryption key can communicate, which can prevent the adversary attack. The disadvantage of proposed scheme is that any attacker can spoof identities i.e. using HELLO packet, and can begin Sybil attack.

In [4] authors proposed that HELLO flood attack can be prevented using identity verification protocol. Anticipated scheme verifies the bi-directionality of communication signal between two links. If an attack node has a very over the bound link quality, the base station can examine the irregularity by verifying number of trusted neighbors of each node. Approximately all sensing nodes flood the traffic towards base station, which created congestion near centralized sink. In [5] authors recommended that link layer confidentiality and authentication, multi-path routing, and bidirectional link validation can guard sensor nodes from HELLO flood attacks. Considering resource constraint (in processing, communication, energy) of WSN devices, extensively used encryption models and security features used by standard network are not appropriate in WSN. TESLA [8] considers that base station is trusted entity and should be used for distribution the keys between nodes. Receiving nodes has to buffer the packets, which may direct to denial of service (DoS) attack. Furthermore, harmonization is expensive in terms of public key operations. Before transmission starts, sending node has to synchronize with all receivers, which raises the scalability issues. LEAP [13] used dynamic network partitioning, which build extra overhead on energy constraint devices. More over, in case of heterogeneous receiving nodes, proposed algorithm has to use many keys with different discloser delays. SeRLoc [6] utilized a distributed range free localization algorithm and it does not restrict any communication among neighboring sensor nodes. SeRLoc protect well against WSN routing threats such as Hello flood, Sybil attack and wormhole attack.

### 3. Proposed Idea

The motivation of our work is to guarantee the secure routing along with smooth communication i.e. every legitimate network node should receive relevant, updated and accurate packet. Communication paradigm halt if tiny sensor nodes are unavailable due to reasons like spoof hello packet, wormhole attack on control packet and Sybil attack. We categories the attacks as:

**Reactive Information Assembly:** Cleartext communication between sensor nodes and between sensor node and base station is easily accessible and readable by powerful receiver and well designed antenna holder adversary. This may give liberty to attacker to view the network topology [9].

**Node Capture:** Attacker can capture and compromise the sensor node. Occupied sensor device may hinder secret data, which adversary can fetch for further use or it can alter the sensor node with updated malicious code, which force node to act abnormal during internodes communication.

**False Node:** Attacker can add false node to the network, which may broadcast the false information or can deliver the local sensed information to the adversary.

#### 3.1 Witness Based Control Packet Broadcast Authentication

Upon deployment sensor nodes broadcast hello packet to their neighboring nodes to establish link among each other. SNs continue hello packet broadcasting on periodical bases, as sensors is energy constrained devices which are deployed in unattended hostile environments [forest, war field, etc], may die because of damage or power failure.

**Problem Statement:** A laptop class adversary can falsify this postulation by broadcasting control packet (for example, hello packet) with large enough transmission power which can prove to any node in the network that the sender is its neighbor. In sensor network, a hello flood attack uses a single hop broadcast to transmit a message to large number of receivers. An adversary does not always need to be able to construct legitimate traffic in order to use hello flood attack. It can simply rebroadcast overheard by every node in the network. False hello packet may force sensor nodes to forward their packets towards week or possibly dead links.

**Solution:** The notion of witness requires view of witness when interacting with honest approver to be independent of witness used by prover. Even though witness yields weaker security but in several cases witness indistinguishability is sufficient for specific verification task in hand. Hello packet only satisfies the property of one hope communication, just to get knowledge of node's neighbors to establish neighboring table and routing path towards base station. In our scenario, after establishing network, if any set of sensor nodes (SA) receives a hello verification request, instead of reply the to SA, forward the Hello request packet to farthest set of nodes which do not lie in the communication range of sender. If  $\hat{n}$  number of nodes verifies the packet receipt from a unique node ID than consider the Hello packet request as false spoofed request and ignore the request. If adversary continuously use the same spoofed node ID for sending high signal false hello packet, than periodically selected secure nodes will vote to base station to block the malicious node ID.

Sensors are tiny reprogrammable device. Any previously authenticated node may become malicious and start misbehaving by sending rapidly generated hello broadcast packet with in its limited communication radius[7]. We maintain a hash table in sensor node.s database, which keeps record of information of control packet sent by certain node ID. For some time duration, if neighboring nodes get hello packets more than predefined threshold than it will report this abnormal behavior to the base station.

Consider a case when adversary using multiple spoofed ID.s send false control packet to base station stating the misbehaving activity of unique/ multiple nodes. Our scheme

effectively encounters this problem by verifying the encrypted header of packet at base station, as each node share a secret key with base station. Further more, because

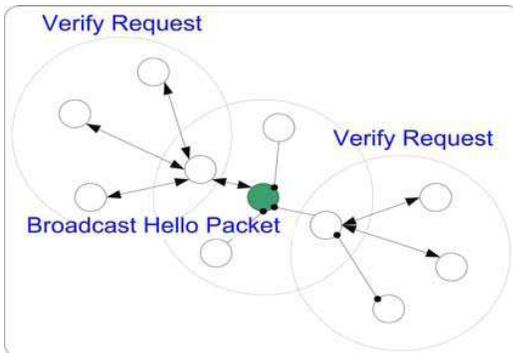


Fig. 1: Broadcast Packet Authentication.

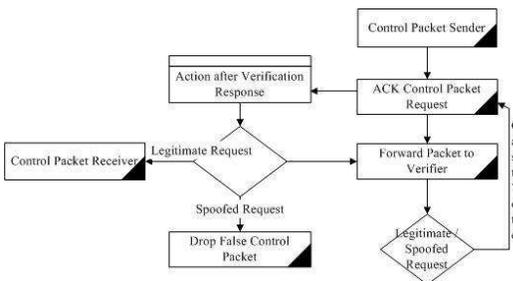


Fig. 2: Broadcast Authentication Procedure.

multiple neighboring nodes will encounter any abnormal activity, and most of them will chose to report to base station because of their homogeneity nature. At base station we also compare the reporting nodes ID.s with the neighboring table of identified possible malicious node. If we found most of reporting nodes do not know reside In side the communication radius of reported node, than we will simply ignore the black listing the particular node request. Consider a case when adversary using multiple spoofed ID.s send false control packet to base station stating the misbehaving activity of unique/ multiple nodes. Our scheme effectively encounters this problem by verifying the encrypted header of packet at base station, as each node share a secret key with base station. Further more, because multiple neighboring nodes will encounter any abnormal activity, and most of them will chose to report to base station because of their homogeneity nature. At base station we also compare the reporting nodes ID.s with the neighboring table of identified possible malicious node. If we found most of reporting nodes do not know reside In side the communication radius of reported node, than we will simply ignore the black listing the particular node request.

Lets consider a scenario where an adversary is generating the false hello flood using several spoofed ID.s. Upon successful

hello packet verification, receiving node add senders ID in its neighboring table of tiny database. Most of protocol restricts neighboring table length to some threshold. By using our proposed method, not only we avoid adversary.s hello broad cast attack but also we hinder the buffer overflow attack on neighboring table and routing table

### 3.2 Data Authentication

Data authentication is a key ingredient of WSN and it plays a vital role in different control processes in network administration for example controlling sensor node duty cycle[3]. Adversary can deploy false nodes in sensor fields and send BS false data which may be essential for decision making process. In this case, during two party communications, data authentication is necessary.

A good cipher should have good randomness, high period; linear span and security against know attacks. We used identity based encryption (IBE), because even in the case of clear text communication, the node must have to learn some basic information before it can communicate with other node. It would be a paradigm shift if the basic information can replace the need of encryption. For example, node M can send message to any near node, with out knowing about its public key (PK). In IBE framework, encryption is always possible. Practically, if receiving node of message does not aware of its private decryption key, it will not be able to decrypt receiving message. However, it will give a strong motivation to node to inquire about the required key. As security point of view, the system can not fully rely on the use of specific information other than PK. As a result, private key computation should be based on global trapdoor information, common to a set of nodes, based on quad zones. This implies, owner of trapdoor (i.e. possibly a key generation center (KGC)), holds the authority to compute the private keys of all sensor field. KGC keep the copies of all keys it generates or has the ability to re-compute the keys of any time stamp and decrypt the communication that it eavesdrops. In other words, KGC can act as key escrow and with additional functionality may behave as IDS watchdog. We can now construct KGC scheme as given under:

**Setup:** KGC runs *RegPairKeyGenerate*, and generate a PK and the related private key. PK is issued as part of system parameters *pmk*. The private key becomes the KGC.s main key *smk*.

**Key Generation:** Nodes performs following steps to get their IBE private key:

- Node execute *RegPairKeyGenerate* and generates *pmk* and related *suk* key.
- Node transmits its identity string ID and PK *puk* to KGC. For optimum security, KGC require a verification of awareness of private key *suk*.

- After user.s identity verification, KGC forms a message, "the PK puk is signing key of node ID" and signs it with private key *smk* using *RegSgn*. The resulting output (Cert) is returned to node.

---

**Algorithm 1** Generating a Key

---

```

1:  Encryption Model (  $x, \pi_s, \pi_p, (K^1, \dots, K^{Nr+1})$  )
2:   $w^0 \leftarrow x$ 
3:  for  $w^r \leftarrow (v_{\pi_p(1)}^r, \dots, v_{\pi_p(m)}^r)$   $r \leftarrow 1$  to  $Nr - 1$ 
4:   $u^r \leftarrow w^{r-1} \oplus K^r$ 
5:  for  $i \leftarrow 1$  to  $m$ 
6:  Do {
7:  do  $v_{(i)}^r \leftarrow \pi_s(u_{(i)}^r)$ 
8:   $w^r \leftarrow (v_{\pi_p(1)}^r, \dots, v_{\pi_p(m)}^r)$ 
9:   $u^{Nr} \leftarrow w^{Nr-1} \oplus K^{Nr}$ 
10: for  $i \leftarrow 1$  to  $m$ 
11: do  $v_{(i)}^{Nr} \leftarrow \pi_s(u_i^{Nr})$ 
12:  $y \leftarrow v^{Nr} \oplus K^{Nr+1}$  }
13: Output ( $y$ )
    
```

---

Here,  $u^r$  is the input to the S-boxes in random  $r.u^{r+1}$   $v^r$  is the output of the S-boxes in round  $r.W^r$  is obtained from  $v^r$  by applying permutation  $\prod_p$ , and than  $u^{r+1}$  is connected from  $v^r$  by x-or-ing the round key  $k^{r+1}$ , which is a round key mixing. In the last round, the  $\prod_p$  permutation is applied. Signature(s): Nodes are now able to sign messages by using generated certificate Cert. Signing node executes *RegSig* on message *Meg* to be signed, using *suk* as signing key.

$$\partial = (puk, ncert, s)produceIDbasedsignature \quad (1)$$

To verify generated signature, execute *RegVer* with PK *pmk* on *ucert* and the message  $\$$ the PK *puk* is signing key of node ID $\$. Verifying node executes *RegVer* with PK *puk* on signature and message. If verification return void, the output "false" Other wise return "true".$

Key Distribution: After establishing the link key through multi paths, we used multi path key enforcement method to enhance the security among nodes. This technique is very resilient against node capture attack. The only draw back to adopt such technique is network communication overhead [2]. Nodes discover different paths to route data by exchanging hello packets between them. The more routes discover between two nodes, the more security multi path key reinforcement provides between each link.

### 3.2.1 Location Information Processing

Research community did splendid tasks to accurately precise location of any node resided in the sensor field. The finite objective of localization algorithm is to gather measurements or related angels between one or many nodes

and multiple anchors in order to estimate accurate location estimation. When a node achieves its position evaluation, it may work as new anchor and help neighboring nodes to estimate their position. In our scheme, we focused on non centralized algorithm which distributes the computational load fairly athwart the network nodes, helping all nodes to save computational power available to all network. We did not use signal measurement to infer range. For better understanding, consider a node of interest  $x$  resided near  $Z$  anchor nodes with harmonizes  $(a_n, b_n)$ , where  $n = (1, \dots, Z)$ . The anchor nodes commune and spread their coordinate points to node of interest. Upon receiving the anchor location information, the node of interest guesstimates its location as the barycenter of given points. The coordinates are estimated as followed:

$$(\hat{a}, \hat{b}) = \left( \frac{1}{z} \sum_{n=1}^z a_n, \frac{1}{N} \sum_{n=1}^z b_n \right) \quad (2)$$

The barycenter algorithm [12] is localized, disseminated scheme, in which a node of interest needs to be in the neighborhood of  $Z$  anchor nodes. Note that barycenter algorithm is applicable for single and multi hop setups, but the accuracy of the location estimation will be minimized as the reach ability radii of sensor field nodes increases.

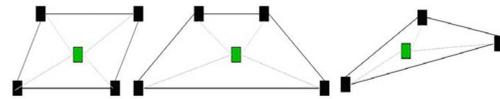


Fig. 3: Multiple scenarios of location estimation using Barycenter Algorithm.

## 4. Simulation Environment

To verify the effectiveness of our proposed scheme we have simulated our scheme in NS-2<sup>2</sup>. We have assumed identical sensors sensitivities where coverage depends only on geometrical distances from sensors also we have a centralized control server where nodes are connected with each other in peer-to-peer fashion which leads to connectivity with base station. The distance measurement range of the nodes is equal to the communication range. Each node is aware of its two hope neighboring node all the time to avoid data from adversary. Table.1 shows the list of parameters we adopted: Figure 3 illustrates that he amount of computational energy consumed by a security function (cryptography) on a given microprocessor is primarily determined by the processor power consumption, the processor clock frequency, and the number of clocks needed by the processor to compute the security function. The cryptographic algorithm and the efficiency of the software implementation determine the number of clocks necessary to perform the security function.

<sup>2</sup><http://www.isi.edu/nsnam/ns/>

Table 1: Simulation Parameters.

Parameter	Values	
Area size of simulation	370m * 60m	
Total number of nodes in simulation	550	
Total time for simulation	100s	
Nodes transmission range	15m	
Packet or frame error rate	Relative delivery rate	
Data rate	Decided by graph	
Data Packet Size	70 Bytes	
Traffic type	Constant bit rate	
RREQ packet Size	36	
RREP packet size	40	
Inter-Packet transmission delay	Decided by the graph	
Beacon Time period	4 sec.	
Energy Consumption	Idle	5µJoules/s
	Sense	300Joules
	Transmit	203Joules
	Receive	212 Jouls
Battery Size	0.06 mAh	

For cryptographic processing, we assume that energy consumption cannot be significantly reduced via a reduction in clock frequency, since a corresponding reduction in voltage would be required; a capability not ideally available in today's embedded processors.

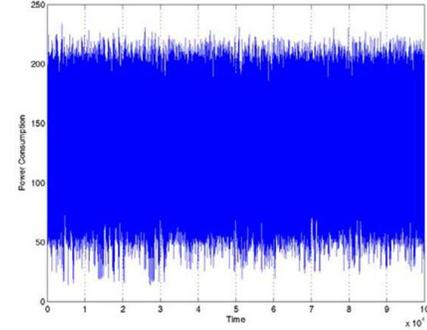
$$LEE = \frac{\text{Encountered Inaccuracy During Attack}}{\text{Habitual Inaccuracy}} \times 100 \quad (3)$$

Figure 4 shows the impact of reach ability radius relative to coverage radius. It's clear that position estimation accuracy improved with node density, for example, for 550 nodes in the sensor field location error is smaller than the case of 100 nodes. Accurate location awareness at quad and whole sensor field can help prevent selfish behavior, Sybil and wormhole (tunneling the broadcast control packets) attacks. Figure 5 demonstrates the association between location error estimation ( $LEE$ ) and the  $R$  factor while the intensity of malicious nodes generation wormholes is equivalent to 25% of the entire set of sensor nodes in sensor field. The figure illustrates that wormholes influence localization outcome of proposed scheme; where  $R$  varies from 5m to 12 m. Since when  $R = 5$  and 12m, the outcome of localization accuracy, which also perform as the denominator in the equation of  $LEE$  is not as excellent as when  $5 = 5-12m$ , so it influence that the defection of wormhole, which is calculated with  $LEE$ , is not as large as when  $r = 3, 11m$ .

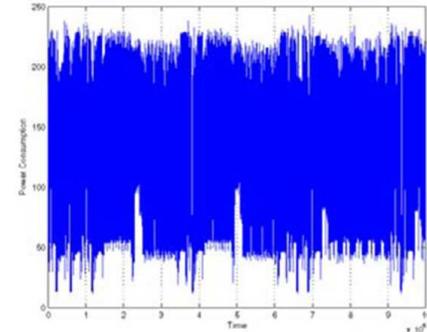
We used radius of signal which has its center at the mean and enclose half of the insight of random vector coordinate evaluation.  $d$  is the measure of ambiguity in position estimation relative to its mean  $M^d$ . If position estimator is impartial, position error probability (PEP) is a quantifier of the estimator ambiguity relative to accurate node of interest position. If degree of vector is bounded by  $O$ , then the probability of  $1/2$ , a particular estimate is within a distance of  $O + PEP$  from the exact position.

$$\frac{1}{2} = \iint_x P\hat{d}(\zeta) d\zeta_1 d\zeta_2 \quad (4)$$

Here  $P\hat{d}(\zeta)$  is a probability density function of vector estimator  $\hat{d}$  and the incorporation area is defined as  $x = \{ \zeta : | \zeta - M \{ \hat{d} \} | \} \leq PEP$  where  $\zeta$  is the distance between the estimated location and actual location. Errors are larger for sparse network densities.



(a) Series a



(b) Series b

Fig. 4: Energy consumption in key distribution.(a) With constant frequency (b) With variable frequency.

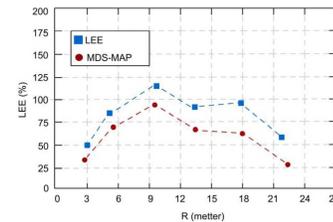


Fig. 5: Effect of number of nodes on Location Estimation Error

From figure 5 we can predict that in the presence of few attack nodes, the localization error may increase drastically. To minimize the localization estimation error percentage encountered because of wormhole we merge wormhole detection and defense mechanism into the localization scheme. The energy reduction achieved via our scheme in contrast to the standard packet with source/destination MAC addresses. To collect a packet, we suppose that the working out takes

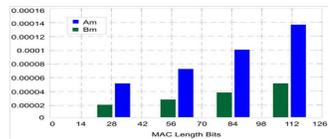


Fig. 6: Total Energy Consumption

on typical 55 commands and additional calculations such as deciphering consumes additional 34 commands, for an overall cost of 12 nJ; packets are 56 bits in size. Energy consumption grows as the packet size expands. Nevertheless, because of the diminutive calculation cost of 1.2 nJ per instruction, the outlay only augments by 5.5 nJ, considering four extra instructions/byte. This implies, in proposed method, the packet requirements to rise only an additional 12.1 kb prior to calculation rate holds the similarity with the rate of broadcasting MAC addresses.

## 5. Conclusion

In this paper we have presented a simple but efficient mechanism which presents the collateral damage effect caused by control packet flood and clear text communication among nodes. To launch a packet flood an adversary does not always need to be able to construct legitimate traffic in order to use packet flood attack. It can simply rebroadcast overhead by every node in the network.

Proposed IBE encryption model portray, good randomness, high period; linear span and security against know attacks. We believe that it would be paradigm shift if the basic information can replace the need of encryption. This technique is very resilient against node capture attack. The only drawback to adopt such technique is network communication overhead but the more routes discover between two nodes, the more security multi path key reinforcement provides between each link.

In our proposed location information processing scheme, we focused on non centralized algorithm which distributes the computational load fairly athwart the network nodes, helping all nodes to save computational power available to all network. Note that proposed algorithm is applicable for single and multi hop setups, but the accuracy of the location estimation will be minimized as the reach ability radii of sensor field nodes increases. Overall, simulation results show the valuable feasibility of our broadcast control packet authentication, and encryption scheme for embedded architectures. We observed less energy consumption, longer life of network, and better packet authentication. Due to versatile nature of WSN, where authentication and location aware processes come into play, it is and will be an important research field. In future, we have planned to evaluate our encryption scheme to be compared with other scheme like SEAL 3.0 [10] and TEA [11], in terms of software implementation, processing and energy consumption. We

look forward for more information on the strengths of the algorithm.

## References

- [1] S. H. A Hamid, "Defense against lap-top class attacker in wireless sensor network," in *Advanced Communication Technology, ICACT 2006. The 8th International Conference*, vol. 1, 2006.
- [2] R. Anderson, H. Chan, and A. Perrig, "Key infection: smart trust for smart dust," in *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, oct. 2004, pp. 206 – 215.
- [3] C. Boyd and A. Mathuria, "Key establishment protocols for secure mobile communications: A selective survey," in *Information Security and Privacy*, ser. Lecture Notes in Computer Science, C. Boyd and E. Dawson, Eds. Springer Berlin / Heidelberg, 1998, vol. 1438, pp. 344–355, 10.1007/BFb0053746. [Online]. Available: <http://dx.doi.org/10.1007/BFb0053746>
- [4] V. C. Giruka, M. Singhal, J. Royalty, and S. Varanasi, "Security in wireless sensor networks," vol. 8, no. 1, 2008, pp. 1–24.
- [5] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *In First IEEE International Workshop on Sensor Network Protocols and Applications*, 2002, pp. 113–127.
- [6] L. Lazos and R. Poovendran, "Serloc: secure range-independent localization for wireless sensor networks," in *Proceedings of the 3rd ACM workshop on Wireless security*, ser. WiSe '04. New York, NY, USA: ACM, 2004, pp. 21–30. [Online]. Available: <http://doi.acm.org/10.1145/1023646.1023650>
- [7] S. Lindsey and C. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 3, 2002, pp. 3–1125 – 3–1130 vol.3.
- [8] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The tesla broadcast authentication protocol," 2002.
- [9] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Commun. ACM*, vol. 47, pp. 53–57, June 2004. [Online]. Available: <http://doi.acm.org/10.1145/990680.990707>
- [10] P. Rogaway and D. Coppersmith, "A software-optimized encryption algorithm," *JOURNAL OF CRYPTOLOGY*, 1997.
- [11] D. J. Wheeler and R. M. Needham, "Tea, a tiny encryption algorithm," in *Fast Software Encryption*, 1994, pp. 363–366.
- [12] S.-S. Yu, J.-R. Liou, and W.-C. Chen, "Computational similarity based on chromatic barycenter algorithm," *Consumer Electronics, IEEE Transactions on*, vol. 42, no. 2, pp. 216 –220, may 1996.
- [13] S. Zhu, S. Setia, and S. Jajodia, "Leap: efficient security mechanisms for large-scale distributed sensor networks," in *Proceedings of the 10th ACM conference on Computer and communications security*, ser. CCS '03. New York, NY, USA: ACM, 2003, pp. 62–72. [Online]. Available: <http://doi.acm.org/10.1145/948109.948120>

# Detecting Undetectable Metamorphic Viruses

Sujandharan Venkatachalam<sup>1</sup> and Mark Stamp<sup>1</sup>

<sup>1</sup>Department of Computer Science, San Jose State University, San Jose, California, USA

**Abstract**—*Signature-based detection provides a relatively simple and efficient method for detecting known viruses. At present, most antivirus systems rely primarily on signature detection.*

*Metamorphic viruses are potentially one of the most difficult types of viruses to detect. Such viruses change their internal structure, which provides an effective means of avoiding signature detection. Previous work has shown that a specific and straightforward metamorphic engine can generate viruses for which reliable detection using “static analysis” is NP-complete. In this paper, we implement this metamorphic engine and show that, as expected, popular antivirus scanners fail to detect the resulting viruses. Finally, we analyze these same viruses using a previously developed detection approach based on hidden Markov models (HMM). This HMM-based detector, which by most definitions would be considered a static approach, easily detects the viruses.*

**Keywords:** malware, metamorphic, static analysis, hidden Markov models

## 1. Introduction

Since the advent of malware, virus creation techniques and detection methodologies have evolved in an ongoing “arms race” [3]. Virus writers want their handiwork to evade detection and since signature detection is the most popular, considerable effort has gone towards hiding or obfuscating signatures.

Metamorphic viruses rely on code morphing to prevent signature detection [12]. While metamorphism can effectively obfuscate signatures, the paper [16] shows that metamorphic viruses produced by the hacker community are generally not very effective, and of those tested, even the most highly metamorphic are detectable using machine learning techniques—specifically, hidden Markov models (HMMs).

The authors of [4] claim to have obtained the following intriguing result:

In particular, we prove that reliable static detection of a particular category of metamorphic viruses is an NP-complete problem. Then we empirically illustrate our result by constructing a practical obfuscator which could be used by metamorphic viruses in the future to evade detection.

Note that the authors of [4] appear to have the usual concept of “static analysis” in mind, as indicated by the following quote:

Here, static analysis is conceived as a process whose goal is to extract the semantics of a given program without any help of code execution.

It is well known that, in theory, metamorphic virus writers have an insurmountable advantage [6], [10], [17]. However, it is a curious fact that relatively few metamorphics have appeared in the wild and, furthermore, very few of these provide strong metamorphism and, in any case, none has proven particularly difficult to deal with in practice. This suggests that there are many practical difficulties for virus writers to overcome if they want to take advantage of metamorphism. When viewed in this light, it might seem that the most impressive result in [4] is its claim to provide a simple and practical design for a metamorphic generator that yields viruses that cannot be reliably detected using “static analysis.”

In this paper, we have implemented a stand-alone metamorphic generator that satisfies the conditions in [4] and we have applied it to selected viruses. We show that, as expected, the resulting morphed viruses are not detected using popular signature-based antivirus software. However, we also show that these metamorphic viruses are, in fact, easily detected using the machine learning technique developed in [16]. Note that the detection method in [16] would generally be considered a static approach, since it only relies on extracted opcode sequences—no code execution or emulation is used. Indeed, this would seem to fit the informal description of static analysis given in [4], as quoted above.

The work presented in this paper demonstrates that metamorphic viruses generated following [4] would not be particularly difficult to detect in practice, even if we restrict ourselves to static analysis, as the term is generally understood. The loophole here is that the formal definition of “static analysis” in [4] is extremely narrow—much narrower than suggested by the informal discussion in the paper itself.

The organization of this paper is as follows. In Section 2 we briefly discuss the evolution of metamorphic viruses. Then in Section 3 we consider elementary code obfuscation techniques used in metamorphic generators, and in Section 4 we briefly discuss the use of HMMs for metamorphic detection. In Section 5 we provide details of the metamorphic generator that we have developed—a generator that satisfies the conditions given in the paper [4]. Section 6 summarizes our experimental results. Finally, Section 7 concludes the paper.

## 2. Evolution of Metamorphic Viruses

Early on in the Titanic struggle between the forces of good and evil (code, that is), signatures became the preferred means of detecting malware. Predictably, virus writers reacted by developing new techniques designed to evade signature detection. Here, we briefly outline the evolution of virus development and the parallel history of virus detection.

As the name indicates, encrypted viruses try to bypass virus detection by self-encryption. The code encryption implemented in such viruses effectively hides the signature of the underlying virus. However, the virus body must be decrypted before it can execute, and the decryption code is susceptible to signature scanning [3].

Like encrypted viruses, polymorphics try to bypass detection by self-encryption. However, unlike encrypted viruses, these viruses mutate the decryptor code, making scanning much more challenging [12]. Figure 1 illustrates different variants of a polymorphic virus. Note that the decrypted body is the same in each case.

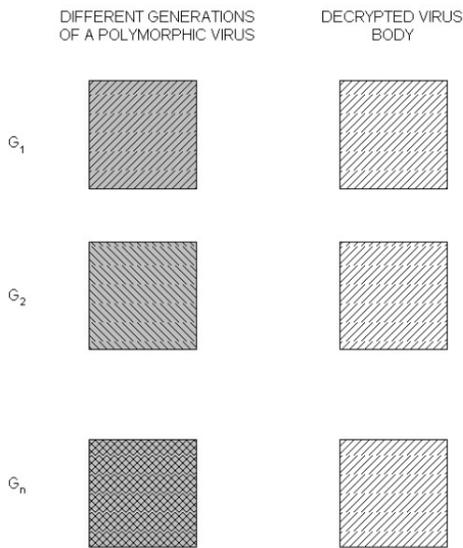


Fig. 1: Polymorphic viruses [12]

Polymorphic viruses are often detected using emulation—if the emulated code is actually a virus, it will eventually decrypt itself, at which point standard signature scanning can be applied [3].

Metamorphic viruses modify their entire code in such a way that each copy is functionally the same, but structurally different [3]. If the copies are sufficiently different, no common signature will be present. Figure 2 illustrates different generations of a metamorphic virus. Note that the code structure differs in each case, yet the viral copies all have the same function.

It is intuitively clear that well designed metamorphic code cannot be effectively detected via signature-based methods—a fact that is made rigorous in [4]. However, it has previously

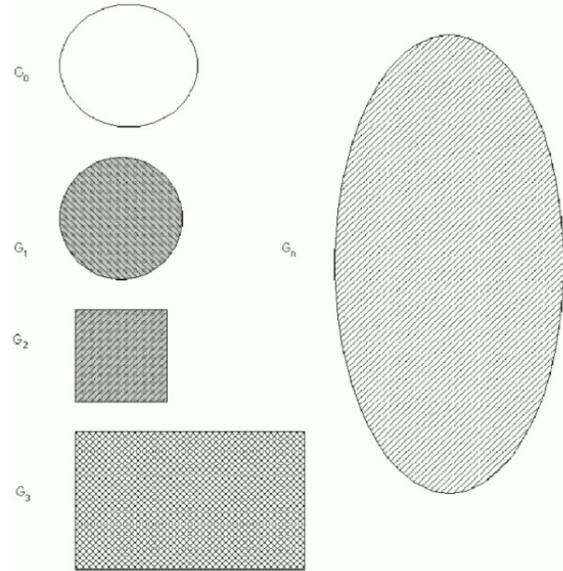


Fig. 2: Metamorphic viruses [12]

been shown that metamorphic viruses produced by the hacker community can be detected using machine-learning techniques [16]. Below, we have more to say about the approach used in [16].

## 3. Code Obfuscation

Metamorphic viruses use one or more obfuscation techniques to produce structurally different versions of the same virus, while not altering the function of the code. The primary goal of the obfuscation is to avoid signature detection—if the viruses are sufficiently different, no common signature will exist and, ideally, no fixed signature will detect any significant percentage of the viruses. Below, we briefly discuss a few of the most common code morphing techniques. The code obfuscation techniques implemented in several hacker-produced metamorphics are summarized in Table 1.

Table 1: Code obfuscation techniques [4]

	Evol 2000	Zmist 2001	Zperm 2000	Regswap 2000	MetaPHOR 2001
Substitution				X	
Permutation	X	X			X
Garbage code	X	X			X
Variable substitution	X	X		X	X
Alter control flow		X	X		X

Inserting garbage instructions between useful code blocks is a simple obfuscation technique used in all of the virus generators listed in Table 1. Garbage instructions do not alter the functionality but will tend to increase the size of the code. Viruses that contain garbage instructions are harder to

detect using signatures since these instructions tend to break up predetermined signatures.

Instruction reordering is another common metamorphic technique. In this method, the instructions in the virus code are shuffled, with the control flow adjusted (via jump statements, for example) to make the code execute in the appropriate order. Thus, the instructions are reordered within the code without altering the actual control flow. This method can also effectively break signatures. However, if too many jump instructions are inserted, this could be used as a heuristic for detecting malware. Figure 3 shows an example of code reordering. Subroutine reordering is a special case of code reordering. Reordering subroutines in the virus does not change the control flow but could make signature detection more difficult.

Instruction interchange is another useful obfuscation technique. In this method, instructions are replaced with equivalent instructions. Then metamorphic versions of a given base virus will have different patterns of opcodes that perform the same function.

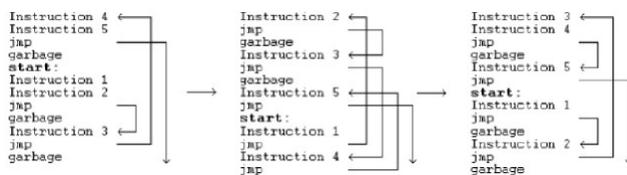


Fig. 3: Code reordering [13]

Register swapping, where different registers are used for equivalent operations, is a simple special case of interchanging instructions. Again, the idea is to change the opcode pattern and thereby bypass signature detection. This technique was the primary means of obfuscation used in one of the first metamorphic viruses, W95/Regswap. Register swapping is a particularly weak form of metamorphism, since it is subject to signature detection using wildcards [13].

Of the hacker-produced metamorphic generators tested in [16], the most advanced is the Next Generation Virus Construction Kit (NGVCK) [14]. Table 2 provides examples of code from NGVCK viruses.

## 4. HMMs for Virus Detection

Machine learning techniques have been successfully applied to the problem of detecting metamorphic viruses [16]. These techniques extract statistical information from training data and the resulting model can then be used to test any input file for similarity to the training data.

A hidden Markov model (HMM) is a state machine that attempts to model a Markov process. The Markov process is hidden, in the sense that it cannot be directly observed. The actual observations are probabilistically related to the hidden process. In the context of metamorphic viruses, an HMM is trained to detect a specific metamorphic family. The

training data consists of a sequence of opcodes derived from viruses, all of which were produced by a single metamorphic engine. Once the model is trained, it can be used to score an unknown file, using an extracted opcode sequence, to determine its similarity to the metamorphic family. For more details on the use of HMMs for metamorphic virus detection, see [16]; for related work involving profile HMMs see [2]; for additional background on HMMs in general, see [11] or [9].

## 5. Metamorphic Generator

We have implemented a metamorphic virus generator that satisfies the conditions given in [4]. Recall that the paper [4] provides a rigorous proof that viruses generated using their approach cannot be efficiently detected using static analysis (as they define the term). Next, we briefly discuss the details of our metamorphic generator, which implements the practical generator given in [4].

A seed virus is input to our metamorphic generator. The seed virus assembly code is split into small blocks, which are then reordered using conditional jump instructions and labels. The number of instructions in each block is variable and for the experiments described here is set to an average value of six. The virus code is split into blocks, respecting the conditions given in [4], namely, code blocks cannot end with a label, jump, or a NOP instruction. A precondition on the seed virus is that the entire code must appear in the code section of the assembly file, which implies that viruses that hide code in their data section cannot be used.

After splitting the code into blocks, the blocks are randomly shuffled. Then labels are inserted and conditional jump instructions are used so as to maintain the original control flow. Optionally, garbage code insertion is applied for additional code obfuscation. In summary, our metamorphic engine performs the following steps:

- 1) Input a base virus file
- 2) Blocks are identified subject to the following conditions:
  - a) The first and last block of the code are fixed
  - b) The last instruction of a block is not a label, jump, or NOP
- 3) Blocks are randomly permuted and labels and conditional jumps are inserted
- 4) Garbage instructions are randomly inserted according to a threshold value
- 5) Write the morphed output file

The garbage insertion is optional and the amount of garbage inserted is adjustable. The garbage instructions include various copy instructions and opaque predicates, with the garbage inserted between pairs of code blocks, after the block shuffling is completed. Our generator has been successfully tested with several virus families. A typical test

Table 2: Code obfuscation in NGVCK

Base Version	Morphed Version 1	Morphed Version 2
call delta	call delta	add ecx, 0031751B ; junk
delta: pop ebp	delta: sub dword ptr[esp], offset delta	call delta
sub ebp, offset delta	pop eax	delta: sub dword ptr[esp], offset delta
	mov ebp, eax	sub ebx, 00000909 ; junk
		mov edx, [esp]
		xchg ecx, eax ; junk
		add esp, 00000004
		and ecx, 00005E44 ; junk
		xchg edx, ebp
HEX equivalent:	HEX equivalent:	HEX equivalent:
E8000000005D81ED05104000	E800000000812C2405104000588BE8	*812C240B104000*8B1424*83C404*87EA

case is discussed in the next section; for more examples, see [15].

Next, we applied an HMM virus detection technique to our metamorphic viruses. Here, we mimic the training and scoring methodology used in [16]. To train an HMM model, 200 distinct metamorphic copies of a given seed virus were created using our metamorphic engine. The metamorphic engine generates ASM files, each of which yields executable code having the same functionality as the seed virus. These 200 files were assembled using the Borland Turbo TASM 5.0 assembler and linked using the Borland Turbo TLINK 7.1 linker to produce EXE files. The EXE files thus obtained were then disassembled using IDA Pro [7] and opcode sequences were extracted. The steps performed in preparing the test data are summarized in Figure 4.



Fig. 4: Test data preparation

Note that disassembled files obtained from EXE files were used for training and testing. Consequently, our training and testing is realistic in the sense that only EXE files would be available to antivirus software.

We performed 5-fold cross validation, that is, we split the 200 metamorphic virus files into 5 subsets of 40 viruses each. From among these five subsets, four were used for training and the remaining one was reserved to test the trained HMM model. This process was repeated five times, once for each distinct 4-subset collection of morphed files. In each case, 40 metamorphic files were scored along with 40 “normal” files. For the normal files, we used Cygwin utility files, since these files were also used as the representative normal files in [16] and [8].

## 6. Experimental Results

For the test case considered in this paper, the Next Generation Virus Creation Kit (NGVCK) was used to create the seed viruses. Other viruses were considered, with equally

strong results obtained in each case; see [15] for more details on these other experiments.

In each case, popular antivirus scanners could detect the seed virus, but not the viruses produced by our metamorphic generator. That is, our metamorphic generator is able to successfully bypass signature detection, as expected. However, regardless of the seed virus used, the HMM engine was able to distinguish the morphed viruses from normal code, as discussed below.

Virus creation, analysis and testing experiments were conducted using the platform and tools listed in Table 3. Again, the procedure followed here follows that used in [16].

Table 3: Experimental setup [15]

Platform:	Windows XP/VMware
Language:	Perl5
Disassemblers:	OllyDbg v1.10 and IDA Pro 4.9
Assembler:	Borland Turbo Assembler 5.0
Linker:	Borland Turbo Linker 7.1
Virus generators:	MPCGEN (Phalcon/Skism Mass Code Generator) G2 (Generation 2 Virus Generator) VCL32 (Virus Creation Lab for Win32) NGVCK (Next Generation Virus Creation Kit)
Virus scanners:	Avast Home Edition 4.8 McAfee Antivirus 2009

As mentioned above, the seed viruses were detected by commercial antivirus software. For example, Figure 5 shows a screenshot of the security alert displayed by McAfee antivirus when it scanned one of our seed viruses.

Next, we present the results from one typical experiment. In this example, we used our engine to generate 200 metamorphic variants of an NGVCK seed virus. The parameters were set to generate variants with a threshold of two garbage instructions. Snippets of code from two of the resulting metamorphic variants appear in Figure 6.

The metamorphic viruses were then assembled and the resulting morphed executables scanned using popular antivirus scanners by McAfee and Avast [1]. As expected, these scanners were not able to identify the morphed executables as viruses.

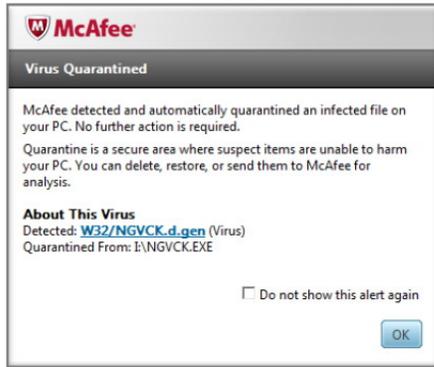


Fig. 5: Seed virus scanned with McAfee

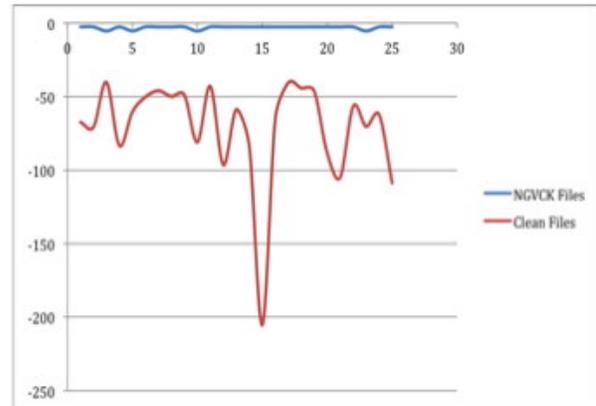


Fig. 7: HMM scores

```

ngvck98.asm                               ngvck941.asm
Win32.NGVCK by SnakeByte                  Win32.NGVCK by SnakeByte
This Virus is created with                This Virus is created with
the Next Generation VCL by SnakeByte     the Next Generation VCL by SnakeByte
to get a copy of this Kit                to get a copy of this Kit
check www.kryptocrew.de/snakebyte/       check www.kryptocrew.de/snakebyte/

586p                                       586p
model flat                                  model flat
jumps                                       jumps
pushd dword ptr [ebp+MapAddress]          pushd dword ptr [ebp+MapAddress]
pop ebx                                     pop ebx
add ebx, [ebx+3Ch]                          add ebx, dword ptr [ebp+MapAddress]
mov ebx, dword ptr [ebp+CheckSum]          mov ebx, dword ptr [ebp+CheckSum]
mov dword ptr [ebx+0h], ebx                 mov dword ptr [ebx+0h], ebx
NcCheckSum                                  NcCheckSum
mov ecx, dword ptr [ebp+IntCounter]        mov ecx, dword ptr [ebp+IntCounter]
odd ecx                                      odd ecx
jnp labelblock51                            jnp labelblock51
cr cr, 0                                     cr cr, 0
odd cx, 0                                    odd cx, 0
labelblock55:                               labelblock55:

```

Fig. 6: Sample metamorphic code

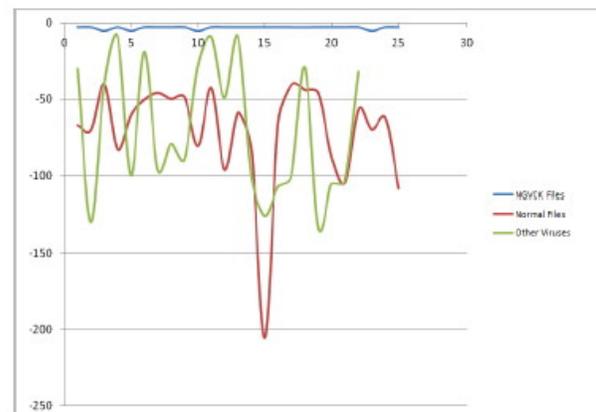


Fig. 8: HMM scores including non-family viruses

This 5-fold cross validation, HMM models were trained using this set of 200 metamorphic viruses. The number of distinct observation symbols (i.e., opcodes) ranged from 40 to 42 and the total number of observations ranged from 41,472 to 42,151. The resulting model was then used to score 40 viruses and 40 normal files. A typical HMM score graph appears in Figure 7, where the scores are given as log likelihood per opcode (LLPO). That is, the scores were computed based on log odds, then normalized on a per opcode basis. In every case, a threshold could easily be set that would provide flawless detection, that is, no false positives or false negatives would occur. In fact, the score differences are quite large given that the scores are computed on a per opcode basis.

Figure 8 shows a similar graph as that which appears in Figure 7, but with 40 additional, non-family viruses included. Note that some of the non-family viruses score significantly higher than any of the normal files. However, we can still set a threshold that results in no false positives or false negatives. The fact that other viruses have relatively high scores is not too surprising, and might be considered beneficial, since we could adjust the threshold and thereby detect some additional related malware.

## 7. Conclusions and Future Work

In this paper, we analyzed a metamorphic generator satisfying the conditions given in [4]. The paper [4] provides a rigorous proof that such viruses cannot be efficiently detected using “static analysis,” according to their definition of the term. As expected, these metamorphic viruses are not susceptible to signature detection. However—and perhaps surprisingly—these viruses are detected via a machine learning approach. Specifically, we trained HMM models to detect such viruses. While the work presented here does not directly contradict [4], it does call into question the utility of relying on such a narrow definition of “static analysis” since, by most accounts, our HMM approach would be considered a static technique.

At this point, a natural question to ask is whether a practical metamorphic generator can be produced that will evade both signature detection and our HMM-based detector. The paper [5] was a first attempt to settle this question while [8] shows conclusively that a practical metamorphic generator can evade both signature detection and the HMM based approach used in this paper. However, it is not yet entirely clear where, in a practical sense, the ultimate balance of

power lies between metamorphic virus writers and detection.

At first blush, the results in [4] seem to prove that in a very practical and real sense, metamorphic virus writers have an insurmountable advantage, at least from the perspective of static analysis. However, the results in this paper show that the reality of the situation is considerably more nuanced.

## References

- [1] Avast Antivirus, <http://www.avast.com/>
- [2] S. Attaluri, S. McGhee and M. Stamp, Profile hidden Markov models and metamorphic virus detection, *Journal in Computer Virology*, vol. 5, no. 2, May 2009, pp. 151–169
- [3] J. Aycock, *Computer Viruses and Malware*, Springer, 2006
- [4] J. Borello and L. Me, Code obfuscation techniques for metamorphic viruses, *Journal in Computer Virology*, vol. 4, no. 3, August 2008, pp. 211–220
- [5] P. Desai, Towards an undetectable computer virus, Master's project report, Department of Computer Science, San Jose State University, 2008, [http://www.cs.sjsu.edu/faculty/stamp/students/Desai\\_Priti.pdf](http://www.cs.sjsu.edu/faculty/stamp/students/Desai_Priti.pdf)
- [6] E. Filiol, Metamorphism, Formal grammars and undecidable code mutation, *World Academy of Science, Engineering and Technology*, vol. 26, 2007
- [7] IDA Pro, <http://www.hex-rays.com/idapro/>
- [8] D. Lin and M. Stamp, Hunting for undetectable metamorphic viruses, to appear in *Journal in Computer Virology*
- [9] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, vol. 77, no. 2, 1989
- [10] D. Spinellis, Reliable identification of bounded-length viruses is NP-complete, *IEEE Transactions on Information Theory*, vol. 49, no. 1, January 2003, pp. 280–284
- [11] M. Stamp, A revealing introduction to hidden Markov models, January 2004, <http://www.cs.sjsu.edu/faculty/stamp/RUA/HMM.pdf>
- [12] P. Szor and P. Ferrie, Hunting for metamorphic, Symantec Security Response, <http://www.symantec.com/avcenter/reference/hunting.for.metamorphic.pdf>
- [13] P. Szor, *The Art of Computer Virus Defense and Research*, Symantec Press, 2005
- [14] VX Heavens, <http://vx.netlux.org/>
- [15] S. Venkatachalam, Detecting undetectable computer viruses, Master's project report, Department of Computer Science, San Jose State University, 2010, [http://www.cs.sjsu.edu/faculty/stamp/students/venkatachalam\\_sujandharan.pdf](http://www.cs.sjsu.edu/faculty/stamp/students/venkatachalam_sujandharan.pdf)
- [16] W. Wong and M. Stamp, Hunting for metamorphic engines, *Journal in Computer Virology*, vol. 2, no. 3, December 2006, pp. 211–229
- [17] P. Zbitskiy, Code mutation techniques by means of formal grammars and automatons, *Journal in Computer Virology*, vol. 5, no. 3, August 2009, pp. 199–207

# A Methodology to Identify Complex Network Attacks

Lisa Frye<sup>1,2</sup>, Liang Cheng<sup>2</sup>, and Randy Kaplan<sup>1</sup>

<sup>1</sup>Computer Science Department, Kutztown University, Kutztown, PA, USA

<sup>2</sup>Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, USA

**Abstract**—*Networks are attacked on a daily basis. Identifying these attacks is a crucial task for network managers. Tools exist to assist in this task. One of the more common tools used is an Intrusion Detection System. These systems may fall short in identifying all attacks that have occurred on a network. Often an attack consists of a sequence of simple attacks. These complex attacks are more difficult to identify. The proposed methodology identified a family of complex attacks to aid in the understanding of these attacks to improve their detection. A reusable representation of these attacks was developed using ontology. Lastly, an algorithm was developed to utilize the ontological representations to extend the knowledge of complex attacks and allow for better detection of such attacks by taking advantage of the advanced reasoning inherent in ontology.*

**Keywords:** Computer network security, Nonmonotonic reasoning, Security, Site security monitoring

## 1 Introduction

Networks are common in everyday life. Networks consist of many nodes with a plethora of data traversing the network daily. This data consists of legitimate data for professional and personal purposes, as well as undesirable data, often injected willingly by users intending to attack the network or a node on the network. The intention of the attack may vary from the user that just wants to prove the attack can occur, to more malicious attacks intending to disrupt services or destroy data. It is important to be able to identify when an attack occurs, stop any damage being caused by the attack and implement additional security measures to prevent such attacks in the future.

One tool often used to identify network attacks is an Intrusion Detection System (IDS) [1]. There are many different types of IDSs, ranging from host-based (HIDS) [1], residing on a specific host watching for attacks against that host, to network-based (NIDS) [1], deployed on the network to monitor network resources and identify network attacks. Organizations typically use a combination of network-based and host-based IDSs.

IDSs can also be classified by how they detect attack attempts. A signature detection IDS [2] uses a set of rules to look for attacks. The rules contain patterns that correspond to the data that represents a possible attack. If the pattern matches data on the network or in a host there may be a possible attack in progress.

Another type of IDS, called an Anomaly Detection IDS [2], looks for abnormal network behavior to identify a possible attack. Obviously a single type of IDS will not be sufficient to detect all types of attacks. Therefore, by combining the capabilities of various IDSs a more comprehensive approach to attack detection can be deployed.

The IDSs described thus far have one characteristic in common. That characteristic is that they examine snapshots of data in order to detect an attack. A snapshot of data that is examined from a network represents a single state of an aspect of the network. If an attack is such that it can be recognized by examining a single piece of data (usually a packet) then these methods will be successful in detecting a potential network attack.

A complex network attack is one in which the temporal domain and the spatial domain of the data must be examined in concert in order to determine if an attack is underway. The temporal domain of an attack consists of examining a period of time in which network events occur. During this period of time there may be multiple events that represent an attack in progress.

The spatial domain refers to position or location in the network. For an IDS this might mean physical location, logical location in the network, such as the subnet, or the node's neighbor nodes. The spatial domain consists of two aspects. One of these is strictly spatial and the other is a temporal-spatial aspect. The first aspect is where the events occur in the topography of the network. Again, certain events occurring at certain locations in the network may indicate an attack while those at other locations may not. The second aspect is represented by the sequence of events. In other words, a specific sequence of events during a specific time period may represent a possible attack whereas a different sequence of the same events may not.

As attacks become more complex, i.e., having several attack vectors across multiple temporal and spatial domains, more sophisticated IDSs will be necessary to detect these more sophisticated attacks. Our research involves designing a process for the detection of complex network and host attacks.

### 1.1 Detecting Sophisticated Network Attacks

In order to create an IDS that can detect complex network and host attacks it is necessary to employ some mechanism that can recognize an attack based on multiple events occurring in multiple locations at different times. In order to accomplish this it is necessary to be able to recognize

combinations of events that represent possible attacks.

Our research involved developing a reasoning system to monitor network traffic and identify the occurrence of a complex attack. Packets and sequences of packets were examined so a more robust analysis could be completed.

Consider a simple example of the kind of network event to observe in order to detect a complex attack attempt. One of the ways that attackers breach systems is by searching for open ports. Observing an attempt being made to determine if a specific port is open indicates very little about whether an attack is underway. Alternatively, if network traffic is observed indicating that a sequence of open port checks is being made over a period of time consistent with a port scanner, then, in fact, an attack may be in progress.

Our reasoning system consisted of a description of specific attack elements that could take place over a network. An attack element is a description of a complex attack component. This component combined with other components comprises the attack.

Decomposing attacks into attack elements enables the ability to better understand how attacks are constructed by attackers. For example, it may be the case that a certain combination of exploits is used to achieve a certain type of breach. Knowing the result of this combination provides the capability to consider other related combinations of attacks that yield the same results. Our research entailed exploring to what extent the assemblage of attack elements into potential attacks could be performed automatically. By doing so, the initial knowledge of specific attacks could be leveraged into a wider, more comprehensive set of attack descriptions.

This automation process allows for a more extensive range of attacks to be identified, including potential zero-day attacks (a new attack that exploits a vulnerability unknown to the general public or software developers). The reasoning system employs an ontological representation of attack elements and attacks. Such a representation describes entities that will be reasoned about and also how the reasoning will take place.

Our contribution to the field of intrusion detection is threefold. First, by identifying a family of complex attacks we enable better detection of these types of attacks. Second, by representing these complex attacks ontologically we create an advanced and reusable representation of network attacks. Third, by developing an algorithm that reasons about attacks using ontology [3], which we will develop, we provide a means to extend our knowledge of complex attacks allowing for better detection of such attacks.

The remainder of this paper is organized as follows. Section II discusses related work. Section III discusses the methodology for the approach developed in this research. A simulation environment is described in Section IV. Section V provides conclusions and future work.

## 2 Related Work

Over the years there has been a significant amount of IDS research. Many surveys of the current state of IDSs and IDS

research exist, such as in [2]. The basic types of IDSs are either signature- or anomaly-based. In a signature-based IDS packet data is examined to determine if it represents a known attack pattern. In an anomaly-based IDS the behavior of the network is considered and an attempt is made to recognize aberrant network behavior(s).

The evolution of IDSs consists of more complex approaches to analyzing network data to make assessments of the state of the network to determine if there is a potential breach. More advanced techniques incorporate data mining [4], fuzzy logic [5], learning [6], and logic [7].

Huang and Wicks [8] use the analogy of an intrusion to that of a battlefield. In intrusion detection as well as in the battlefield they cite a number of shared characteristics including an environment that is heterogeneous and widely distributed, a significant amount of data that is constantly changing and which can be extremely noisy, incomplete and inconclusive information that makes decision making difficult, and attack patterns which are constantly changing. When thinking about intrusion detection, one must take these characteristics into account when devising mechanisms for detection.

Huang and Wicks [8] point out that if a file-access-violation is detected, the true purpose of this event cannot be determined without additional information referred to as context. Such contextual information would include such information as the present machine configuration, the location of the files, permissions, and account configuration. The important point that Huang and Wick make is that by the time sufficient information arrives at a central analysis point, the situation (context) may have changed drastically. Huang and Wicks approach to analyzing what may be happening it is appropriate to consider the strategy the attacker may be using. This in turn calls for a description of the attacks that more abstract in nature. This is consistent with the approach described in this paper, namely to represent descriptions of attacks in the form of a conceptual ontology.

In Camtepe and Yener [9] an approach to detection complex attacks is presented that is based on the construction of finite automata that represent the "patterns" of complex attacks. The define a non-deterministic enhanced finite automata to be a tuple consisting of  $Q$ , a set of states,  $Q_{PA}$ , a set of partial attack states,  $Q_A$ , a set of attack states,  $F$ , the input alphabet,  $D$ , a set of derivation rules for goals and subgoals,  $\Delta_F$  and  $\Delta_B$ , sets of forward and backward transition rules. The finite automata can recognize complex attack patterns. The automata implicitly specify the relationships between the attack elements and therefore, unlike a conceptual representation, no ability to generalize or specialize exists without the specification of another automata.

A Process Queuing System (PQS) was the method used in [10] to detect complex attacks. The complex attacks were represented as finite state machines (FSM) with the attack elements represented as states and the transitions were triggered by observations about the occurrence of an attack element or a response to an attack element. The FSM were represented as models, which could be incorporated into a hierarchy of models, allowing for high-level models to be

developed to detect complex attacks based on results of lower-level models.

The MulVAL [7] system uses a logical deduction process to determine the existence of an attack on the network. It consists of generic rules, including rules to determine if a vulnerability exists and the consequence of an exploit against the vulnerability. The network properties are obtained by the scanning process, consisting of scanners that run on each host and report their findings to the host running MulVAL. MulVAL then runs an analyzer on the properties received from all the scanners. This analysis is done in two phases, an attack simulation phase and a policy checking phase. The attack simulation phase identifies all possible data accesses of an attacker, which are then sent to the policy checking phase. This phase compares the output of the attack simulation phase with the specified security policy and identifies violations. The scanner must be run on each host and identifies vulnerabilities specific to each host.

An ontology-based intrusion detection system was described by Mandujano [11] and Mandujano, Galvin, and Nolzco [12]. In this approach, the authors are looking to detect outgoing intrusions. The ontology they propose enables the detection of code and network activity that identifies a possible intruder. The ontology specifies concepts like hostile and safe processes as subclasses of process for example. There ontology, unlike the ontology proposed in this paper does not distinguish between traffic and attack. It is our contention that such a distinction is necessary to successfully identify sequences of incoming attacks and also to be able to recognize the type and kind of attack that is transpiring.

Another system that makes use of ontology as its basis for operation is the Reaction after Detection (ReD) Project [13]. ReD was developed to determine the best reaction to an attack. The system architecture consists of several components that assist in deciding short and long term reactions. The long term reaction consists of the deployment of new security policies to the network. An ontology-based approach is used to instantiate these new security policies. ReD makes decisions based on existing security policies and reacts by instantiating new policies. The ontologies employed in ReD provide the basis for recognizing policy violations. Rules are employed to analyze a violation and also to determine an appropriate reaction. An attack recognition and remedy may involve additional aspects beyond security policies, which were considered in the model proposed in our research.

Undercoffer, Joshi, and Pinkston [14] described a model for a host-based IDS system that uses ontology. The anomaly detector detects abnormal behavior at the system level and determines if data samples fall within a normal state compared to a baseline. Subsequent samples that fall outside the bounds of the normal state are identified as possible abnormal behavior. The ontologies define the properties of the target of the attack and the attack itself, including the consequences and means of the attack. Once again, this system is host-based and does not consider all the network components, such as the routers and switches in the network.

Similar to many of these examples, the IDS we are proposing uses ontology to represent complex types of attacks

that can take place. Unlike these examples, our system utilizes ontology to identify complex attacks against any node in the network, including network devices. Our literature survey indicates that the detection of complex attacks remains an area of research that has not reached a viable solution. Our proposed method for improving performance of IDS for detecting complex attacks represents a significant contribution to this research.

### 3 Methodology

The occurrence of simple attacks may indicate that an attacker is just trying an easy attack. The assemblage of several simple attacks may indicate the occurrence of a more complex attack. In order to understand the way simple attacks may fit together to form a complex attack, it is necessary to consider their spatial and temporal properties. For instance, pings to hosts on the same network, with incrementing IP addresses, over a span of several days, may indicate a network manager doing simple management or troubleshooting tasks. Given the same set of pings, over a span of several minutes, typically indicates an attacker looking for available hosts to attack. It is necessary to see that these ping packets are generated from the same source host and also within the same time period.

This research analyzed network packets captured using the Wireshark [15] packet capture software from both a spatial and temporal view. The analysis made use of some existing tools. Snort [16, 17], a combined NIDS and Intrusion Prevention System (IPS), was used to check the captured packets for possible attacks identified by the snort rules. The tcpdump [18] utility was used to convert the pcap file from Wireshark to an ASCII format for processing.

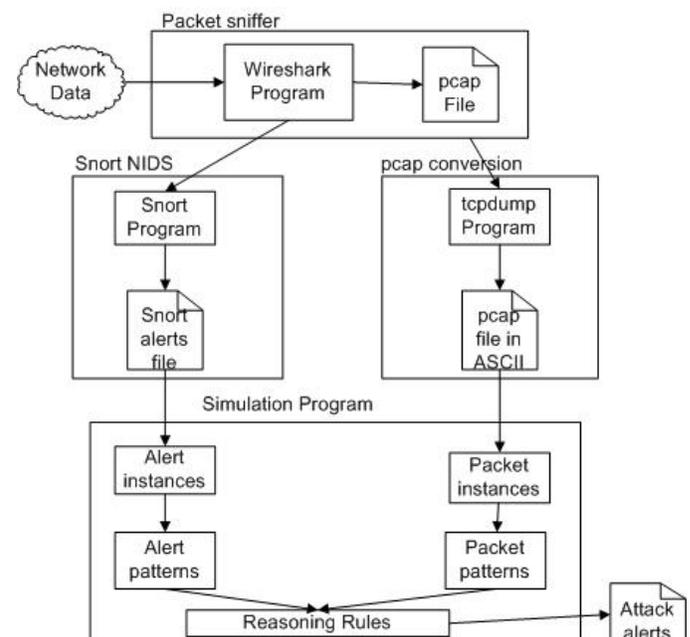


Figure 1: Reasoning system design.

The overall system design is illustrated in Fig. 1. The network data was captured using Wireshark, which created a pcap file of all the data captured. This pcap file was processed by snort and tcpdump, which generated output for use by the simulation program. The simulation program used the snort alert file, output from the snort processing, and the ASCII form of the pcap file, output from the tcpdump processing. Both of these outputs were used to create instances of the alerts and packets. These instances were used as inputs to the reasoning rules using pattern matching, which produced attack alerts.

TABLE I  
PROPERTIES FOR ALERTS AND PACKETS

Property Description	Alert Field	Packet Field
Date and time of item	X	X
Item description	X	
Classification of item	X	
Source IP address	X	X
Destination IP address	X	X
Source port number	X	X
Destination port number	X	X
Source MAC address		X
Destination MAC address		X
IP version number		X
Length of IP header	X	X
Length of IP datagram	X	X
IP upper-layer protocol		X
IP flags		X
IP fragment offset		X
IP time to live		X
IP checksum		X
TCP sequence number	X	X
TCP acknowledgement number	X	X
TCP flags	X	X
TCP window size		X
Upper-layer protocol checksum		X
ICMP type	X	X
ICMP code	X	X
Application-layer protocol		X
Application-layer data (first 80 bytes)		X

The alert field and packet field columns indicate if the property is maintained for an alert or packet, respectively.

Instances were created for the alerts and packets. These represented all the alerts found by snort and all the packets captured with Wireshark. Table 1 shows the properties that were maintained for the alerts and packets. Ontologies are used to map data instances into attacks. The result of this mapping is an instance of a colored attack tree.

The instances were searched for the occurrences of patterns representing simple attacks. If a pattern was found, data corresponding to the attack was written to an output file.

A complex attack is the combination of several simple attacks. Many times it is necessary for these simple attacks to occur within a specified timeframe. To determine if the simple attacks occurred within the specified timeframe, the time from the first node being colored in an attack tree to the time the

root node is colored is measured. Finding the most effective timeframe is a critical step in complex attack identification and will be determined in future work.

### 3.1 Attack Trees

Attack trees were created for the complex attacks used as examples (see Fig. 2). Each step was given a unique node id. For each complex attack example, data was generated as the attack was conducted. The data was manually analyzed to identify patterns so the attack could be identified from either the snort alerts or the captured data packets. A coloring scheme was used in the attack trees to identify the nodes that correspond to identified simple attacks.

Steps in a complex attack are often similar across several complex attacks. For instance, many attacks begin by the attacker identifying available hosts on a network and then proceeding by identifying open ports on each available host found. These generic simple attacks were identified with a unique attack id number. A mapping was created to map the

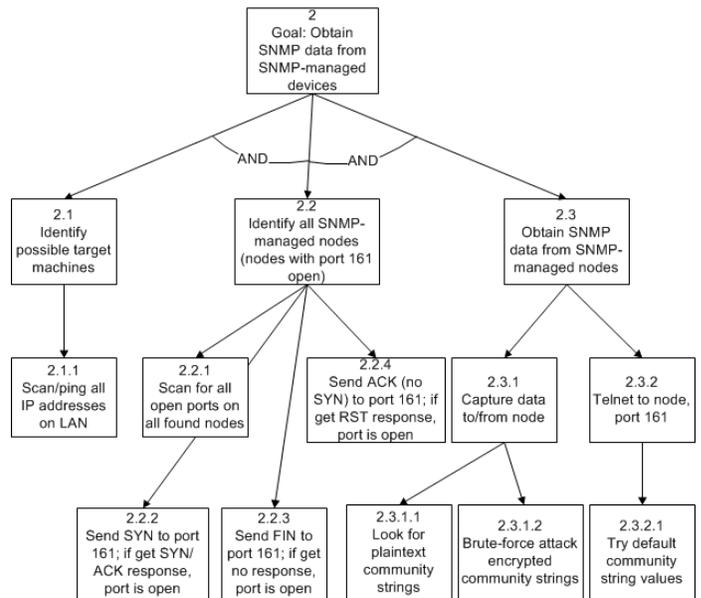


Figure 2: An example attack tree.

steps in the attack tree to its corresponding generic simple attack, if such a mapping exists. For instance, several different complex attacks include a step to identify an open port on a host. This port-scanning node in each specific complex attack was mapped to the generic simple attack of “identify an open port”. When looking for the occurrence of an attack, only the generic simple attack must be identified. The mapping was then consulted to determine all the nodes in all the complex attacks that were to be colored (selected) based on the fact that the generic simple attack was attempted.

### 3.2 Coloring Scheme

The alerts and captured data packets were processed and attack patterns identified. If the attack was a generic attack, all

corresponding attack tree nodes were identified. The identified nodes in the attack tree were colored according to a three-color scheme: 1) green to indicate no attack occurred, 2) yellow to indicate an attack may have occurred, and 3) red to indicate that an attack most likely occurred.

Nodes are colored according to a priority assigned to the attack element. This priority is determined based on three categories of analysis: rate, access level and alert priority. These categories were chosen as input to the coloring scheme from empirical evidence of manual analysis by network managers. Manual analysis has typically been devoted to attack elements that occurred more frequently, affected a higher-level of user, or were assigned a higher priority by an IDS.

The rate is how often the element occurred in a given temporal frame and can be assigned a value of 1 through 4. If the element occurred once, the rate was assigned a value of 1, occurring more than once but less than a specified threshold is a value of 2, more than the threshold but less than twice the threshold is a value of 3 and more than twice the threshold is a value of 4. The determination of the most effective threshold is an ongoing evaluation process.

The access level is the user or privilege level gained by a successful attack. The level may vary based on the node type, such as host (a user level) or network device (privilege level). Four access levels have been defined. The lowest level is the access level, which is similar to anonymous, and gives a remote user access to a network resource. An example of this would be the web user on a host running a web server for accesses. This category is assigned a value of 0 because it is an access level provided to any remote user for specific services in a network. The user level, assigned a value of 1, is a typical user on a host. The admin (value of 2) and root (value of 3) users are separated as different access levels since they can have separate privileges based on the operating system. For instance, on a host running Windows, the administrator user is not the same as a root user on Unix since some privileges in Windows require the local administrator. For a network device running SNMP there are also SNMP privileges that can give an attacker access to device information. If the attacker gains read-only access to SNMP on the device, the access level is given a value of 1 as this is similar to a basic user on a host. Read-write access in SNMP gives the user full control of all SNMP data, including the ability to modify the device configuration, so a value of 3 is assigned to this type of attack element.

The last category considered in the coloring scheme is the snort alert priority. Snort has priorities assigned to some of the alerts raised with default values ranging from 0 to 3, with 0 being the highest priority. Since this work uses 0 as the lowest priority, this value is assigned by the formula

$$\text{value} = 3 - \text{snort\_priority} + 1 \quad (1)$$

The highest numeric value assigned in a snort priority is 3, so the snort priority is subtracted from this value to reverse the order of the snort priority (make the highest priority having a value of 0 now be the highest priority with a value of 3). One is added to this value because since the alert was raised by snort, it must be of some importance and should be considered

as an attack element. Table 2 shows the three categories used to determine the attack element priority with their corresponding values.

TABLE 2  
ATTACK ELEMENT PRIORITY

Category or item	Value
Rate	
Occurs once	1
$1 < \text{occurrence} < \text{threshold}$	2
$\text{Threshold} < \text{occurrence} < 2 * \text{threshold}$	3
$\text{Occurs} > 2 * \text{threshold}$	4
Access level	
Access (anonymous)	0
User and SNMP read-only	1
Admin	2
Root and SNMP read-write	3
Snort priority	$3 - \text{snort priority} + 1$

The values of the three categories (rate, access level and snort priority) are added together to obtain the priority of the attack element. The attack element, which corresponds to a node in the attack tree, is then colored accordingly. The attack element priority can be a value from 0 to 11. This range is divided evenly to determine the appropriate color for the node coloring. If the priority is less than or equal to 3, the node is colored green, if it is more than 3 but less than 8, the node is colored yellow, and if it is 8 or more, the node is colored red.

After the node is colored properly, the coloring must propagate up the attack tree. The parent nodes in the attack tree are colored based on their children nodes' colors. Nodes in an attack tree can have an OR condition or an AND condition. If the children nodes are an OR condition, then the parent node is colored with the "largest" color (green < yellow < red) of its children. Coloring the parent nodes gets more complicated if the children nodes have an AND condition. In this case, unless all the children nodes are green, in which case the parent node is colored green, the green children nodes are excluded in the color analysis. First, the majority color of the children nodes is found; if there is the same number of yellow and red children nodes, then the color of the node that was just colored is used. If the parent node is the same color as the majority of its children, then the parent node remains that color. If the parent is currently colored a higher color (red), but it was colored red more than a time threshold ago (meaning the attack was too long ago to be meaningful now), then it is colored the same color as the majority of its children; otherwise the color of the parent node is left unchanged. If the root node of the attack tree is colored yellow or red, an alert is generated that a complex attack may have occurred (if the root node was yellow) or most likely did occur (if the root node was red). The coloring scheme algorithm is outlined in Fig. 3.

For example, the attack tree in Fig. 2, may be colored based on simple attacks observed. If it was observed that the attack scanned all hosts on the network and did a telnet to the SNMP

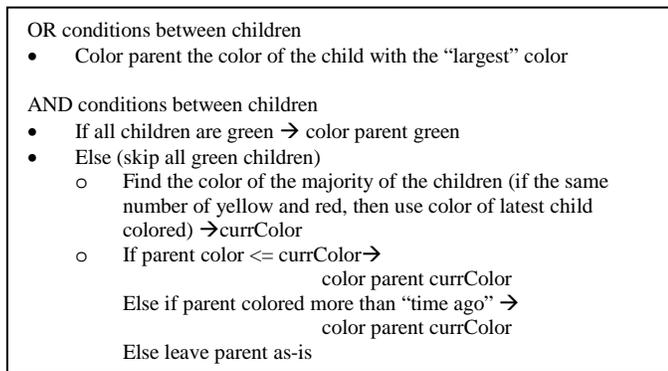


Figure 3: Coloring scheme.

port of a SNMP-managed node, these nodes are colored yellow (assuming the algorithm identified them as yellow and not red) (see Fig. 4; yellow is colored light grey in the figure and red is dark grey). The program found several ports scans over a short period of time so it was determined that this simple attack most likely occurred and the node was colored red.

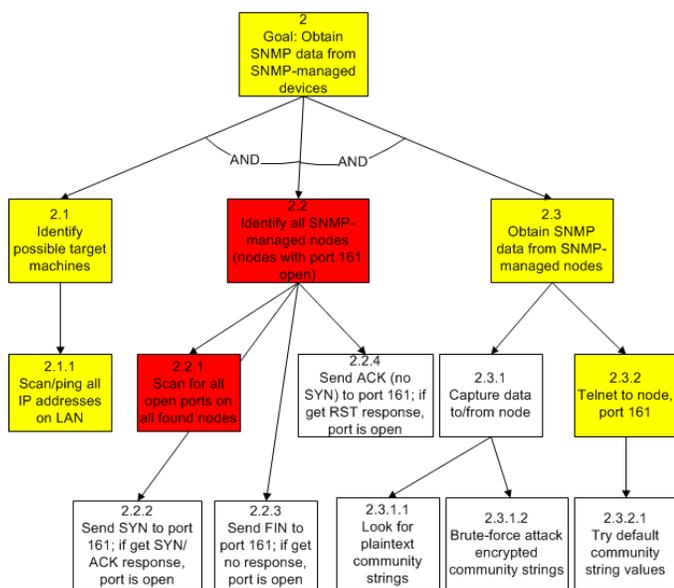


Figure 4: An example attack tree colored.

## 4 Simulation

Six complex attacks, varying in nature, were analyzed and tested. The first attack was an exploit of a Cisco HTTP vulnerability attack. In this attack the attacker will find a Cisco router on the network and exploit an HTTP vulnerability that will allow the attacker to download and modify the router configuration. Another complex attack, also against a router on the network, was an SNMP exploit that would allow an attacker to get access to the router's configuration file by obtaining the SNMP community strings. A TCP SYN flood attack against a router would potentially take the router off-line by consuming all the available TCP connections in the

router thus causing future TCP connection requests to be denied. The common man-in-the-middle attack was also used as an example, possibly leading to eavesdropping or packet interception/modification. An RPC Locator attack was another example, which may cause 100% CPU utilization on a node causing a denial of service by the device. The last example used was exploiting a vulnerability in apache, taking advantage of chunk encoding, that would allow arbitrary code to be executed on the apache web server.

### 4.1 Environment

Each attack was executed in a test environment, consisting of two laptops, one running Windows XP and the other Fedora 13, a Nortel switch, and a Cisco catalyst router. The execution of each attack consisted of at least one attack path with some attacks having multiple paths attempted. Wireshark was used to generate a pcap file during each attack. A truth file was created for each attack generation / pcap file combination.

Snort and tcpdump were executed with the pcap file for each complex attack. The analysis of the snort alert file and tcpdump file was simulated in a C program. The program (see Fig. 5) created an output file for all simple attacks identified, indicating the generic attack id or specific tree node id and the appropriate color for each corresponding node in the attack trees.

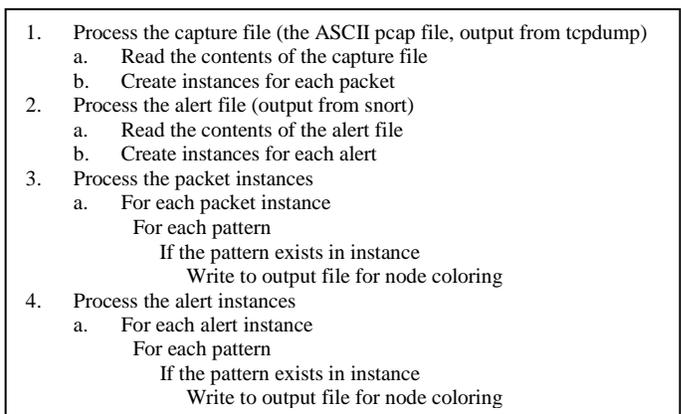


Figure 5: Steps in the simulation program.

### 4.2 Results

The format of the output included three fields: the date and time of the packet, the attack tree node to be colored, and the color (1 – green, 2 – yellow, 3- red). An example output for a run against the SNMP exploit example attack is provided in Fig. 6. This output is simply a textual-based version representing the nodes to be colored in the attack trees based on the simple attacks identified. By looking at the sample output, it indicates that the nodes numbered 2.2.2, 5.2.2, and 6.2.2 were all colored red (represented by the 3 in the third field of the output). The coloring scheme was then applied to color the nodes higher in the attack trees appropriately.

An attack path in each of the six complex attacks was followed and the corresponding network data captured. Fig. 4

08/12-16:17:51.307640	2.2.2	3
08/12-16:17:51.307640	5.2.2	3
08/12-16:17:51.307640	6.2.2	3
08/12-16:17:51.307793	2.2.2	3
08/12-16:17:51.307793	5.2.2	3
08/12-16:17:51.307793	6.2.2	3

Figure 6: An example output

shows an example of the colored attack tree for one of the complex attacks, after the coloring was propagated up the attack tree. The simulation was run against each complex attack. In each case, the correct patterns for simple attacks were recognized in the network data. All the correct nodes were identified and colored appropriately in the attack trees, proving the developed approach and algorithm worked as expected.

## 5 Conclusions

Networks are attacked daily. It is imperative that these attacks be identified, and if possible, a remedy determined. In an ideal situation, a remedy would also help prevent future similar attacks. The use of IDSs is one method used to identify attacks. Incorporating more advanced reasoning into an IDS would prove beneficial, particularly if it would lead to the remedy.

We proposed an approach to intrusion detection that used advanced reasoning through representing knowledge about attacks in ontology. Ontology will be utilized to correlate the various attack elements in a complex attack for easier identification of complex attacks. It will also allow high-level attacks to be abstracted from the low-level attacks.

Rules will be used in conjunction with the ontology to create an instance of an attack tree. Nodes of the attack tree are colored using a three-color scheme to indicate when an attack possibly occurred or most likely occurred. The root node color is used to indicate the possibility that a complex attack occurred.

The ontologies will be developed and incorporated into the system. The goal of the research is to design and implement a reasoning system using ontologies that will identify the occurrence of a complex attack as well as a remedy for the attack. The ontological representation of complex attacks will allow a better understanding of complex attacks and assist with the creation of an advanced and reusable representation of network attacks.

## 6 References

- [1] A. Fuchsberger, "Intrusion Detection Systems and Intrusion Prevention Systems", *Information Security Tech. Report*, vol. 10, issue 3, pp. 134-139, Jan. 2005.
- [2] K. Sailesh, "Survey of Current Network Intrusion Detection Techniques", Available at <http://www.cs.wustl.edu/~jain/cse571-07/ftp/ids.pdf>, 2007.
- [3] P. Spyns, R. Meersman, and M. Jarrar, "Data modeling versus Ontology engineering", *ACM SIGMOD Record*, vol. 31, issue 4, pp. 12-17, Dec. 2002.

- [4] T. Lappas and K. Pelechrinis, "Data Mining Techniques for (Network) Intrusion Detection Systems", Department of Computer Science and Engineering UC Riverside, Riverside CA 92521, 2007.
- [5] N. Bashah, I. B. Shanmugam, and A. M. Ahmed, "Hybrid Intelligent Intrusion Detection System", Paper presented at the *World Academy of Science, Engineering and Technology*, June 2005.
- [6] M. Dass, "LIDS: A Learning Intrusion Detection System (Thesis), University of Georgia, Athens, GA, 2003.
- [7] X. Ou, S. Govindavajhala, A. Appel, "MulVAL: A logic-based network security analyzer", *14th USENIX Security Symposium*, Baltimore, MD, Aug. 2005.
- [8] M.-Y. Huang and T. M. Wicks, "A Large-scale Distributed Intrusion Detection Framework Based on Attack Strategy Analysis", *Second International Workshop on Recent Advances in Intrusion Detection, RAID'98*, Louvain-la-Neuve, Belgium, Sept. 1998.
- [9] S. A. Camtepe and B. Yener, "Modeling and detection of complex attacks", *Third International Conference on Security and Privacy in Communications Networks and the Workshops, 2007 (SecureComm 2007)*, Nice, France, pp. 234-243, Sept. 2007.
- [10] I. Gregorio-deSouza, V. H. Berk, et al, "Detection of complex cyber attacks", *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense V*, May 2006.
- [11] S. Mandujano, "An Ontology-supported Outbound Intrusion Detection System", *Proceedings of the 10th Conference on Artificial Intelligence and Applications, Taiwanese Association for Artificial Intelligence (TAAI 2005)*, Kaohsiung, Taiwan, Dec. 2005.
- [12] S. Mandujano, A. Galvan, and J. A. Nolazco, "An ontology-based multiagent approach to outbound intrusion detection", *ACS/IEEE 2005 International Conference on Computer Systems and Applications (AICCSA'05)*, Cairo, Egypt, Jan. 2005.
- [13] N. Cuppens-Boulahia, F. Cuppens, J. E. López de Vergara, E. Vázquez, J. Guerra, and H. Debar, "An ontology-based approach to react to network attacks", *International Journal of Information and Computer Security*, vol. 3, issue 3/4, pp. 280-305, Jan. 2009.
- [14] J. Undercoffer, A. Joshi, and J. Pinkston, "Modeling computer attacks: an ontology for intrusion detection", In G. Vigna, E. Jonsson, and C. Kruegel (Ed.), *The Sixth International Symposium on Recent Advances in Intrusion Detection*, pp.113-135, Springer, 2003.
- [15] "Wireshark." Retrieved May 7, 2010, from <http://www.wireshark.org/>.
- [16] "Snort." Retrieved May 18, 2010, from <http://www.snort.org/>.
- [17] B. Caswell, J. Beale, and A. Baker, *Snort IDS and IPS toolkit*. Burlington, MA: Syngress Publishing, Inc., 2007.
- [18] "Tcpdump." Retrieved June 10, 2010, from <http://www.tcpdump.org/>.

# Database Security Architecture for Detection of Malicious Transactions in Database

Udai Pratap Rao, Dhiren R. Patel

Dept. of Computer Engineering, S.V. National Institute of Technology Surat, Gujarat, INDIA-395007  
(upr@coed.svnit.ac.in, dhiren29p@gmail.com)

**Abstract** - *The protection of the data over the database is some how mandatory for the organization, so there is a demand of the security mechanism to protect the database. Even the existing security measures at the database application level are not able to protect the database completely from some malicious actions and reason may be especially because of insider attack. The main objective here, is to design and develop the acceptable database security mechanism and giving indication on how it can be ensured that the designed database security system is acceptable or not? The designed system can be considered if it detects the insider misuse over the database and give the complete protection to database. In this paper, we discuss the acceptable database security system for detection of malicious transactions in database.*

**Keywords:** Database Security, Database Intrusion Detection, Insider Attack

## 1 Introduction

There are many aspects to security in database applications, including security at the application layer and security at the database layer. While applications typically support a fairly complex set of access control policies, any one is having the direct access to the database can bypass the access control policies together. In addition to database administrators, anyone who discovers the database login/password used by the application has the ability to directly modify the database. Thus, even if all security measures have been taken to ensure security at the application logic level, we need to have the ability to detect any malicious actions into the database. To support this type of detection, database intrusion detection system is required wherein malicious transactions may be detected while system still compromising with the application level security measures. However there is a requirement of the approach based on database intrusion detection. Application based on the database systems is ubiquitous these days, often storing critical data that should not be compromised in any way. Such applications are built on multiple layers of software: at the top level is the application software, typically running on a web-enabled application server, at the next level is database system which stores the data, and below the database system are the operating system and storage system layers. Application security requires actions at each of these

levels. In this work we consider the security at database layer where data are stored and protected from the malicious actions.

Applications typically have a complex security model built into the application, but when communicating to the database, an application typically connects as single database user. Anyone who gets access to the database login/password used by the application has the ability to frequently read or modify the database, bypassing all the security features built into the application. This problem is exacerbated since the database login and password are often stored in clear text in the application code or configuration files, accessible to system administrators. In addition, database administrators have full access to the data in the database. When dealing with mission critical data, preventing, or detecting and repairing, unauthorized updates to the database is absolutely critical, even more than preventing or detecting unauthorized attacks, since it may severely affects the ability of the organization to function. In this paper, we address the problem of detecting unauthorized updates/ malicious actions to the database.

Rest of the paper is organized as follows: in Sec. 2, we discuss related work in this area. In Sec. 3, generic database security mechanism is discussed based on which we discuss our design in Sec. 4; with proposals' comparisons in Sec. 5 and conclusions and references in Sec. 6 & 7 respectively.

## 2 Related Work

The early research mainly focused on network-based and host-based intrusion detection. However, in spite of the significant role of databases in information systems, very limited research has been carried out in the field of intrusion detection in databases. We need intrusion detection systems that work at the application layer and potentially offer accurate detection for the targeted application.

The approaches used in detecting database intrusions mainly include data mining and Hidden Markov Model (HMM). Chung et al. [1] this paper presents a misuse detection system called DEMIDS which is tailored to relational database systems. DEMIDS uses audit logs to derive profiles that describe typical behavior of users working with the DBS. The profiles computed can be used to detect

misuse behavior, in particular insiders abuse. DEMIDS sue “working scope” to find frequent itemsets, which are sets of feature with certain values. They define a notation of distance measure that captures the closeness of set of attribute with respect to the working scopes. These distance measures are then used to guide the search for frequent item-sets in the audit logs. Misuse of data, such as tampering with the data integrity, is detected by comparing the derived profiles against organizations security police or new audit information gathered about users. The main drawback of the approach presented as in [1] is a lack of implementation and experimentation. The approach has only been described theoretically, and no empirical evidence has been presented of its performance as a detection mechanism. However, the impact of feature granularity level is not explored in this model. Lee et al. [2] have proposed a real-time database intrusion detection using time signatures. Real-time database systems have a deal with data that changes its value with time. These temporal data objects are used to reflect the status of object in the real world. Whenever the value of a real world object changes, the data that describes this object should change as well, but a certain lag between the moment of change in real world and the updates in the database is unavoidable. This intrusion detection model observes the database behavior at the level of sensor transaction. If a transaction attempts to update a temporal data which has already been updated in that period, an alarm is raised. Wenhui et al. [3] proposed a two-layer mechanism to detect intrusions against a web-based database services. They use web-server behavior modeling and database system behavior modeling by a profile process in the first layer. Layer one built historical profiles based on audit trails and other log data provided by the web server and database server. The pre-alarms generated from the first layer are passed to the second layer for further analysis. In layer one the tree topology was adopted to profile web server behavior. Moreover, to profile database server, a role-based model is adopted to deter describe the characteristics of the super user behavior. However, they have not used different level of granularity or intra-transactional and inter-transactional features in their model. Hu et al. [4] determine the dependency among data items where data dependency refers to the access correlations among data items. These data dependency are generated in the form of classification rules, i.e. before one data item is updated in the database, which other data items probably need to be read and after this data item is updated, which other data items are most likely to be updated by same transactions. Transactions that do not follow any of the mined data dependency rules are marked as malicious transactions. Database contain many attribute, all attribute to be consider for dependency rules generation, maintaining such rules are difficult. In this approach there is no concept for attribute sensitivity. These problem addresses by Srivastava et al [5], who consider attribute sensitivity in their IDS. In every database, some of the attributes are considered more sensitive to malicious modification compared to others. They suggest a weighted data mining algorithm for finding dependencies

among sensitive attributes. Any transaction that does not follow these dependency rules is identified as malicious. The main problem with this concept is the identification of proper support and confidence values. Srivastava et al. [5] can extract only intra-transactional attribute dependency and there is no consideration of inter-transactional attribute dependency. Recently, it has been shown that some of the algorithms used in the field of bioinformatics can also be applied to network intrusion detection. Takeda et al. [6] used series arrangement techniques to arrange series of network transfer pattern and known attack patterns. Arrangement score is evaluated based on their similarity, which is later used to detect intrusions. Zhong et al. [7] use query templates to mine user profiles. They developed an elementary transaction level user profile. A constrained query template is a four tuple  $\langle op, f, t, c \rangle$  where  $op$  is type of the SQL query,  $f$  is the set of attributes,  $t$  is the set of tables, and  $c$  is the constrained condition set. It uses an algorithm that mines user profile based on the pattern of submitted queries. An algorithm of mining database user query profiles of transaction level is presented. This algorithm changes the computing method of the support and confidence in association rules mining by adding query structure and attribute relations to the computation. Since there is no causal relationship in the access of attributes in queries, the method is more appropriate to describe user query behaviors than itemsets used by association rules. There is, however, no provision for handling various levels of granularity of access in their query template.

Lee et al. [8] DIDAFIT (Detecting Intrusions in Database through Fingerprinting Transactions) is a system developed to perform database intrusion detection at application level. It works by fingerprinting access patterns of the legitimate database transactions, and using them to identify database intrusions. The framework for DIDAFIT has been described in [9]. This paper describes how the fingerprints for database transactions can be represented and presents an algorithm to learn and summarize SQL statements into fingerprints. The main contribution of this work is a technique to efficiently summarize SQL statements queries into compact and effective regular expression fingerprints. If a given query does not match any of the existing fingerprints, it is reported as malicious. Kamra et al. [10] have proposed a role based approach for detecting malicious behavior in RBAC (role based access control) administered databases. Classification technique is used to deduce role profiles of normal user behavior. An alarm is raised if roles estimated by classification for given user is different than the actual role of a user. The approach is well suited for databases which employ role based access control mechanism. It also addresses insider threats scenario directly. But limitation of this approach is that it is query-based approach and it cannot extract correlation among queries in the transaction. This problem is resolved by Rao et al. [11] it extracts correlation among queries in the transaction. In this approach database log is read to extract the list of table accessed by transaction and list of attribute read and written by transaction. This

approach supports the correlation between queries of transaction. By using this approach if a transaction contains two queries and it is supported by the application, then authorized user of particular transaction must issue the both query of transaction one by one. If any user issues only one query of the defined transaction then the executable transaction is marked as malicious transaction. This approach is well suited for handling of insider attack completely. Srivastava et al. [12] have proposed the use of weighted association rule mining for speeding up web access by pre-fetching the URLs. These pages may keep in a server's cache to speed up web access. Existing techniques of selecting pages to be cached do not capture a user's surfing patterns correctly. It use a weighted association rule (WAR) mining technique that finds pages of the user's current interest and cache them to give faster net access. This approach captures both user's habit and interest as compared to other where emphasis is only on habit.

### 3 Design of Database Intrusion Detection Architecture

The existing security mechanism in the DBMS is based on mainly auditing mechanism, wherein the auditing mechanism does not guarantee about the full protection of the database. So there is a need of additional security mechanism over the database to ensure the protection of database from malicious actions. Currently database intrusion detection system has become the new area of research in the DBMS and people have taken interest to develop the database IDS. Database intrusion detection system (IDS) can serve as an additional layer of database security, which applies the ideas and results from generic intrusion detection research to detect misuses and basically targeted towards database systems. The general security architecture in DBMS is shown in the figure 1, as it is the case in the existing scenario of the database protection without inclusion of the database intrusion detection system (IDS).

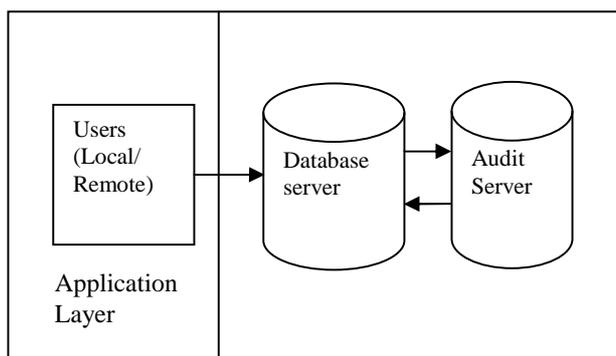


Figure 1. The general security architecture in DBMS

The database server is extended into an other unit and it is known as audit server, the audit server supports the auditing mechanism to ensure that executable query is to be allowed over the database or not. The audit mechanism over the

database server is designed to both deter and reveal attempted, as well as successful, security violations. However, the audit data is gathered from the log file and it is being taken out by the existing auditing mechanism. If we talk about the database transactions then we know that the transaction is consisting the number of operations and each may be intended for particular object. The history of the log file consists the information about the many attributes and these information's are useful in the view of the auditing server, on the other hand, database server routinely analyze and review the audit data, then it might discover suspicious activity before a successful violation occurs and it can be done periodically to update the audit data. In the view of the databases security the main concerned is to provide the protection of the database completely from internal and external attacks, to achieve this there is a requirement of such database IDS to act as an additional layer of security over the database. The design issue and analysis of proposed database intrusion detection systems are discussed in next section.

### 4 Our Design and Analysis

Based on the literature of designing of database intrusion detection system we investigated and proposed three possible combination of database intrusion detection system and these may be used to handle the malicious transaction over the database and prove the complete protection to database. The details about each one are produced herewith as below.

#### 4.1 First Approach: Database Intrusion Detection System for Role Based Enabled Database

The proposed approach in this section is the extension of [10] , as from query based approach to transaction based approach. The proposed approach we are discussing is presented in [11], the main advantage of this approach is to extract the information among queries in the transaction. For example consider the following transaction:

```

Begin transaction
select a1,a2,a3,a4,a5 from t1,t2;
update t2 set a4= a2+1.2(a3);
End transaction
    
```

Where t1 and t2 are tables of the database and a1, a2, a3 are the attributes of table t1 and a4, a5 are the attributes of table t2 respectively. This example shows the correlation between the two queries of the transaction. It states that after issuing select query, the update query should also be issued by same user and in the same transaction. The approach based on the RBAC database uses the Naïve Bayes classifier as a learning algorithm to generate the role profiles on training data, and the training data which one is extracted from the log file and

stored into the form of particular representation to represent the user transaction behavior.

### 4.2 Second Approach: Database Intrusion Detection System Using Legal Transaction Profiles

Basically this proposed approach is divided into three steps: Auto-generation legal profile phase, Detection phase, Action phase. It takes the advantage over the manual transaction profiles mechanism. As in this case the time to generate the legal transaction profile is reduced, also it overcomes the disadvantage of the existing system based on manual profile generation [13]. The log file is used from which the history of the transactions are extracted and stored into the offline audit trail and this can be done using the inclusion of existing auditing mechanism. Later the generated legal transaction profiles from offline audit trail are used at the detection phase to match with the executable transactions; if any deviation is there then particular executable transaction is marked as malicious otherwise committed into the database. The last phase is the action phase and it may take the action based on the alarm generated by the database IDS.

### 4.3 Third Approach: Database Intrusion Detection System Using Counting Bloom Filter (CBF)

A Bloom filter [14] is used to define the bit array of  $m$  elements of  $n$  bits size and initially all set to 0. The filter uses a group  $H$  of  $k$  independent hash functions  $h_1, \dots, h_k$  with range  $\{1, \dots, n\}$  that independently map each element in the universe to a random number uniformly over the range. For each element  $x \in S$ , the bits  $B[hi(x)]$  are set to 1 for  $1 \leq i \leq k$ . (A bit can be set to 1 multiple times.) To answer a query of the form "Is  $y \in S$ ?", we check whether all  $h_i(y)$  are set to 1. If not,  $y$  is not a member of  $S$ , by the construction. If all  $h_i(y)$  are set to 1, it is assumed that  $y$  is in  $S$ , and hence a Bloom filter may yield a false positive. The main problem with the bloom filter is the false positive i.e. it gives the wrong answer with correct query, and it is resolved using the counting bloom filter (CBF) where insertion and deletion of the set of the elements are possible. It also uses as similar to the bloom filter,  $k$  (random hash) functions, each of which maps or hashes some set element to one of the  $n$  bits array positions. To insert an element into a set, the element is passed into  $k$  hashing functions and  $k$  index values are obtained. All counters in counting bloom filter at corresponding index values are incremented. To delete an element from the set reverse process is followed and corresponding counters are decremented. Thus a counting Bloom filter (CBF) generalizes a Bloom filter data structure by allowing the membership queries and CBF can be changed dynamically by insertions and deletions operations by this it resolves the problem of a

standard bloom filter with false positive. The overall approach based on the CBF is divided into the three phases.

The initial phase is as similar to the automatic transaction profile generation algorithm to generate the authorized transactions. This process insures the correctness of the genuine profiles as declared as the legal profiles, its do automatically instead of manually thus it reduces the time to require for manual transaction profile generation.

This next phase is all about the construction of the counting bloom filter (CBF) where random weights are assigned automatically corresponding to commands of legal transactional profile. After that the construction of the CBF is done by incorporating the hash functions.

At the final stage of detection phase the constructed CBF along with the weights are loaded and the counter values in CBF are decremented using weight of identified command based on the executable transaction, if all the bits in the CBF are zero then the transaction is declared as valid.

## 5 Comparison of All Three Proposed Approaches

For the comparison we consider the set of parameters to evaluate each approach with other one. The complete details of comparison are given in below table 1.

Table 1. Comparison of proposed approaches for database IDS

Approaches	Learning Time	False Positive	False Negative	Load on Server
First Approach	less	no	no	Yes
Second Approach	less	no	no	Yes
Third Approach	more	no	no	Yes
Only Based on Auditing Mechanism	-	no/yes	no/yes	Less

Based on the information in the above table as we can see the proposed approaches are very much useful to handle the malicious transaction once it is executed by the unauthorized user. The proposed approach also applicable to handle the internal misuse over the database. If we see the load on the database server for proposed mechanisms then it is quite high because of the inclusion of one additional layer of security into the database but it is less in auditing mechanism.

## 6 Conclusion and Future Work

The security in the DBMS is one of the main concerns of the researchers now-a-days and there is an interest to develop the possible database intrusion detection systems. We discuss the three approaches for database IDS and basic design of such architectures. We again emphasize that to the best of our knowledge, this is the one literature presenting the design of database IDS architectures. We further intend to extend our work to support the actual implementation of action phase and then further for database recovery.

## 7 References

- [1] C. Y. chung, M. Gertz, K. Levitt, "DEMIDS: A Misuse Detection System for Database systems", IFIP TC-11 WG 11.5 Conference on integrity and internal control in information system, pp. 159-178, 1999.
- [2] V. C. S. Lee, J.A. Stankovic, S. H. Son, "intrusion detection in real-time database system Via time signatures", real time technology and application symposium, pp. 124, 2000.
- [3] Wenhui, S., Tan, T., "A novel intrusion detection system model for securing web based database systems", In proceedings of the 25th annual international computer software and application conference (COMPSAC), pp. 249-254, 2001.
- [4] Y. Hu, B. Panda, "A data mining approach for database intrusion detection", In Proceedings of the ACM Symposium on applied computing, pp. 711-716, 2004.
- [5] A. Srivastava, S. Sural, A. K. Majumdar, "Weighted intra-transactional rule mining for database intrusion detection", In proceedings of the Pacific-Asia knowledge discovery and data mining (PAKDD), lecture notes in artificial intelligence, Springer. pp. 611-620, 2006.
- [6] Takeda, K., "The application of bioinformatics to network intrusion detection", In proceedings of the international carnahan conference on security technology (CCST), pp. 130-132, 2005.
- [7] Zhong, Y., Qin, X., "Database intrusion detection based on user query frequent itemsets mining with constraints", In proceedings of the 3rd international conference on information security, pp. 224-225, 2004.
- [8] Low, W. L., Lee, S.Y., Teop, P., "DIDAFIT: Detecting Intrusion in database through Fingerprinting Transactions", In proceedings of the 4th International conference on enterprise information systems (ICEIS), 2002.
- [9] Lee, S.Y., Low, W.L., Wong, and P.Y., "Learning Fingerprints for a Database intrusion detection system", In proceedings of the 7th European symposium on research in computer security, pp.264-280, 2002.
- [10] Bertino, E., Terzi, E., Kamra, A., Vakali, A., "Intrusion Detection in RBAC-Administered Database", In proceedings of the 21st annual computer security application conference (ACSAC), pp. 170-182, 2005.
- [11] U. P. Rao, G. J. Sahani, D. R. Patel, "Detection of Malicious Activity in Role Based Access Control (RBAC) Enabled Databases", In proceedings of Journal of Information Assurance and Security ,Volume 5, Issue 6, pp. 611-617, 2010.
- [12] A. Srivastava, A. Bhosale, S. Sural, "Speeding up Web access using weighted association rules", Lecture notes in computer science, Springer Verlag, Proceedings of international conference on pattern recognition and machine intelligence (PReMI'05) , pp. 660-665, 2005.
- [13] Marco Vieira , Henrique Madeira, "Detection of Malicious Transactions in DBMS", in Dependable Computing proceedings 11th Pacific Rim International Symposium, IEEE Computer Society, pp. 350-357, 2005.
- [14] Flavio Bonomi, Michael Mitzenmacher, Rina Panigrahy, Sushil Singh and George Varghese, "An Improved Construction for Counting Bloom Filters", ESA 2006, LNCS 4168, pp. 684-695, 2006.

# Defence Against Dos Attacks Using a Partitioned Overlay Network.

Muhammad Usman Saeed  
iResearch, Interactive Group, Islamabad, Pakistan

**Abstract** - According to general statistics, around thousands of DOS and DDOS attacks have been carried out in the years 2009 and 2010. Choosing this problem for research was because everything in the industrial or mechanical sector is now controlled over the network through applications thus, securing these networks against DOS attacks is very important because once compromised it can cause a major damage to the infrastructure. This paper's idea revolves around the fact that hiding the network nodes mitigates DOS attack. This paper further extends the idea of node hiding with the architecture of a junkyard network where the suspected traffic will be routed and another overlay network which would contain the legitimate traffic to the proper destination.

**Keywords:** Denial of service, Network, Overlay, Security, Flood

## 1. Introduction

Denial of Service (DOS) attacks are a major issue in today's network. DOS attacks are the attacks in which millions of packets, either normal or malformed are sent to a server. Thus resulting in denial of service. Most of the security experts are working on ways to mitigate DOS attacks but due to the complexity and limitation of the underlying network DOS mitigation is the biggest problem nowadays.

Overlay networks are logical networks which basically are used to support an underlying network. Applications of overlay networks include Virtual private networks, P2P networks etc. Many techniques which use overlay are devised to counter act a DOS attack.

DOS attacks include Smurf Attack [12], UDP DOS attacks [13], SYN floods [14]. In a SYN flood attack the attacker can send many SYN requests to the client. Thus as a result the target system's memory increases to a very critical point thus the system goes into a state of Denial of service.

In a UDP flood attack thousands of UDP packets are aimed towards a target host thus crashing the target system.

Nowadays internet has become the lifeline for many crucial systems such as industrial, military. And due to the open nature of the internet it is vulnerable to many attacks including DOS attacks. As the systems are so crucial a microsecond of delay or interruption causes millions of worth of damage. Thus DOS attack defense is very important. Thus there is a need that more and more DOS attack prevention strategies may be devised so that the interruption or Denial of service problem may be reduced.

Sequence of the paper is as following:

An overview of what work is done related to this topic is given in Section 2. Section 3 gives the detail of what our proposed idea is and how we have planned the design and how it is implemented and in Section 4 conclusion of the whole discussion is given. Next section to 4 which is section 5 gives the future directions of our research which means that what can be added. The last section i.e. section 6 lists the references.

## 2. Related work

Many techniques have been devised so far for the defense against DOS attacks which include reactive techniques, proactive techniques.

These include ingress filtering [2], Source Trace back [3,4], rate control techniques [6,7]. Location hiding mechanisms are widely used for the defense against DOS attack [5,8,9]. The overlay protection layer was also devised to prevent DOS attacks [1].

The source trace back and ingress filtering comes in Reactive approach. Where as rate control, and location hiding comes in Proactive approach. Location hiding works on the principle that if the network nodes are hidden from the attacker then it means that the attacker can not attack on those nodes. That is for instance there is a hidden nodes network and one entry point, though the attacker can attack on the entry point but he/she won't know the location or the IP of other nodes let alone the target webserver. Overlay networks can be used as a mean for the location of a network device to be hidden. Overlay network is the only public interface for anyone to access the web server.

In an ingress filtering technique the routers have an ingress address range and the routers check the source IP of the incoming packet and if the address is out of the range of the ingress range, the packets are dropped.

There are many Chord [16] based overlay network Architectures, designed to defend against DDOS attacks and DOS attacks. Chord is a highly adaptive routing protocol used for the overlay networks.

Further more HOURS [10] using hierarchal overlay layers achieved DOS resilience in an open service hierarchy. Secure Overlay service (SOS) [8] and WEBSOS [5] were introduced for the protection of web servers against DDOS attacks. The OPL (overlay protection layer) [1] is uses an

overlay protection layer to help web servers / Applications communicate with each other even when there is a DOS attack in progress. And its goal is to distinguish between authorized and unauthorized traffic. In the paper "Deployable overlay network for defense against distributed SYN flood attacks" [11], an overlay method is described which defends against Distributed SYN flood attack. An Integrated notification architecture based on Overlay Networks against DDOS attacks on converged networks [15] also explains an overlay approach to how create a resilient network to fight against DDOS attacks in Converged Networks.

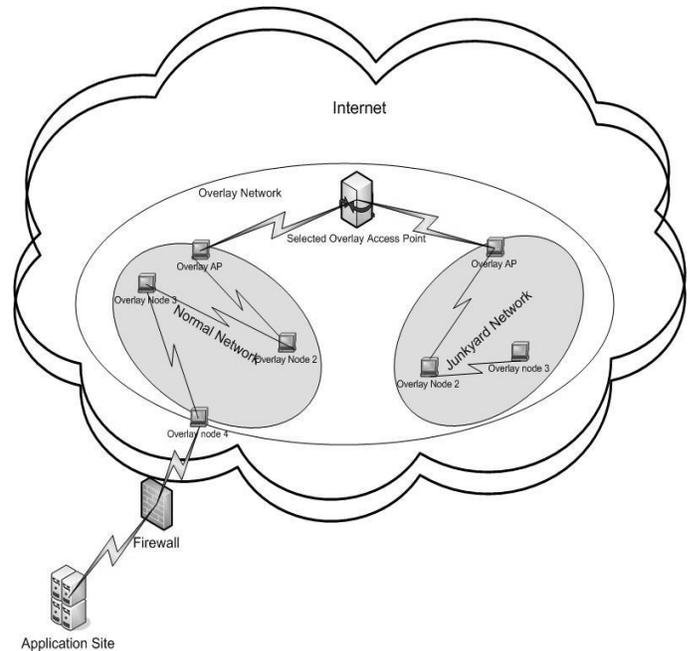
Every DOS Defense mechanism has its strengths and weaknesses None of the defense mechanism gives protection without any tradeoffs.

### 3. Junkyard Overlay Network

We propose an architecture which caters two problems. One is the problem at hand , that is Resistance against a DOS attack. The other can be included into the future work that is resistance against port scanning. The architecture proposed consists of an overlay network which forms the entry point to an application site. But the overlay network is further divided into two partitions, A normal overlay network and a junkyard overlay network. The overlay access point allows the traffic to enter into the overlay network. The overlay access point acts as a proxy and also as an intrusion prevention system. The communication between the overlay access point and the sub overlay network's access points is encrypted and authenticated using special signature.

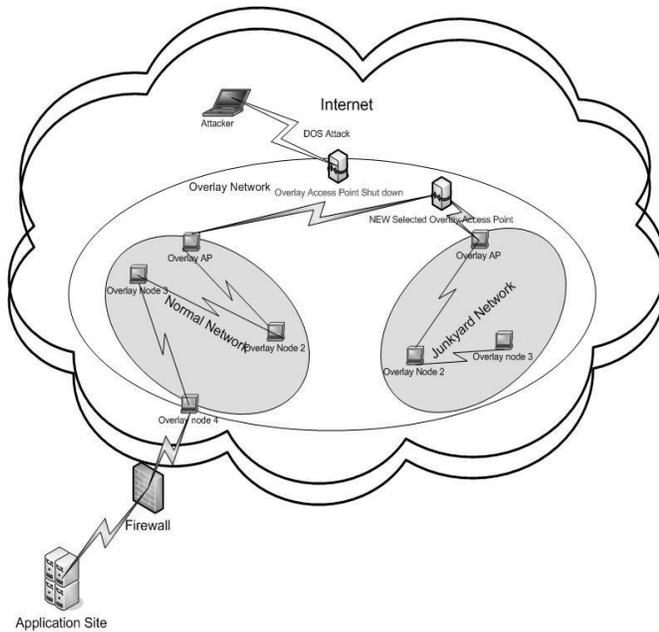
The functionality of the architecture is that when the DOS attack is once detected the suspected traffic is tunneled to the junkyard network where it is then dropped. Where as the legitimate traffic is sent through another overlay network which is connected to the Application Site . Figure 1 shows the Upper level view of the proposed architecture.

The decision as to how DOS is detected can be done on the basis of rate assessment. That is if the rate of the packets is very high or greater then a specific threshold then the defense mechanism is activated and the packets start going towards the junkyard network.



**Figure 1. The packets from the internet enter through the overlay access where its decided whether the traffic is legitimate or not. Then are routed according to the decision , to the entry nodes of the Normal Overlay network or the Junkyard overlay network.**

Now to solve the problem, that if the Overlay access point is attacked or flooded then what would happen? As soon as the packets hit the overlay access point and DOS is detected, the access point will shut itself down but before shutting itself down it chooses another OAP (Overlay Access Point) in the vicinity and sends a NOTIFY message containing the IP of the new OAP, to OAPs of both the partitions that is the Normal and the junkyard, telling them about the new assigned OAP. The OAPs of both the partitions send REG message to the new OAP where the authentication takes place and once registration is ok the OAP responds with REG\_OK message. Figure 2 shows the scenario.



**Figure 2. The attacker sends a DOS attack onto the OAP , It shuts itself down and a new OAP is selected and the communication continues.**

Though there are some tradeoffs that, like any system our system can also generate false positives in theory that is. But as the junkyard network is totally isolated from the normal legitimate network, even if the legitimate traffic is recognized after it has been sent to the junkyard overlay, there won't be any way to send it to the normal network because of the lack of any interface between Normal and junkyard network.

## 4. Conclusion

This paper showed the design of how the overlay network could be partitioned in order to minimize the load on the network and also to mitigate flooded packets targeted at a single host. The partition [junkyard] was used because detection is easy if the filtering is applied on a gateway of any target network. But it is difficult to detect whether filtering is in place or not if the filtering is done after 3 or four hop from the gateway going inside the network. Thus the proposed technique can one , form location hiding [5,8,9] as well as the factor of deception. Similarly if the gateway of the target network is under a DOS attack, the network automatically changes the overlay access point.

## 5. References

[1] Beitollahi, H.; Deconinck, G. An Overlay Protection Layer against Denial-of-Service Attacks.Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on Volume , Issue , April 2008

[2]. P. Ferguson and D. Senie. Network ingress filtering: defeating denial of service attacks which employ ip source address spoofing. In Proceedings of the IETF,RFC2267, January 1998

[3] S. Savage, D. Wetherall, A. karlin, and T. Anderson. Network support for ip traceback. ACM/IEEE Transactions on Networking, 9(3):226–237, June 2001.

[4] A. Snoeren. Hash-based ip traceback. In Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIG-COMM'01), pages 3–14, 2001.

[5] A. Stavrou and et al. Websos: An overlay-based system for protecting web servers from denial of service attacks. The International Journal of Computer and Telecommunications Networking, 48(5):781–807, August 2005.

[6] A. Garg and A. N. Reddy. Mitigation of dos attacks through qos regulation. In Proceedings of the 10th IEEE International Workshop on Quality of Service, 2002.

[7] K. Yau, C. Lui, and F. Liang. Defending against distributed denial of service attacks with max-min fair server-centric router throttles. In Proceedings of the IEEE International Workshop on Quality of Service (IWQoS'02), 2002.

[8] A. Keromytis, V. Misra, and D. Rubenstein. Sos: Secure overlay services. In Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'02), August 2002.

[9] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. In Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIG-COMM'02), 2002

[10] Yang, H., Luo, H., Yang, Y., Lu, S., Zhang, L.: HOURS: Achieving DoS Resilience in an Open Service Hierarchy. In: Proc. of DSN, pp. 83–92, 2004

[11] Ohsita, Y. Ata, S. Murata, M , Deployable overlay network for defense against distributed SYN flood attacks , ICCCN 2005. Proceedings. 14th International Conference on Publication Date: Oct. 2005

[12] “CERT advisory CA-1998-01 smurf IP Denial-of-Service attacks.” , Jan. 1998.

<http://www.cert.org/advisories/CA-1998-01.html>

[13] “CERT advisory CA-1996-01 UDP port Denial-of-Service attack.”.

<http://www.cert.org/advisories/CA-1996-01.html>.

[14] “CERT advisory CA-1996-21 TCP SYN flooding and IP spoof- ing attacks.”, Sept. 1996

<http://www.cert.org/advisories/CA-1996-21.html>

[15] Mihui Kim, Jaewon Seo, and Kijoon Chae , Integrated Notification Architecture Based on Overlay Against DDoS Attacks on Convergence Network , IFIP International Federation for Information Processing 2007

[16] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan , Chord: A Scalable Peer-to-

peer Lookup Service for Internet Applications , SIGCOMM'01, August 27-31, 2001, San Diego, California, USA.

[17] <http://asert.arbornetworks.com/2010/12/the-internet-goes-to-war/>



## **SESSION**

# **ALGORITHMS AND APPLICATIONS + MANAGEMENT INFRASTRUCTURES**

**Chair(s)**

**TBA**



# Application Resilience with Process Failures

K. McGill\*<sup>1</sup> and S. Taylor<sup>1</sup>

<sup>1</sup>Thayer School of Engineering, Dartmouth College, Hanover, NH, USA

**Abstract** - *The notion of resiliency is concerned with constructing mission-critical applications that are able to operate through a wide variety of failures, errors, and malicious attacks. A number of approaches have been proposed in the literature based on fault tolerance achieved through replication of resources. In general, these approaches provide graceful degradation of performance to the point of failure but do not guarantee progress in the presence of multiple cascading and recurrent attacks. Our approach is to dynamically replicate message-passing processes, detect inconsistencies in their behavior, and restore the level of fault tolerance as a computation proceeds.*

*This paper describes a novel operating system technology for resilient message-passing applications that is automated, scalable, and transparent. The technology provides mechanisms for process replication, multicast messaging, and process failure detection. We demonstrate resilience to failures and benchmark the performance impact using a distributed exemplar representative of applications constructed using domain decomposition.*

**Keywords:** resiliency; mission-assurance; distributed systems; process replication; failure detection

## 1 Introduction

Commercial-off-the-shelf (COTS) computer systems have traditionally provided several measures to protect against hardware failures, such as RAID file systems [1] and redundant power supplies [2]. Unfortunately, there has been relatively little success in providing similar levels of fault tolerance to software errors and exceptions. In recent years, computer network attacks have added a new dimension that decreases overall system reliability. A broad variety of technologies have been explored for detecting these attacks using intrusion detection systems [3], file-system integrity checkers [4]-[5], rootkit detectors [6]-[7], and a host of other technologies. Unfortunately, creative attackers and trusted insiders have continued to undermine confidence in software. These robustness issues are magnified in distributed applications, which provide multiple points of failure and attack.

Our previous attempts to implement resilience resulted in an application programming library called the Scalable Concurrent Programming Library (SCPLib) [8]-[9]. Unfortunately, the level of detail involved in programming

resilience in applications compounded the already complex activity of concurrent programming. The research described here explores operating system support for resilience that is automatic, scalable, and transparent to the programmer. This approach dynamically replicates processes, detects inconsistencies in their behavior, and restores the level of fault tolerance as the computation proceeds [8]-[9]. Fig. 1 illustrates how this strategy is achieved. At the application level, three communicating processes share information using message-passing. The underlying operating system implements a resilient view that replicates each process and organizes communication between the resulting resilient process groups. Individual processes within each group are mapped to different computers to ensure that a single attack or failure cannot impact an entire group. The base of Fig. 1 shows how the process structure responds to attack or failure: It assumes that an attack is perpetrated against processor 3, causing processes 1 and 2 to fail or to portray communication inconsistencies with other replicas within their group. These failures are detected by communication timeouts and/or message comparison. Detected failures trigger automatic process regeneration; the remaining consistent copies of processes 1 and 2 dynamically regenerate a new replica and migrate it to processors 4 and 1, respectively. As a result, the process structure is reconstituted, and the application continues operation with the same level of assurance.

This approach requires several mechanisms that are not directly available in modern operating systems. Process replication is needed to transform single processes into resilient process groups. Process migration is required to move a process from one processor to another. As processes move around the network, it is necessary to provide control over where processes are mapped. Point-to-point

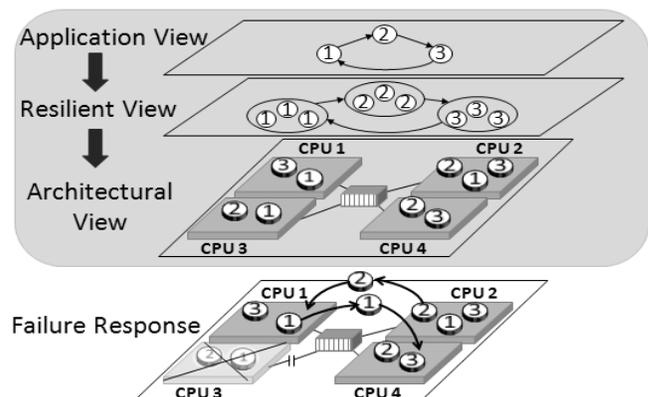


Figure 1. Dynamic process regeneration.

\*This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) under agreement number FA8750-09-1-0213.

communication between application processes must be replaced by group communication between process groups. It is desirable to maintain locality within groups of replicated processes: This allows transit delays from replicated messages to be used to predict an upper bound on the delay for failure detection timeouts. Finally, mechanisms to detect process failures and inconsistencies must be available to initiate process regeneration.

This paper describes a distributed message-passing technology to support resilience. The technology has been implemented through a Linux loadable kernel module that manages resilient process groups through multicast communication and process failure detection. This module provides a minimal MPI-like message-passing API implemented through kernel TCP functions. Multicast communications provide an environment for adaptive failure detection based on process group locality. The communication module is combined with a migration module [10] to provide process mobility. Through cooperation between the communication and migration modules, the operating system can detect and dynamically regenerate failed processes of a distributed application.

To evaluate our resilient technology, we use a distributed application exemplar to represent a broader class of applications. These evaluations demonstrate application resilience to failed processes and assess the performance impact of resilience.

## 2 Related Work

### 2.1 Distributed Application Fault Tolerance

A variety of approaches have emerged to provide *fault tolerance* for distributed applications that rely on the replication of resources. These include checkpoint/restart [11]-[13] and process replication [14]-[16]. In general, these approaches provide graceful degradation of performance to the point of failure but do not *guarantee* progress in the presence of multiple cascading and recurrent attacks.

Checkpoint/restart systems save the current state of a process to stable storage for recovery. There are many distributed checkpoint/restart systems [11]-[13], [17] closely tied with the Message Passing Interface (MPI). These systems emphasize coordination of distributed process checkpoints to achieve a consistent global state of an application. In contrast, our technology does not require a global checkpoint: We use resilient process groups to provide fault tolerance, avoiding the overhead of global coordination protocols.

Process replication systems provide fault tolerance through redundant execution. P2P-MPI [14] and VolpexMPI [15] are two MPI libraries that transparently replicate MPI processes for fault tolerant execution. P2P-MPI uses a gossip-protocol to detect *node failures* so that failed nodes can be removed from the computation. VolpexMPI emphasizes performance optimization by allowing applications to progress at the speed of the fastest replica.

The key difference between these approaches and ours is that these libraries use static replication. No attempt is made to dynamically recover failed processes, so multiple failures may ultimately cause application failure.

Dynamic process replication has been proposed to provide recovery of failed processes within distributed applications. MPI/FT is an MPI middleware that implements redundancy with process monitoring and dynamic process recovery [16]. Unfortunately, the approach requires a central coordinator and does not scale well. SCPlib, the predecessor to the proposed approach, provides a distributed dynamic process regeneration solution [8]-[9]. However, because SCPlib places the burden of resilience on the application programmer, it is not practical. In contrast, our approach applies these concepts in the operating system for transparent and automatic application resilience.

### 2.2 Distributed Failure Detection

Several protocols have been proposed for distributed failure detection. Many are based on a heartbeat mechanism, in which processes periodically send heartbeat messages to indicate liveness [18]-[20]. In contrast to these approaches, we conduct failure detection using application messages. This tactic avoids the need for an additional heartbeat message. In addition, we have the capability to detect compromised processes through message inconsistencies.

Regardless of the protocol, communication timeouts are a common means for detecting failed processes. Traditionally, timeouts were determined by a fixed delay. However, Chandra and Toueg [21] demonstrated that fixed delays are unreliable because of variations in processor loads and network traffic.

More recent work in failure detection has moved to adaptive time delays. Fetzer *et al.* introduced a simple adaptive protocol in which the timeout is adjusted to the maximal delay time of prior heartbeat messages [22]. Chen *et al.* [23] and later Bertier *et al.* [24] improved on this concept by including historical message delays and network analysis to set timeouts. Unfortunately, these works revealed that the algorithms converge too slowly for reliable failure detection. Ding *et al.* proposed an alternative approach in which timeouts are determined from historical message delays alone with efficiency comparable to quality of service requirements [25]. All of these approaches rely on historical message delays to detect failures in point-to-point communications. In contrast, our approach uses the synchronized multicast messaging within process groups to provide adaptive failure detection.

Failure detection schemes within process groups have been proposed for symmetric [26] and asymmetric [27] communication configurations. Asymmetric applications are those in which a leader process is designated as the monitored process. Both approaches use *spatial multiple timeouts* in which multiple processes cooperate to monitor another process at a given time. Reliable failure detection is achieved through consensus. Our approach differs in that we use the

process group multicast messaging primitives to detect failures directly. The multicast messages from process replicas serve as an adaptive baseline to detect real-time anomalies in message latency. Process group locality provides the basis for this tactic.

### 3 Design and Implementation

Fig. 2 shows the software architecture of the technology. It consists of two Linux loadable kernel modules: A communication module and a migration module. These modules are implemented as character devices that provide services to user-level processes through system calls. The technology also utilizes user-level daemon processes and Linux kernel threads. The daemon processes are necessary for tasks that require a user-level address space, such as forking new processes. The message daemon forks processes to comprise distributed applications, and the migration daemon forks processes in which to regenerate failed processes. A Linux kernel thread is a process that is spawned by the kernel for a specific function. The TCP servers are kernel threads that receive incoming messages for the communication module. These servers require kernel privileges to access to the module functions and memory efficiently.

#### 3.1 Message-passing API

The communication module provides an MPI-like message-passing interface through kernel TCP functions, where a computation is a set of processes numbered from 0 to  $n-1$ . This minimalist API reduces the complexity of the communication state for process migration and has only three basic functions based on blocking communication:

- *msgid(&id)* --- sets *id* to be the current process identifier within the application.
- *msgsend(dest,buf,size)* – sends a message to *dest* containing the *size* bytes located at *buf*.
- *msgrecv(src,buf,size,&status)* – receives a message from *src* (or ANY) into *buf* of length *size*; *status* is a structure designating the *source* of the message and its *length*.

All functions return TRUE if successful and FALSE otherwise. In addition, as an artifact of using a character device to implement the module, two calls, *msgInitialize()* and *msgFinalize()*, are used to open and close the device and to provide a file handler for the device throughout the computation. This API can be implemented directly on top of MPI through the appropriate macros. However, the minimalist API is sufficient to support a variety of applications in the scientific community and explicitly disallows polling for messages, a major source of wasted resources. The underlying mechanisms of the API support our resiliency model through the management of process groups, multicast messaging, and failure detection. For example, the blocking *msgrecv()* function is used to provide

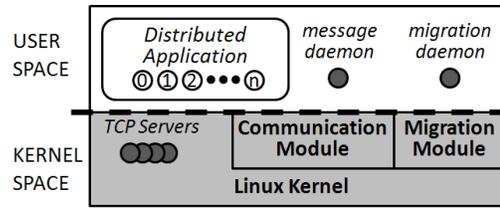


Figure 2. Software architecture of the technology.

an environment to detect process failures and trigger regeneration.

The communication module provides four primary functions for resilient applications: application initiation, message transport, failure detection, and regeneration support.

#### 3.2 Application Initiation

A distributed message-passing program is initiated through the *msgrun* program (similar to *mpirun* used in MPI) of the form:

```
msgrun -n <n> -r <r> <program> <args>
```

The program takes as arguments the number of processes to spawn for the computation (*n*), the level of resiliency (*r*), and the executable program name with arguments. The *msgrun* program loads this information into the communication module, and the module sends execution information to remote modules of the cluster. At each host, the communication module signals the message daemon to fork processes for the distributed application. Recall that the applications initiated through the *msgrun* program view *n* processes total. However, in the resilient implementation, a total of  $n*r$  processes are forked to establish *n* process groups with resiliency *r*.

A key component of application initiation is mapping processes to hosts. To demonstrate the concepts, we use a deterministic mapping to allow each host to build a consistent process map at startup. Fig. 3 shows an example mapping of *n* process groups with resiliency *r* to *m* hosts. Processes are distributed as evenly as possible across the cluster. In order to accommodate resilient process groups, replicas are mapped to maintain locality within process groups without multiple replicas on the same host. The map is constructed by assigning replicas to hosts in ascending order. For more sophisticated mapping strategies, see [28]-[31].

On initiation, an array is allocated at each communication module to store necessary information on application processes. This *process array* stores, for each process, the current mapping, the process id, the process replica number, and a *communication array* to track which

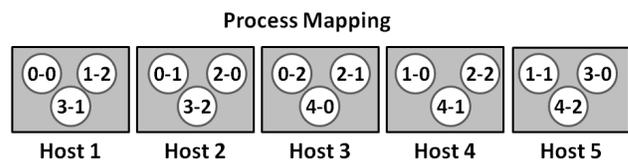


Figure 3. Mapping  $n=5$  process groups with resiliency  $r=3$  to  $m=5$  hosts.

hosts have sent or received messages from the process. The *msgid(&id)* call performs an *ioctl()* system call on the module to copy the process id and the total number of processes into process memory. Consistent with the application view of Fig. 1, application processes are unaware of replication. The total number of processes reported is actually the number of *process groups*.

### 3.3 Message Transport

The communication module provides message transport for local and remote application messages. In the resilient model, point-to-point communication between application processes becomes multicast communication between process groups, as shown in Fig. 4. The multicast messaging protocols are transparent to the application. The *msgsend()* call performs an *ioctl()* on the module. For each call, the communication module iterates through a loop to send one application message to each replica of the destination process group. For each replica, the module determines whether the destination process is local or remote by referencing the *process array*. Local messages are placed on a message queue in kernel memory. Remote messages are sent via kernel-level TCP sockets to the appropriate host.

The *msgrecv()* call performs an *ioctl()* on the module to search the kernel message queues for the specified message. For each call, the communication module iterates through a loop to locate *r* application messages from the source process group. The first message located in a queue is copied to the designated user-space buffer. However, the *msgrecv()* call blocks until *r* messages are received. If a message is missing from the kernel queue, the module places the process on a wait queue until all messages are received or until a maximum wait time has elapsed. In this way, the *msgrecv()* function serves as an implicit synchronization point for process groups and provides a setting for failure detection.

### 3.4 Failure Detection

The failure detection protocol is currently designed to detect process failures through communication timeouts. Within a *msgrecv()* call, a local timestamp is stored to indicate when each message is retrieved from the kernel queue. Processes that have waited the maximum time to receive messages initiate the failure detection protocol. In this protocol, two conditions indicate a process failure. First, the destination process must have already received *r-1* messages. In other words, only one message is missing. Second, the average time elapsed since the other messages timestamps must be greater than the prescribed timeout. If these two conditions are met, the destination process triggers process regeneration for the latent message's source. In the current implementation, the timeout is set nominally at one second.

To trigger process regeneration, the destination process performs two actions. A process failure message is sent to the host of the failed process. A process regeneration message is sent to the host of the lowest live replica number of the failed

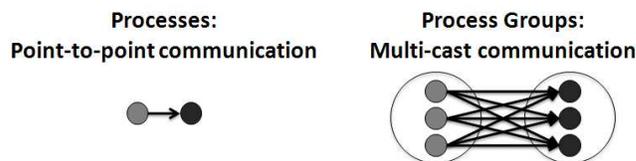


Figure 4. Point-to-point communication between processes becomes multicast communication between process groups.

process. If either the failed or replicating process is local, a signal is used instead. After triggering regeneration, the destination ignores the missing message and proceeds with the computation.

### 3.5 Process Regeneration

The communication module provides support for process regeneration, including cleaning up after failed processes, sending the packaged process image for regeneration, and resolving communication for the regenerated process at its new location.

If a module receives a message indicating a process failure, it cleans up the process structures. If the process exists in any state, it is killed, and the process descriptor is deleted. All outgoing communication sockets for the process are closed, and any messages in the kernel queues are deleted. This approach encompasses multiple causes of failure, such as CPU and socket failures, without detecting the failure itself.

The communication module that hosts a process replica receives a message to regenerate the process. The next time the replicating process makes a *msgrecv()* call, it is interrupted for replication. The actual process replication and regeneration is accomplished by the migration module. On the host of the live replica, the migration module packages a copy of the replica process image into a buffer. The communication module selects the destination host for the regenerated process. A simple selection policy is used to prevent mapping members of the same process group to the same machine while maintaining locality. As shown in Fig. 5, the destination host selected is the nearest machine, in ascending order, unoccupied by a member of the regenerated process's group.

After the migration module has packaged the process image, the communication module executes a regeneration protocol. First, the local process map is updated to reflect the pending regeneration of the failed process at its new location. Second, update messages are sent to remote modules stored in the *communication array*. These update messages contain the

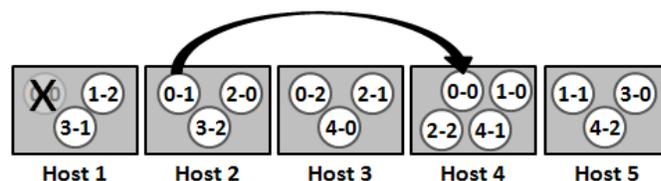


Figure 5. Dynamic process regeneration: Process 0-0 is regenerated on host 4.

process id, the process replica number, and the new host to update remote process maps. In addition, the update messages allow those processes waiting for messages from the failed process to progress until regeneration is complete. Third, all messages for the replicating process in the kernel message queue, are duplicated and forwarded to the destination host for the regenerated process. Finally, the process image is sent to the destination host.

When the packaged process image is received by the destination host, the migration module restores the process image. The regenerated process resumes execution by repeating the original *msgrecv()* call that was interrupted for regeneration. For more detail on the process replication and migration mechanisms, see [10].

Application messaging for failed processes may occur simultaneously within the regeneration protocol. Messages for the regenerated process may arrive at its original host after it has failed or at the destination host before the process update messages are received. Two features of the technology guarantee that these messages eventually reach the destination process: automatic message forwarding and reactive process update messages. The communication module forwards messages for failed processes after regeneration and sends reactive updates to ensure the remote process map is revised. These features enable guaranteed message delivery for failed or migrated processes without global coordination protocols. For more detail on the communication module support for messages-in-transit during process migration, see [10].

## 4 Experimental Evaluation

To evaluate the technology, preliminary benchmarks were conducted using a domain decomposition exemplar. These benchmarks serve two purposes. The first is to evaluate performance impact of the resilient message-passing technology. The second is to demonstrate application resilience in the presence of process failures.

The Dirichlet exemplar is a simple problem taken from fluid dynamics that is representative of a wide class of applications solved with domain decomposition [32]. It involves a large, static data structure and uses an iterative numerical technique over a cellular grid that converges to the solution of Laplace's Equation. Our parameterization solves the Dirichlet problem using a two-dimensional decomposition in which nearest-neighbor dependencies are resolved through communication with neighboring partitions.

The primary metrics used to assess performance are the average execution time of the application and the overhead of resilience. The overhead of resilience is the percentage increase in average execution time of applications with replication. The metrics used to demonstrate application resilience are associated with the completeness and accuracy of failure detection. Completeness refers to the percentage of failures that are detected, and accuracy refers percentage of false positive detections. In addition, we measure the overhead of process failures as the difference in average

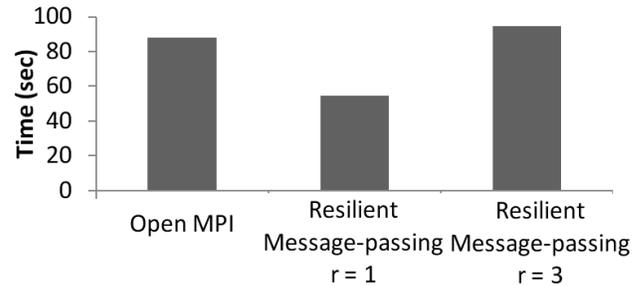


Figure 6. Average execution times using Open MPI, resilient message-passing without replication ( $r = 1$ ), and resilient message-passing with triple resiliency ( $r = 3$ ).

execution time of resilience applications with and without failures.

These benchmarks were executed on a dedicated Dell PowerEdge M600 Blade Server with 8 hosts. Each host has dual Intel Xeon E5440 2.83GHz processors and 16 GB of memory. The hosts are connected by a 1 Gbps Ethernet network. The operating system is Ubuntu 10.04.01 LTS Linux 2.6.32-26 x86\_64. The cluster also has Open MPI v. 1.4.1 for comparison with a standard message-passing system. For each test, 32 processes, or process groups, were spawned.

### 4.1 Technology Performance without Failures

The performance of the resilient message-passing technology was evaluated to assess the impact of process replication. In addition, the performance was compared to Open MPI to ensure that the technology does not incur prohibitive overhead. Fig. 6 shows the average execution times of the exemplar using Open MPI, using resilient message-passing without replication ( $r = 1$ ), and using resilient message passing with triple resiliency ( $r = 3$ ). Without replication, the resilient MP outperforms Open MPI. The Open MPI execution time is 61% longer than resilient message-passing.

The overhead of triple resilience is approximately 73%. This overhead is a result of both multicast communications and redundant computations. In multicast messaging with triple resiliency, a single application message corresponds to 9 resilient messages. The resilient process groups in these benchmarks result in a total of 96 processes running on 64 cores, overloading the system resources. These results are consistent with the priority placed on resilience in the technology development. To date, no optimizations have been made which could compromise resilience.

### 4.2 Application Resilience with Failures

To demonstrate application resilience, the exemplar was executed with process failures. In each execution of the application, one process was manually killed from the shell console. Fig. 7 shows the average execution times of the exemplar using triple resiliency with and without a single failure. In each test, the application operates through the process failure in order to complete the computation. The

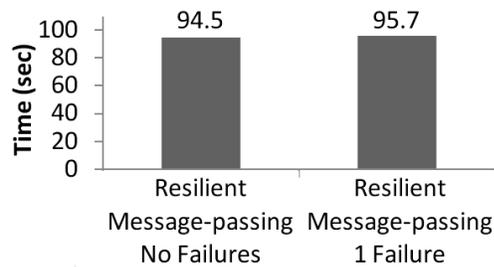


Figure 7. Average execution times using triple resiliency with and without a single failure.

overhead of a process failure is 1.2 seconds, which is less than 2% of the execution time. This overhead is attributed to the one second timeout to detect the process failure and the duration of the process regeneration protocol.

The failure detection protocol detects 100% of process failures in these tests. As a result, each failed process is dynamically regenerated, and the application continues with reconstituted resilience. In addition, there are no false positives detected in the failure tests or during the benchmarks of the technology without failures.

## 5 Conclusions and Future Work

This paper describes a resilient message-passing technology for mission-critical distributed applications. The technology is comprised of operating system support for resilience. The communication module is a minimal MPI alternative that provides a resilient message-passing API, detects process failures, and resolves message transport for regenerated processes. The migration module packages a process image into a kernel buffer, sends the buffer to a remote machine, and restores the process execution at the new location. This technology achieves application resilience to failures automatically and transparently without global communication.

This technology is designed for applications that prioritize resilience over performance. The performance of resilient message-passing *without failures* is quantified using a distributed exemplar that represents mission-critical applications. The technology incurs 73% overhead in execution time with triple resiliency. For those mission-critical applications which cannot tolerate the performance impact, the target system must be supplemented with additional compute resources to reduce the overhead due to computational redundancy.

Application resilience is also demonstrated in the presence of failures. The failure detection algorithm detects all process failures with no false positives. This result enables dynamic process regeneration, so the application operates through failures to restore triple resiliency. The approximate overhead of process failure is 1.2 seconds for a single failure of the exemplar application.

This technology serves as a demonstration of our approach to application resilience. We continue to explore

alternative failure detection algorithms and the capability to detect compromised processes through message comparison. In addition, while we have seen evidence of resilience to simultaneous failures, we have not benchmarked these events in a controlled test. We seek to perform extensive benchmarking using multiple application exemplars, larger scales, and multiple cascading and recurring failures.

## 6 Notice

The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA) or the U.S. Government.

## 7 References

- [1] D.A. Patterson, G. Gibson, and R.H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in Proc. of the 1988 ACM SIGMOD International Conference on Management Data, 1988, pp. 109-116.
- [2] Lefurgy, X. Wang, and M. Ware, "Server-level power control," in Proc. of the International Conference on Autonomic Computing, 2007, pp. 4-13.
- [3] R. Di Pietro and L.V. Mancini, *Intrusion Detection Systems*, New York, Springer-Verlag, 2008.
- [4] G.H. Kim and E.H. Spafford, "The design and implementation of tripwire: A file system integrity checker," in Proc. of the 2<sup>nd</sup> ACM Conference on Computer and Communications Security, 1994, pp. 18-29.
- [5] J. Kaczmarek and M. Wrobel, "Modern approaches to file system integrity checking," in Proc. of the 1<sup>st</sup> International Conference on Information Technology, 2008.
- [6] N.L. Petroni, T. Fraser, J. Molina, and W.A. Arbaugh, "Copilot- a Coprocessor-based Kernel Runtime Integrity Monitor," in Proc. of the 13<sup>th</sup> USENIX Security Symposium, 2004, pp. 179-194.
- [7] R. Riley, X. Jiang, and D. Xu, "Guest-transparent prevention of kernel rootkits with VMM-based memory shadowing," in Proc. of the 11<sup>th</sup> International Symposium on Recent Advances in Intrusion Detection, 2008, pp. 1-20.
- [8] J. Lee., S.J. Chapin, and S. Taylor. "Computational Resiliency", *Journal of Quality and Reliability Engineering International*, vol. 18, no. 3, pp 185-199, 2002.

- [9] J. Lee, S. J. Chapin, and S. Taylor, "Reliable Heterogeneous Applications," *IEEE Transactions on Reliability*, special issue on Quality/Reliability Engineering of Information Systems, Vol. 52, No 3, pp. 330-339, 2003.
- [10] K. McGill and S. Taylor, "Process Migration for Resilient Applications," Technical Report TR11-004, Thayer School of Engineering, Dartmouth College, January 2011.
- [11] S. Chakravorty, C.L. Mendes, and L.V. Kale, "Proactive Fault Tolerance in MPI Applications via Task Migration," in *Proc. of HIPC 2006*, LNCS volume 4297, page 485.
- [12] S. Sankaran, J. M. Squyres, B. Barrett, A. Lumsdaine, J. Duell, P. Hargrove, and E. Roman, "The LAM/MPI checkpoint/restart framework: System-initiated checkpointing," in *LACSI*, Oct. 2003.
- [13] J. Hursey, J.M. Squyres, T.I. Mattox, and A. Lumsdain, "The design and implementation of checkpoint/restart process fault tolerance for OpenMPI," in *Proc. of IEEE International Parallel and Distributed Processing Symposium*, March 2007.
- [14] S. Genaud and C. Rattanapoka, "P2P-MPI: A peer-to-peer framework for robust execution of message passing parallel programs on grids," *Journal of Grid Computing*, Vol 5, pp 27-42, 2007.
- [15] T. LeBlanc, R. Anand, E. Gabriel, and J. Subhlok. "VolpexMPI: An MPI Library for Execution of Parallel Applications on Volatile Nodes." In *Proc. of EuroPVM/MPI 2009*, Helsinki, Finland, September, 2009.
- [16] Batchu, R., Neelamegam, J., Cui, Z., Beddhua, M., Skjellum, A., Dandass, Y., Apte, M. "MPI/FTTM: Architecture and taxonomies for fault-tolerant, message-passing middleware for performance-portable parallel," In *Proc. of the 1st IEEE International Symposium of Cluster Computing and the Grid*, Melbourne, Australia, 2001.
- [17] G. Bosilca, A. Boutellier, and F. Cappello, "MPICH-V: Toward a scalable fault tolerant MPI for volatile nodes," in *Supercomputing*, Nov. 2002.
- [18] M. Pasin, S. Fontaine, and S. Bouchenak, "Failure detection in large scale systems: a survey," in *Proc. of the IEEE Network Operations and Management Symposium Workshops*, July 2008.
- [19] C. Dobre, F. Pop, A. Costan, M. Andreica, and V. Cristea, "Robust failure detection architecture for large scale distributed systems," *Proc. of the 17th Intl. Conf. on Control Systems and Computer Science*, pp. 433-440, May 2009.
- [20] M. K. Aguilera, W. Chen, and S. Toueg, "Heartbeat: a timeout-free failure detector for quiescent reliable communication," in *Proc. of 11<sup>th</sup> International Workshop on Distributed Algorithms*, pp. 126-140, September 1997.
- [21] T. Chandra and S. Toueg, "Unreliable failure detectors for reliable distributed systems," *Journal of the ACM*, 43(2), pp. 225-267, March 1996.
- [22] C. Fetzer, M. Raynal, and F. Tronel, "An adaptive failure detection protocol," in *Proc. of the 8th IEEE Pacific Rim Symp. on Dependable Computing*, pp. 146--153, 2001.
- [23] W. Chen, S. Toueg, and M. Aguilera, "On the quality of service of failure detectors," *IEEE Trans. on Computers*, 51(5), pp. 561- 580, 2002.
- [24] M. Bertier, O. Marin, and P. Sens, "Implementation and Performance Evaluation of an Adaptable Failure Detector", in *Proc. of the 15th International Conference on Dependable Systems and Networks*, 2002, pp. 354-363.
- [25] X. Ding, Z. Gu, L. Shi, Y. Hou, and L. Shi, "A failure detection model based on message delay prediction," in *Proc. of the IEEE International Conference on Grid and Cooperative Computing*, Lanzou, China, 2009.
- [26] I. Gupta, T. Chandra, and G. Goldszmidt, "On scalable and efficient distributed failure detectors," in *Proc. of the 20<sup>th</sup> Annual ACM Symposium on Principles of Distributed Computing*, pp. 170, 2001.
- [27] X. Li and M. Brockmeyer, "Fast Failure Detection in a Process Group," in *Proc. of the Parallel and Distributed Computing Symposium*, 2007.
- [28] A. Heirich and S. Taylor "Load Balancing by Diffusion", *Proc. of 24th International Conference on Parallel Programming*, vol 3 CRC Press pp 192-202, 1995.
- [29] J. Watts, and S. Taylor, "A Vector-based Strategy for Dynamic Resource Allocation", *Journal of Concurrency: Practice and Experiences*, 1998.
- [30] K. McGill and S. Taylor. "Diffuse algorithm for robotic multi-source localization", In *Proc. of the 2011 IEEE International Conference on Technologies for Practical Robot Applications*, Woburn, Massachusetts, April 2011.
- [31] K. McGill and S. Taylor. "Operating System Support for Resilience", Technical Report TR11-003, Thayer School of Engineering, Dartmouth College, October 2010.
- [32] Taylor and Wang, "Launch Vehicle Simulations using a Concurrent, Implicit Navier-Stokes Solver", *AIAA Journal of Spacecraft and Rockets*, Vol 33, No. 5, pp 601-606, Oct 1996.

# Tamper-resistant Monitoring for Securing Multi-core Environments

Ruchika Mehresh<sup>1</sup>, Jairaj J. Rao<sup>1</sup>, Shambhu Upadhyaya<sup>1</sup>, Sulaksh Natarajan<sup>1</sup>, and Kevin Kwiat<sup>2</sup>

<sup>1</sup>Department of Computer Science and Engineering, State University of New York at Buffalo, NY, USA

<sup>2</sup>Air Force Research Laboratory, Rome, NY, USA

**Abstract** - *Complex software is not only difficult to secure but is also prone to exploitable software bugs. Hence, an intrusion detection system if deployed in user space is susceptible to security compromises. Thus, this 'watcher' of other software processes needs to be 'watched.'* In this paper, we investigate a tamper-resistant solution to the classic problem of 'Who watches the watcher?'

*In our previous work, we investigated this problem in a uni-core environment. In this paper, we design a real-time, light-weight, watchdog framework to monitor an intrusion detection system in a multi-core environment. It leverages the principles of graph theory to implement a cyclic monitoring topology. Since our framework monitors intrusion detection systems, the attack surface it has to deal with is considerably reduced. The proposed framework is implemented and evaluated using AMD SimNow simulator. We show that the framework incurs a negligible memory overhead of only 0.8% while sustaining strong, tamper-resistance properties.*

**Keywords:** Attacks, Graph models, Intrusion detection, Multi-core, Processor monitoring, Recovery, User space components

## 1 Introduction

Growing connectivity of computer networks has made network services like wu-ftp, httpd, BIND, etc. a popular target for cyber attacks. Exploitation of these services makes the entire host vulnerable to further exploits. Since more and more functionalities are added to such software programs each day, their code base becomes larger and more complex. This increases the probability of existing software bugs, resulting in security vulnerabilities. There are many studies conducted over the years that document the increasing trend of software unreliability and growing intelligence of hacker community [1], [2], [3].

Over the years, industry and research communities have produced several prevention, detection and recovery methodologies. However, preventing all kinds of attacks in today's open networking environment is practically impossible. Therefore, the major burden of securing a system effectively lies with the detection techniques employed. Detection of attacks and suspicious behavior is generally achieved with the help of automated tools such as intrusion

detection systems (IDS) [4]. Many IDS tools alert the authorized user when some malicious activity is suspected, or initiate a recovery without attempting to prevent the attack at all. Sometimes they can detect and prevent an attack before it causes any major damage – in which case, they are also called intrusion prevention systems (IPS). In this paper, we discuss intrusion detection systems but the findings equally apply to intrusion prevention systems.

Intrusion detection systems collect and analyze data at two broad levels: the network level and the host level. Host based IDSes have the advantage of proximity and hence can monitor a system closely and effectively. Traditionally, IDSes have been designed to operate in user space which makes them vulnerable to compromises. Advanced malware has been discovered that can disable them upon its installation [5], [6]. Recently, many compromises of IDSes were reported [7], [8], [9]. Moving IDSes completely or partially to the kernel space increases the trusted computing base (TCB), which in turn introduces further (and more serious) problems [10]. Therefore, a simpler and more effective way of ensuring the security of these systems is to design and deploy them in user space, and ensure their correct operation by tamper-resistant monitoring against subversion. If a malware is successful in switching off the IDS, the monitoring system should be able to report this change to an uncompromised authority. This addresses the problem of 'who watches the watcher' that often arises in an end-to-end security system.

We have earlier studied this problem in a uni-core environment [11]. However, with the advent of multi-core technology, this problem needs to be revisited for two major reasons. First, a multi-core environment presents new security and design challenges [12], [13]. Therefore, existing security solutions need to be reevaluated for adoption in this new environment. Second, it offers the concurrency that can increase the efficiency and efficacy of our previously proposed framework [11], [14].

The overall scheme works as follows. We use a cyclic, tamper-resistant monitoring framework that uses light-weight processes (referred to as process monitors in the sequel) to monitor the IDS. Though we focus on monitoring the host-IDS, this framework can generally monitor any crucial process. Thus, this framework can also assist in reducing the size of a system's TCB. Process monitors in this framework

are responsible for performing simple conditional checks. A primary conditional check is to continuously monitor if a process is up and running. Rest of the conditional checks can be implemented and enabled according to security and efficiency requirements of the system. If any of the conditional checks fail, an alert notification is sent to the uncompromised authority (mostly the root). These process monitors monitor each other in a cyclic fashion without leaving any *loose* ends. If one of them is killed, the next in the order raises an alert. Since the monitoring is cyclic, no process monitor is left unmonitored. One of these process monitors has an additional responsibility of monitoring the host-IDS. If an attacker intends to subvert the IDS, he first needs to subvert the process monitor monitoring it. Since this process monitor is being monitored by another process monitor, and so on, the subversion becomes almost impossible (we discuss the possible attack scenarios in Section V). Loop architectures and concepts from graph theory have long been used to make designs reliable and robust [15]. In this paper, we identify the benefits of a cyclic monitoring framework, the issues in maintaining it and the kind of performance overhead it incurs.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 states the assumptions. Section 4 gives the system architecture. Section 5 discusses the threat model, while Section 6 presents the various framework topologies. Section 7 describes the experiments and results, followed by conclusion and future work in Section 8.

## 2 Related Work

As discussed previously, intrusion detection systems are traditionally located in the user space. Tools like DWatch [16] are implemented completely in user space and monitor other daemon processes. Implementing IDS in user space may have its performance advantages [19] but access to these detection systems is not very well protected. Hence, they are equally susceptible to a security compromise as any other process. There are many intrusion detection systems that are implemented completely inside the kernel [17]. Kernel space implementation of an intrusion detection system is a very tempting choice because it provides strong security (process privileges required). However, such an implementation has high associated overhead. Each time a kernel-implemented IDS event is invoked, there is expensive context switching. Besides, IDS being complex software has a high probability of residual software bugs. These bugs can either cause severe negative performance impacts at kernel level, or make the entire kernel vulnerable to a security. Also, IDS tools generally need to apply frequent updates because new attacks are discovered each day. This not only results in high performance overheads but carries a risk of infecting the TCB with bad code. Implementations like the Linux security modules (LSM) [21] provide a diffuse mechanism to perform checks inside the kernel at crucial points, but it is an expensive solution. There are some hybrid detection

techniques [18] that are partially in kernel space and rest in the user space. Such techniques inherit some strengths as well as weaknesses from both the domains. For instance, an entire IDS implementation in kernel space, with just the notification mechanism in user space can be subverted by curbing the notifications.

Some monitoring architectures secure intrusion detection systems by proposing the use of isolated virtual machines [5]. Such systems solve the problem of securing the watcher, but require installation of separate virtual machines and process hooks. These features increase the cost of such security solutions.

Our framework is an extension of the user space framework proposed by Chinchani et al. [11]. They proposed a tamper-resistant framework in a uni-core environment to secure user space services. In this paper, the framework is extended to protect user space components in a multi-core environment.

## 3 Assumptions

- i) All software components are assumed to be susceptible to security attacks.
- ii) We assume a zero-trust model. Therefore, an attacker cannot predict the order of process monitors in the framework topology by observing system behavior.
- iii) All process monitors are identical and light-weight.

## 4 System Architecture

The proposed framework runs on a K-core host. We assume symmetrical multiprocessing (SMP), where all cores are managed by the same operating system instance. This system runs a user space, host-based IDS that monitors other host processes for any suspicious activity. We mentioned earlier that some malware can attack the IDSes upon their installation. Since the IDS here is in the user space, it is susceptible to security compromise as any other process. Therefore, we design a monitoring framework to ensure that the IDS is running in a tamper-resistant mode at all times.

This framework primarily consists of light-weight process monitors. These process monitors are simple programs that monitor other processes for specific conditions. The primary condition is to check continuously if the monitored process is running. If it is killed or any other condition fails, the process monitor detects it and sends an alert to the root user.

In the simplest topology of this framework, process monitors are arranged in a cyclic fashion (as shown in Figure 1). Since there are no loose ends in a cycle, every process monitor is monitored by another. To ensure parallel monitoring, IDS and all the process monitors run on separate

cores. For instance, in a  $K$ -core system, if the IDS is running on Core 1, rest of the  $K-1$  cores run the process monitors, one on each. Process monitor on Core 2 monitors the IDS on Core 1, as well as the process monitor on Core  $K$ .

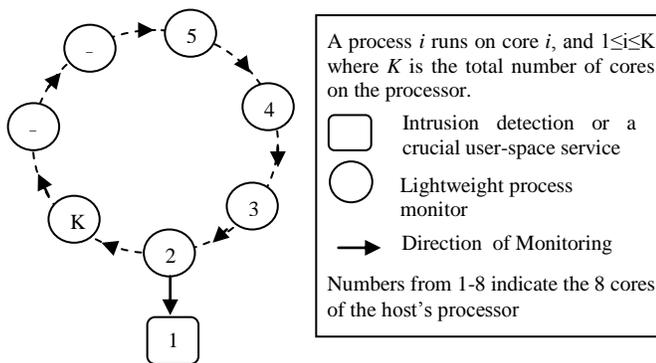


Figure1: A simple-ring topology on a  $K$ -core system.

## 5 Threat model

- i) *Denial of Service attacks in multi-core systems:* Memory hogging [12] is a denial of service attack where one core consumes shared memory unfairly. This results in performance degradation at other cores due to resource scarcity. This kind of attack can be handled by our framework via conditional checks. A process monitor can raise an alert if it observes exceptionally high scheduling delays affecting the monitored process.
- ii) *Window of vulnerability:* There are windows of vulnerability introduced because of multi-core scheduling. We assume that process monitors hosted on separate cores can continuously monitor each other. This, however, is not practically feasible. If the core hosting the process monitor has other processes scheduled on it, the process monitor will have to go back into the scheduling queue periodically. During this window, if monitored process is attacked, the process monitor monitoring it cannot raise an alert. It can only do so when it is rescheduled to run again. If such windows are identified and exploited in order, it is possible to subvert the entire framework. This vulnerability can be patched by employing multiple degree of incidence, meaning that one process monitor is monitored by (and monitors) multiple other process monitors. This way, even when some of them go back into scheduling queues, we can still dynamically maintain at least one monitoring cycle with a high probability. However, this arrangement leads to a performance-security trade-off. Higher degree of incidence provides stronger security but at a higher monitoring cost.
- iii) *Exploiting system vulnerabilities (crash attacks, buffer overflow, etc.):* An attacker can try to crash (kill) any of these process monitors by exploiting system vulnerabilities. These vulnerabilities could be introduced

by other software running on the system. We will see in Section 5 how such attacks are handled.

## 6 Framework Topologies

A monitoring framework, cyclic or not, can have numerous topologies. For a  $K$ -core system, we can choose from a simple ring topology (as shown in Figure 1) to a topology with multiple degree of incidence (as shown in Figure 2). We will present a few basic topologies here that provide strong tamper-resistance properties.

In order to compare the various topologies, we need to understand the basis on which they can be evaluated. There are two questions that can be asked:

- i) How secure a topology is?
- ii) How efficient a topology is?

Any topology that can be compromised with a high probability is insecure. In [11], Chinchani et al. discuss the subversion probabilities of simple replication and layered hierarchy (onion peel) topologies. The paper claims that a circulant digraph configuration provides the strongest tamper resistance properties.

### Topology 1: Simple Ring

Simple ring topology represents an ordered cycle of process monitors, as shown in Figure 1. It offers a much lower probability of subversion compared to the onion peel model [11]. Since we assumed a zero-trust model, an attacker needs to try  $(n!-1)$  permutations (in the worst case), before he figures out the right order. This topology works considerably well for scenarios where all participating cores are minimally loaded, and the process monitors run for most of the times. However, when the workload increases, these process monitors have to wait in scheduling queues for some finite amount of time. If during this time, the processes that they are monitoring are compromised, an alert cannot be raised. Therefore, heavy system load can create windows of vulnerability that if exploited in a certain order, can lead to successful subversion of the framework.

### Topology 2: Circulant Digraph

Earlier research proposed circulant digraph as the primary approach to increase efficacy of this monitoring framework (reduce false negatives) [11]. However, in a multi-core environment it has an added benefit of reducing the creation of windows of vulnerability. Higher the degree of incidence, lower is the probability that a process monitor remains unmonitored itself.

A circulant digraph  $C_K(a_1, \dots, a_n)$  with  $K$  vertices  $v_0, \dots, v_{K-1}$  and jumps  $a_1, \dots, a_n$ ,  $0 < a_i < \lfloor K/2 \rfloor$ , is a directed graph such that there is a directed edge each from all the vertices  $v_i \pm a_j \pmod K$ , for  $1 < i < n$  to the vertex  $v_j$ ,  $0 < j < K - 1$ . It is

also homogeneous, i.e., every vertex has the same degree (number of incident edges), which is  $2n$ , except when  $a_i = K/2$  for some  $i$ , when the degree is  $2n-1$ . Figure 2 shows a circulant digraph with 8 process monitors, degree of incidence 3 and jumps  $\{1, 2\}$ .

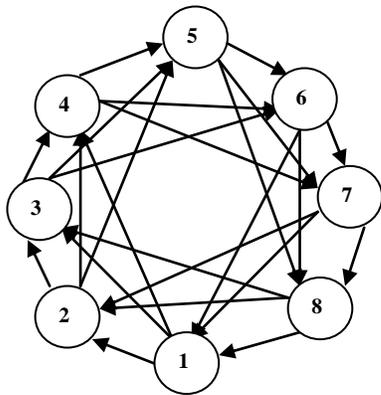


Figure 2: Circulant digraph with 8 process monitors running on 8 cores. One process monitor per core. This circulant digraph has a degree of incidence 3 and jumps  $\{1, 2\}$ .

Simple ring topology is a special case of circulant digraph topology with degree of incidence 1. However, a circulant digraph topology (with degree of incidence  $> 1$ ) is much more secure than a simple ring topology. This is because the number of attempts required to find the right permutation increases exponentially in the worst case (since the attacker does not know the degree of incidence, the jump and the order of process monitors).

**Topology 3: Adaptive Cycle**

Since raising a large number of alarms is counter-productive to a system's performance, a circulant topology though effective, is not optimal. Even if an attacker is not in a position to attack, he can tamper with the framework to make it raise a large number of useless alerts. To counter this threat and reduce the number of alerts produced by the circulant topology, we propose an adaptive topology. It predicts the system load and tries to maintain cyclic monitoring at all times. This requires that process monitors at each core track the load on other cores. As shown in Figure 3, the initial state of this topology is set to be a simple ring. If process monitor on core 2 realizes that core K has just been assigned a lot of new processes, it starts monitoring process monitors K and K-1, both. Similarly, if core 2 gets heavily loaded, process monitor 3 starts monitoring process monitors 2, K and K-1. So, the cores that are lightly loaded take up the responsibility of monitoring the process monitors on heavily loaded cores and the process monitors they were respectively responsible for.

Therefore, the final state of an adaptive cyclic topology can be formally defined: For a  $K$ -core processor, a process monitor on core  $i$  where  $1 \leq i \leq K$ , monitors process monitors on

all cores  $j$ , where  $i+1 \leq j \leq K$ , if there is a directed edge from  $i$  to  $j$ , or if there exists a  $z$  such that there is a directed edge from  $i$  to  $z$  and from  $z$  to  $j$ .

The probability of subversion for adaptive cycle topology is equal to the probability of subversion for circulant digraph topology. However, the number of attempts required to find the right order of process monitors in an adaptive topology is much larger than in circulant digraph topology (in the worst case). This is because the degree of incidence and jumps are always changing dynamically. Therefore, an adaptive topology provides better performance and stronger security as compare to the circulant digraph topology.

**7 Experiment Design**

Companies like Intel, AMD, etc., have made significant progress in multi-core technology. Clearspeed's CSX600 processor with 96 cores [19] and Intel's Teraflops Research chip with 80 cores [20] are the latest in this line. However, such systems do not have a strong presence in the commercial market yet. This generally restricts researchers to use a small number of 2-6 cores. In order to bridge this gap between unavailability of present technology and researching the future needs of this technology, simulators have been developed [21], [22]. These simulators emulate the functioning of a multi-core platform on a system with lesser number of cores (even uni-core processor). Amongst the many open source multi-core simulators that are available today, AMD SimNow closely emulates the NUMA architecture. Therefore, we use it as a test-bed to experiment with simple ring and circulant digraph topologies.

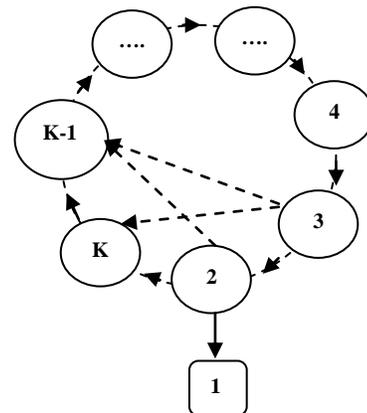


Figure 3: Adaptive topology when cores 2 and K are heavily loaded.

**7.1 Configuration**

Experiments are conducted on Intel Pentium Core2Duo 2.1 Ghz processor with 4GB RAM. AMD SimNow is installed on Ubuntu 10.04 which is the host operating system. Inside AMD SimNow, we run a guest operating system, i.e., FreeBSD 7.3. All experiments run on this guest operating system. This system is configured to use emulated hardware

of AMD Awesim 800Mhz 8-core processor with 1024 MB RAM.

We use kernel level filters to implement process monitoring. This is because inter-process communication support provided by UNIX-like systems (like pipes or sockets) does not suffice for our framework. Inter-process communication delivers messages only between two live processes. However, we require that a communication (alert) be initiated when a process is terminated. For this purpose, we use an event delivery/notification subsystem called Kqueue, which falls under the FreeBSD family of system calls. Under this setup, a process monitor interested in receiving alerts/notifications about another process creates a new kernel event queue (kqueue) and submits the process identifier of the monitored process. Specified events (kevent) when generated by the monitored process are added to the kqueue of the process monitor. Kevent in our implementation is the termination of the monitored process. Process monitors can then retrieve this kevent from their kqueues at any time. A process monitor can monitor multiple processes in parallel using POSIX threads.

Experimental setup consists of 8 simulated cores with process monitors running on each one of them. We report on the evaluation of only the circulant digraph topology as a representative result. We experiment with different circulant digraph topologies with varying number of process monitors and degrees of incidence. The primary performance metrics in a multi-core system are time and memory overheads. Each reading in this analysis represents an average of 100 runs.

## 7.2 Execution Performance

The initial setup time is defined as the time taken for the kqueue subsystem to get loaded before an attacker tries to subvert the process monitors. This is the only major time delay this system has been observed to incur. As shown in Figure 4, initial setup time increases linearly with increasing degree of incidence. With 8 process monitors in a circulant digraph topology, the worst case initial setup delay of 0.3ns is obtained with a maximum degree of incidence (i.e., 7).

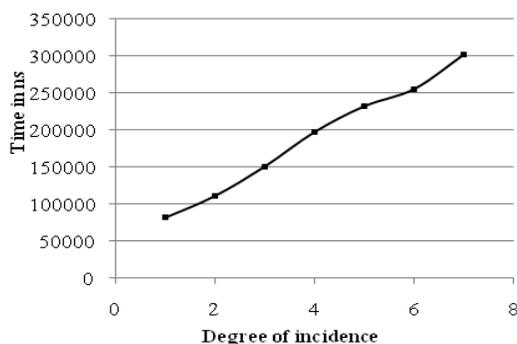


Figure 4: Initial Setup overhead for circulant digraph topology with 8 process monitors.

## 7.3 Memory Performance

We define the memory overhead to be the amount of memory consumed by a running instance of the framework as a percentage of the entire system memory capacity. Memory overhead is observed to increase linearly with the degree of incidence. A circulant topology with 8 process monitors and degree of incidence 7 incurs a 0.8% (0.1% per process) memory overhead.

Table 1: Categorization of circulant digraph topologies

Configuration	Number of processes	Degree of Incidence
Series1	2	1
Series2	3	1,2
Series3	4	1,2,3
Series4	5	1,2,3,4
Series5	6	1,2,3,4,5
Series6	7	1,2,3,4,5,6
Series7	8	1,2,3,4,5,6,7

## 7.4 Attack Tolerance

We experimented with different circulant digraph topologies with varying number of process monitors and degree of incidence, as shown in Table 1. For all topologies, jumps start from a minimum of 1, incremented by 1, until it satisfies the degree of incidence.

The following attack scenarios were executed in order to test the security strength of the framework.

### Experiment 1: Killing process monitors without delay (under light system load)

We experiment with the worst case scenario where the attacker already knows the correct order of the nodes in this topology. We assume that he also identifies the windows of vulnerability and uses them to his advantage (again, the worst case). In Figure 5, the number of alerts generated shows the sensitivity of this framework toward a crash attack executed using SIGKILL.

### Experiment 2: Killing process monitors without delay (under heavy system load)

Experiment 1 was repeated under heavy load conditions to determine the impact of increasing system load on framework's sensitivity (number of alerts) to an attack. A heavy load condition is simulated by running Openssl benchmark in the background. In this emulated multi-core environment, a maximum of 6,164 processes can run on FreeBSD operating system. We ran 6,000 processes to achieve nearly 100% CPU consumption for all cores. As seen in Figure 6, the framework generates lesser number of alerts. This is because the process monitors have to wait in the scheduling queue longer than in Experiment 1.

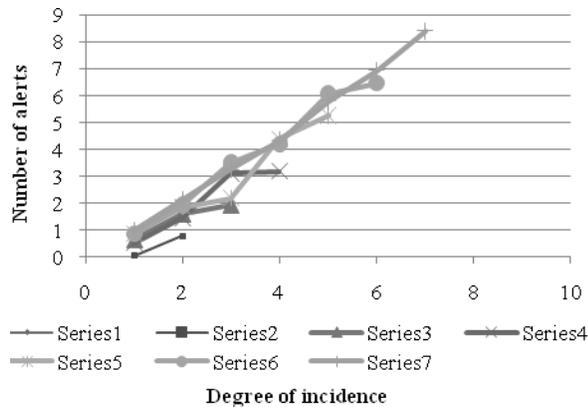


Figure 5: Alerts generated for killing process monitors in sequential order without delay, under light system load.

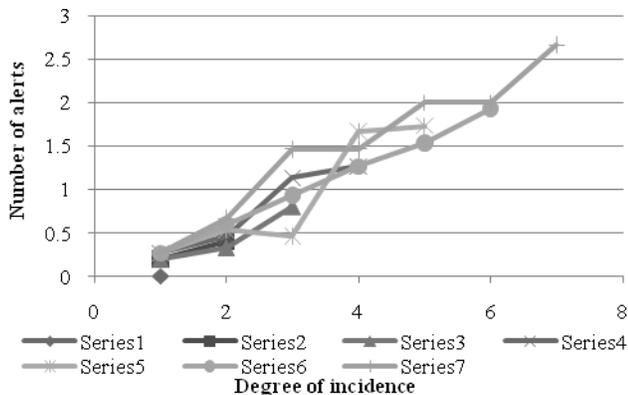


Figure 6: Alerts generated for killing process monitors in sequential order without delay, under heavy system load.

### Experiment 3: Group Kill attack

This framework is created by forking a process into child processes. All child processes forked from the same parent belong to the same group by default (identified by the same group ID). An external process can easily identify the group ID (GID) from the kernel proc structure using commands such as 'ps' from the user space. Any crash attack on this process monitor group can be represented by a SIGKILL signal sent to the GID of the process monitors. This attack successfully subverts the framework and no alerts are raised by any of the process monitors. Thus, this constitutes a successful attack on the framework, where the property of all the process monitors belonging to a common default group becomes a vulnerability.

In order to increase framework's resistance to these kinds of attacks, we organized alternate process monitors under two different groups. Process monitors with even PIDs (process IDs) retain their default GID, which is the PID of the parent process. The GID of process monitors with odd PIDs is changed to their respective PIDs. Now, a SIGKILL signal

sent to the default GID of the group will successfully kill the processes with even PIDs, but the odd ones will raise alerts.

## 8 Conclusion and Future work

This paper proposed a tamper-resistance framework to monitor the intrusion detection systems (IDS) in a multi-core environment. We identified the benefits of our framework and the related issues. We also analyzed two framework topologies, viz. simple ring and circulant digraph. They are found to incur low time and memory overhead, while still retaining strong tamper-resistance properties.

As a future work, we plan to investigate the adaptive ring and other topologies. We plan to add more attack scenarios to this analysis. For instance, a smart attacker can replace a process monitor with a dummy process to subvert the framework.

## 9 Acknowledgement

This research is supported in part by a grant from the Air Force Office of Scientific Research (AFOSR). The work is approved for Public Release; Distribution Unlimited: 88ABW-2011-1929 dated 31 March 2011.

## 10 References

- [1] A. Nhlabatsi, B. Nuseibeh, and Y. Yu, "Security requirements engineering for evolving software systems: A survey," *Journal of Secure Software Engineering*, vol. 1, pp. 54-73, 2009.
- [2] N. Dulay, V. L. Thing, and M. Sloman, "A Survey of Bots Used for Distributed Denial of Service Attacks," *International Federation for Information Processing Digital Library*, vol. 232, 2010.
- [3] T. Heyman, K. Yskout, R. Scandariato, H. Schmidt, and Y. Yu, "The security twin peaks," *Third international conference on Engineering secure software and systems*, 2011.
- [4] F. Sabahi, and A. Movaghar, "Intrusion Detection: A Survey," *Third International Conference on Systems and Networks Communications (ICSNC)*, pp. 23-26, 2008.
- [5] B. D. Payne, M. Carbone, M. Sharif, and L. Wenke, "Lares: An Architecture for Secure Active Monitoring Using Virtualization," *IEEE Symposium on Security and Privacy (SP)*, pp. 233-247, 2008.
- [6] T. Onabuta, T. Inoue, and M. Asaka, "A Protection Mechanism for an Intrusion Detection System Based on Mandatory Access Control," *Society of Japan*, vol. 42, 2001.
- [7] Greg Hoglund, "Malware commonly hunts down and kills anti-virus programs," *Computer Security Articles* 2009.
- [8] Hermes Li, "Fake Input Method Editor(IME) Trojan," *Websense Security Labs*, 2010.

- [9] Christopher Null, "New malware attack laughs at your antivirus software," *Yahoo! News*, 2010.
- [10] M. M. Swift, B. N. Bershad, and H. M. Levy, "Improving the reliability of commodity operating systems," *ACM Transactions on Computer Systems (TOCS)*, vol. 23, pp. 77-110, 2005.
- [11] R. Chinchani, S. Upadhyaya, and K. Kwiat, "A tamper-resistant framework for unambiguous detection of attacks in user space using process monitors," *First IEEE International Workshop on Information Assurance (IWIAS)*, pp. 25-34, 2003.
- [12] T. Moscibroda, and O. Mutlu, "Memory performance attacks: denial of memory service in multi-core systems," *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium (SS)*, 2007.
- [13] C. E. Leiserson, and I. B. Mirman, "How to Survive the Multicore Software Revolution (or at Least Survive the Hype)," *Cilk Arts Inc.*, 2008.
- [14] S.P. Levitan and D. M. Chiarulli, "Massively parallel processing: It's Déjà Vu all over again," *46th ACM/IEEE Design Automation Conference (DAC)*, pp. 534-538, 2009.
- [15] S.L. Hakimi, and A. T. Amin, "On the design of reliable networks," *Networks*, vol. 3, pp. 241-260, 1973.
- [16] U. Eriksson, "Dwatch - A Daemon Watcher," <http://siag.nu/dwatch/>, 2001.
- [17] C. Wright, C. Cowan, S. Smalley, J. Morris, and G. Kroah-Hartman, "Linux Security Modules: General Security Support for the Linux Kernel," in *Foundations of Intrusion Tolerant Systems (OASIS)*, pp. 213, 2003.
- [18] N. Provos, "Improving host security with system call policies," *Proceedings of the 12th conference on USENIX Security Symposium*, 2003.
- [19] Y. Nishikawa, M. Koibuchi, M. Yoshimi, A. Shitara, K. Miura, and H. Amano, "Performance Analysis of ClearSpeed's CSX600 Interconnects," *IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp. 203-210, 2009.
- [20] Intel, "Teraflops Research Chip " <http://techresearch.intel.com/ProjectDetails.aspx?Id=151>.
- [21] J.E. Miller, H. Kasture, G. Kurian, C. Gruenwald, N. Beckmann, C. Celio, J. Eastep, and A. Agarwal, "Graphite: A distributed parallel simulator for multicores," *Sixth IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 1-12, 2010.
- [22] A. Vasudeva, A. K. Sharma, and A. Kumar, "Saksham: Customizable x86 Based Multi-Core Microprocessor Simulator," *First International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 220-225, 2009.

# Observation from Microsoft Zero-Day Vulnerability Examples

Nathaniel Evans and Xiaohong Yuan

Computer Science Department, North Carolina A&T State University, Greensboro, NC, USA

**Abstract** - *Zero-Day vulnerabilities are an intriguing and ever increasing problem. Microsoft has been one of the more exploited companies having Zero-Day vulnerabilities. This paper intends to identify some relationships within the Zero-Day vulnerabilities identified in nineteen news articles from 2010. We tried to collect data on vulnerability report date, attack report date, vulnerability patch date, vulnerability life cycle category, exploit implemented, Microsoft product affected, and affected functionality. Based on this data, we analyzed the duration between vulnerability notification and attack dates, the distribution of different vulnerability life cycle categories, the most common Microsoft product affected, and the most common exploitation technique used. Our data shows that Potential for Attack (POA) is the most common vulnerability life cycle category, Windows XP SP3 is the most affected system, and the most common exploitation technique is by finding Back/Trap doors.*

**Keywords:** Zero-Day attacks, vulnerability life cycle category, attack patterns, white-hat hacker, and black-hat hacker

## 1 Introduction

Zero-Day vulnerabilities have occurred for years. Microsoft has been one of the more exploited companies having numerous Zero-Day vulnerabilities within this year alone. Zero-Day vulnerability refers to a software vulnerability being unknown to others or software developers before a white-hat or black-hat hacker discovers an exploit [1]. Zero-Day vulnerabilities have been identified in various forms of malicious actions. Worms, Trojans, Viruses, are not the only exploitation of software used by hackers. Even though many people will commonly think of these type of attacks, exploits of Zero-Day vulnerabilities are often more specific to the software functionality. This can range from changing various directories to familiarizing themselves with and manipulating vital computer files.

In this paper we performed preliminary analysis on a collection of articles on recent Microsoft Zero-Day vulnerabilities. We intend to discover the characteristics of these Zero-Day vulnerabilities, such as which vulnerability life cycle category is most common, which Microsoft products are most affected, and which exploitation techniques are implemented most often.

The goal of this paper is to observe a trend or pattern amongst the occurrence of Zero-Day vulnerabilities and the steps taken to secure them. We perform analysis based on vulnerability life cycles [2]. Initially, we limited our time to three weeks in the fall of 2010, in order to gather articles highlighting Zero-Day vulnerabilities, and to use these articles to determine if this area of research would begin to demonstrate any promising potential for future endeavors. However, we have not gathered sufficient information to analyze vulnerability life cycles across different software products belonging to the different software companies. An example could be analyzing the internet browsers of Apple's Safari, Google's Chrome, and Microsoft's Internet Explorer. Another would be analyzing the operating systems of GNU Project's Linux and Microsoft's Windows [2]. Instead, we have focused on Microsoft and its Internet related software products due to Microsoft having the most occurrences out of all of the companies mentioned throughout the articles found.

This paper will progress as follows. Section 2 identifies the criteria used in our analysis. Section 3 presents the preliminary analysis result. Section 4 compares related work to the results found in this paper. Section 5 concludes the paper.

## 2 Relating Zero-Day Vulnerabilities

Articles were located online indicating Zero-Day vulnerabilities [3-21]. The websites included in the analysis were Computerworld, PC World, and C|NET, ZDNet, the Tech Herald, and the Register. At times, articles found from each website, would point to the same vulnerability. Therefore, we also had to make sure each article highlighted a different Zero-Day vulnerability from the others before proceeding. We analyzed the Zero-Day vulnerabilities in terms of the following criteria.

### 2.1 Vulnerability Report Date

In some cases, it is hard to find out the vulnerability report dates since black-hat hackers do not notify anyone of the vulnerability. Therefore the vulnerability report date is best determined as the date of the article. In other cases, the vulnerability report date can be pinpointed to an exact day due to white-hat hacker notifying the appropriate software developer of the vulnerability.

## 2.2 Attack Report Date

Unless otherwise stated in the given article, the date the attack was reported will be the same date as the article.

## 2.3 Vulnerability Patch Date

This is the date that a vendor releases a patch for a given vulnerability after being notified of the Zero-Day attack. However, there are instances where the date cannot be ascertained due to one of the following factors: First, at the time of the article, a patch was not yet developed. Second, at the time of the article, the affected company, received or created the patch but were unable to figure out how to apply it. Last, at the time of the article, an effort to develop a patch was not mentioned.

## 2.4 Vulnerability Life Cycle Category (VLCC)

Jumratjaroenvanit and Teng-amnuay [2] classify vulnerability life cycle into five categories: Zero-Day Attack (ZDA), Pseudo Zero-Day attack (PZDA), Potential for Pseudo Zero-Day attack (PPZDA), Potential for Attack (POA), and Passive Attack (PA). Below lists the adaptations used in this paper.

*Zero-Day attacks* are the actual attacks when vulnerability is discovered by a hacker, the hacker exploits the vulnerability to achieve his or her goal, finally the software developer, previously unaware, is notified of the vulnerability. The goal of a hacker usually depends on whether it is a white-hat hacker for improving the source code or a black-hat hacker for his or her own malicious objective.

*Pseudo Zero-Day attacks* are actual attacks when vulnerability was already discovered, the software developer already had the knowledge of the vulnerability, had the ability to release a patch, however the patch was not applied by the administrators. Thus, a black-hat hacker exploited the given vulnerability in an unknown or unpredicted time frame after the software developer was able to apply a patch for the vulnerability.

*Potential for Pseudo Zero-Day attacks* are potential attacks similar to Pseudo Zero-Day attacks. However, a black-hat hacker has not yet exploited the vulnerability.

*Potential for attacks* are potential attacks when vulnerability is discovered by a white-hat hacker and the software developer is immediately notified in order to begin creating a patch. However from the time of the notification of the vulnerability until the present time, an exploitation of the vulnerability has not yet occurred by a black-hat hacker.

*Passive attacks* are the attacks in which white-hat hackers have declared a given vulnerability, but do not have exploit code readily available.

## 2.5 Exploit Implemented

This is the type of intrusion chosen and used by a given white-hat or black-hat hacker. White-hat hackers are commonly known to develop attack code that exploits vulnerability in software in order to educate and enlighten the software developer. These hackers usually find ways to exploit vulnerabilities in the functionality of the software. This is because most white-hat hackers are researchers either on behalf of a company or themselves. Black-hat hackers are commonly known to develop attack code that exploits vulnerability with their malicious motives. These hackers tend to be individuals not representing any company. With the exception of the Stuxnet Worm, the exploits created by black-hat hackers are often not as sophisticated as white-hat hackers.

## 2.6 Microsoft Products Affected

This is simply a list of the Microsoft software that was affected due to the exploitation of the given vulnerability. This includes the various operating systems or web related applications.

## 2.7 Affected Functionality

This indicates the functionality affected by the vulnerability and explains how the vulnerability was exploited by a hacker.

# 3 Analysis of Reported Microsoft Vulnerabilities

This preliminary analysis contains 19 articles dated in the year 2010, ranging from January to December, which identify Zero-Day vulnerability in Microsoft products. Next, data based on criteria from Section 2 is collected from the articles and shown in Table 1 and Table 2. It is important to realize that these articles do not specify the vulnerability life cycle category.

Thus, the means to determine a category go as follows. First, determine how the vulnerability was found. If it was by a black-hat hacker, ZDA, otherwise it begins as a POA found by a white-hat hacker who immediately notifies Microsoft. As time goes by, the next thing is identify Microsoft's response to the vulnerability. If they mention immediately beginning to work on a patch, then it remains a POA. However, if they delay working on a patch, then it becomes a PPZDA. Last, if a PPZDA is exploited by a black-hat hacker, then it is upgraded to a PZDA.

Based on this data, we analyzed the duration between vulnerability notification date and attack date, the distribution of different vulnerability life cycle categories, the most common Microsoft product affected, and the most common exploitation technique used.

**Table 1.** Vulnerability report date, attack date, patch date, and vulnerability life cycle category

REF #	DATES			VLCC
	Reported	Attacked	Patched	
[3]	9/22/2010		9/15/2010	PZDA
[4]	6/17/2010	7/15/2010		ZDA
[5]	9/17/2010		9/28/2010	PPZDA
[6]	9/14/2010			POA
[7]	7/1/2010			PPZDA
[8]	6/5/2010			POA
[9]	1/14/2010			ZDA
[9]	1/14/2010			POA
[10]	6/10/2010			POA
[11]	6/10/2010	6/16/2010		ZDA
[12]	7/20/2010			POA
[13]	4/12/2010			POA
[14]	8/19/2010			PPZDA
[15]	7/7/2010			PZDA
[16]	7/19/2010			PPZDA
[17]	3/1/2010			PPZDA
[18]	8/30/2010			POA
[19]	3/25/2010		4/13/2010	POA
[20]	12/23/2010			POA
[21]	12/22/2010			POA

**Table 2.** Exploit techniques and affected Microsoft products

REF #	EXPLOIT TECHNIQUE	MICROSOFT PRODUCTS
[3]	Malicious Code	XP, Vista, Server 2008
[4]	Physical PC interaction	XP, Vista, Server 2008, 7
[5]	Back/Trap Door	ASP.NET
[6]	Malicious Code	Server 2008, 7 w/ patch; < XP SP3 w/o patch
[7]	Physical PC interaction	Vista, Server 2008
[8]	Pharming	XP, Server 2003
[9]	Back/Trap Door	2000, XP, Server 2003, Vista, Server 2008/R2, 7, IE6, IE7, IE8
[10]	Pharming	2000, Server 2003
[11]	Pharming	XP
[12]	Back/Trap Door	2000, XP SP2, 7
[13]	Malicious Code	SharePoint 2007
[14]	Back/Trap Door	all Microsoft Operating System versions
[15]	Back/Trap Door	XP, IE, Internet Information Services
[16]	Malicious Code	all Microsoft Operating System versions
[17]	Malicious Code	2000, XP, Server 2003
[18]	Back/Trap Door	XP, Vista, 7
[19]	Back/Trap Door	IE8
[20]	Denial-of-Service	IIS, IE, >=Vista
[21]	Back/Trap Door	IE, XP SP3, Vista, 7

### 3.1 Vulnerability Notification to Attack

Only two articles provided the date of reporting the vulnerability and the date of the attack. This information is listed in Table 3.

**Table 3.** Duration from vulnerability notification to attack

Article cited	Vulnerability Notification Date	Attack Date	Duration
[4]	6/17/2010	7/15/2010	29 days
[11]	6/10/2010	6/16/2010	6 days

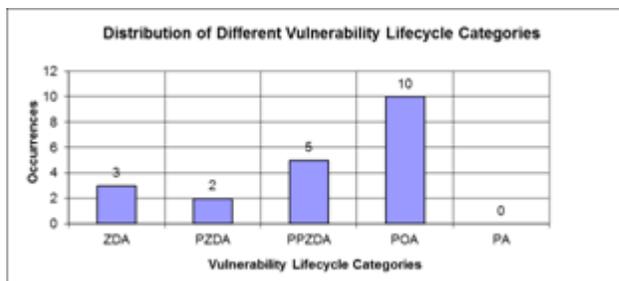
An exploit based on infected USB flash drives caused malicious “shortcut files” to appear [4].

An exploit of the Help and Support Center caused hackers to be able to perform “drive-by” attacks [11]. Drive-by attacks, are when a black-hat hacker is able to send the user prompts to trick them into going to malicious sites or directly downloading malicious files.

Comparing the above articles, we found that, a low visibility hack was used [4]. Low visibility meaning a feature not frequently used by users and thus vulnerability is not easily noticeable due to a lack of prior knowledge to prompt specific observation. This caused the duration between the vulnerability notification date and the attack date to increase as many security analysts tend to overlook those functionalities. However, the Help and Support Center incident perused by multiple users and is therefore of high visibility. Thus, more security analysts would be able to notice something strange and inconsistent. Therefore it seems that products with high visibility functionalities tend to have a smaller duration from the vulnerability notification date to the attack date.

### 3.2 Distribution of Vulnerability Life Cycle Categories

We classified the events from the nineteen articles into the five categories: ZDA, PZDA, PPZDA, POA, and PA. Only one article can be identified as both a POA and a ZDA [9]. Therefore it is included twice. Figure 1 shows the number of occurrences of different vulnerability life cycle categories.



**Figure 1.** Distribution of different vulnerability lifecycle categories

Figure 1 shows there are 20 total occurrences of zero day events. In each of the articles, vulnerability was proven with exploit code that was sent to Microsoft. Therefore, there were zero Passive attacks.

There are three ZDA occurrences [4], [9], and [11]. Infected USB drives created malicious shortcut links connected to the Stuxnet Worm, and when users open the links in a file manager, for instance Windows Explorer, their PC can be hijacked [4]. An exploitation of an invalid pointer reference in Internet Explorer could allow a black-hat hacker to perform remote code execution and thus use any command he/she wants [9]. An exploitation of Microsoft’s Help and Support Center allows black-hat hackers to perform drive-by attacks (or malicious websites that install malware once viewed) on users [11].

PZDA has two occurrences [3] and [15]. An exploitation of Microsoft’s Print Spooler by the Stuxnet Worm occurred even after a 2009 security magazine published the vulnerability [3]. Therefore, at least nine months passed before September 15, 2010, the vulnerability patch date. An exploitation code was produced to show that Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR) could both be bypassed when using Internet Explorer or Internet Information Services on Windows XP [15]. Microsoft delayed creating a patch due to being behind creating patches for earlier Zero-Day vulnerabilities at the time of the article.

PPZDA has five occurrences [5], [7], [14], [16], and [17]. An exploitation of improper error handling during encryption [5], and a minor bug is found in patched systems which could be exploited if black-hat hackers have physical access to the PC [7]. An exploitation of the way that Microsoft applications processes load libraries called remote binary planting [14]. However, it was pointed out that even though Microsoft could create a patch, they have not applied the patch because the patch could break all of the functionality in other applications. Malicious coding was performed in which Microsoft easily provided workarounds, but as of the time of the articles, they have not created a patch for either vulnerability mentioned in either article, [16] and [17].

Finally, there are ten POA occurrences, [6], [8], [9], [10], [12], [13], [18], [19], [20], [21]. In each of these articles the exploits are said to have happened a few days or several different days just before the time of the article. However, the vulnerability report date used is when Microsoft replies and thus acknowledges awareness of the vulnerability.

### 3.3 Most Common Microsoft Product Affected

Table 4 indicates the total number of exploits affecting a given Microsoft product and its percentage over the total number of exploits affecting all the listed Microsoft products.

**Table 4.** Microsoft products affected by exploit

Microsoft Products	Exploit	Exploit %
ASP.NET	1	1.19%
SharePoint 2007	1	1.19%
Internet Information Service	2	2.38%
Internet Explorer 6	4	4.76%
Internet Explorer 7	3	3.57%
Internet Explorer 8	4	4.76%
Windows 2000	6	7.14%
Windows XP	9	10.71%
Windows XP SP2	10	11.90%
Windows XP SP3	11	13.10%
Windows Server 2003	6	7.14%
Windows Vista	8	9.52%
Windows Server 2008	7	8.33%
Windows 7	8	9.52%
Windows Server 2008 R2	4	4.76%
<b>Total</b>	<b>84</b>	<b>100.00%</b>

Table 4 tallies each Microsoft product affected by each single exploit. For example, if a single malicious code infected three products, then each product would add one tally in Table 4. This is because one exploit produced on one product can easily be reproduced on another product and in each of the 19 articles these additional exploits were confirmed as successful. The list begins with ASP.Net, SharePoint 2007, Internet Information service, and three Internet Explorer versions. Each of these is a specific web application used by Microsoft customers that can affect the operating system they are using. The latter portion lists operating systems used by Microsoft customers which otherwise fall victim to an exploit. There are a total of 84 exploits combined across the various products from Microsoft with Windows XP SP3 as the most commonly affected product. However, the analysis shows that the newer versions of Windows are not as secure as one would assume.

### 3.4 Most Common Exploitation Technique

Table 5 indicates exploitation techniques and the number of times each exploitation technique is used.

**Table 5.** Exploitation techniques

Exploit	Times used	% used
Back/Trap Door	9	45.00%
Denial-of-Service	1	5.00%
Malicious Code	5	25.00%
Pharming	3	15.00%
Physical PC interaction	2	10.00%
<b>Total</b>	<b>20</b>	<b>100.00%</b>

The exploits used in Table 5 derive from the category of threats and software attacks described by Whitman and Mattord [22]. Back/Trap doors identify the attacks that bypass security features and expose technical bugs in the software. Denial-of-Service is commonly identified by a black-hat hacker sending a large number of connection or information requests in order to overload a system, and cause it to be unable to respond. Malicious code includes using a worm or malicious scripting in order to sabotage user input. Pharming identifies the attacks that redirect the user to a malicious site. Physical PC interaction denotes the attacks mentioned in the articles that could not directly attack a system through a network connection, but could instead attack offline and online after manual installation.

## 4 Related Work

Attack patterns are methods of abstracting possible vulnerabilities from characteristics of a specific attack [23]. Gegick and Williams performed an analysis using 244 vulnerabilities obtained from four different vulnerability databases and abstracted into 53 attack patterns [24]. They created an attack library which software developers could later compare against. They identified three most used attack patterns [25].

The first attack pattern is to submit an excessively long stream to a socket and cause buffer overflow. Buffer Overflows are commonly known as a method to access back/trap doors such as bypassing Data Execution Prevention and Address Space Layout Randomization [19].

The second attack pattern involves injecting malicious scripts/tags or variables in a web page, message board, email, etc., to obtain access to information such as cookies. The malicious code attack in one article shows that white-hat hackers discovered a way to Cross-Site Script users and gain access to their accounts on SharePoint 2007 which can be mapped to this attack pattern [13].

The third attack pattern deals with a malformed URL (e.g. excessive forward slashes, directory traversals, special chars such as '\*', Unicode chars) possibly causing a Denial-of-Service or in case of directory traversal, the user may obtain private information. This is similar to what occurred when a white-hat hacker discovered that an FTP server could have a Denial-of-Service attack after an attacker encodes specific Telnet characters [20].

## 5 Conclusion

The purpose of this paper was to identify some relationships among Microsoft's recent Zero-Day vulnerabilities. By analyzing 19 recent articles on Microsoft's Zero-Day vulnerabilities, we identified the most common vulnerability life cycle category, the most common Microsoft product exploited, and the most common exploitation technique used.

In the data we collected, Potential for attacks have the most number of occurrences. However, if the time from the vulnerability report date to the vulnerability patch date is long, they could easily become Pseudo Zero-Day attacks. Microsoft had severe delays in creating patches, even more than nine months [3]. As a result, Microsoft had to delay creating the following patches or disregard a patch if it is labeled as a minor vulnerability. Of the 84 exploits found in the articles, we found that Windows XP SP3 was the most commonly affected product. Surprisingly, Windows 7, which is thought of as being more secure and regularly updated, was affected nearly as much as Windows XP SP3. It is important to realize that those products with low numbers in Table 4, are not necessarily more secure, but instead are not as targeted with the techniques mentioned in Table 5. We also found from our data that the most prevalent vulnerability is based on Back/Trap doors. Our limited data also seems to imply that products with high visibility functionalities tend to have a smaller duration from vulnerability notification time to attack time.

In the future, collecting more articles on vulnerabilities of products from other companies and those used in the analysis performed in this paper will greatly increase the data set. Also, by strengthening and better structuring the concept started by this preliminary analysis, this same concept could be applied to other software products produced by other software developers both now and in the future.

## 6 References

- [1] "What is a Zero-Day exploit", accessed on April 10, 2011, available at: [http://what-is-what.com/what\\_is/zero\\_day\\_exploit.html](http://what-is-what.com/what_is/zero_day_exploit.html)
- [2] Amontip Jumratjaroenvanit, Yunyong Teng-amnuay, "Probability of Attack Based on System Vulnerability Life Cycle," *isecs*, pp.531-535, 2008 International Symposium on Electronic Commerce and Security, 2008.
- [3] PC World, accessed on March 13, 2011, available at: [http://www.pcworld.com/article/206010/microsoft\\_confirms\\_it\\_missed\\_stuxnet\\_print\\_spooler\\_zero\\_day.html](http://www.pcworld.com/article/206010/microsoft_confirms_it_missed_stuxnet_print_spooler_zero_day.html)
- [4] PC World, accessed on March 13, 2011, available at: [http://www.pcworld.com/article/201347/microsoft\\_warns\\_of\\_zero\\_day\\_windows\\_hole.html](http://www.pcworld.com/article/201347/microsoft_warns_of_zero_day_windows_hole.html)
- [5] PC World, accessed on March 13, 2011, available at: [http://www.pcworld.com/businesscenter/article/206463/microsoft\\_fixes\\_aspnet\\_zero\\_day\\_flaw.html](http://www.pcworld.com/businesscenter/article/206463/microsoft_fixes_aspnet_zero_day_flaw.html)
- [6] PC World, accessed on March 13, 2011, available at: [http://www.pcworld.com/businesscenter/article/205465/microsoft\\_reveals\\_stuxnet\\_worm\\_exploits\\_multiple\\_zero\\_days.html](http://www.pcworld.com/businesscenter/article/205465/microsoft_reveals_stuxnet_worm_exploits_multiple_zero_days.html)
- [7] PC World, accessed on March 13, 2011, available at: [http://www.pcworld.com/article/200511/angry\\_researchers\\_disclose\\_windows\\_zero\\_day\\_bug.html](http://www.pcworld.com/article/200511/angry_researchers_disclose_windows_zero_day_bug.html)
- [8] PC World, accessed on March 13, 2011, available at: [http://www.pcworld.com/article/198574/microsoft\\_confirms\\_critical\\_windows\\_xp\\_bug.html](http://www.pcworld.com/article/198574/microsoft_confirms_critical_windows_xp_bug.html)
- [9] PC World, accessed on March 13, 2011, available at: [http://www.pcworld.com/article/186957/microsoft\\_warns\\_of\\_ie\\_zero\\_day\\_used\\_in\\_google\\_attack.html](http://www.pcworld.com/article/186957/microsoft_warns_of_ie_zero_day_used_in_google_attack.html)
- [10] PC World, accessed on March 13, 2011, available at: [http://www.pcworld.com/businesscenter/article/198514/protect\\_windows\\_xp\\_from\\_zero\\_day\\_flaw\\_in\\_hcp\\_protocol.html](http://www.pcworld.com/businesscenter/article/198514/protect_windows_xp_from_zero_day_flaw_in_hcp_protocol.html)
- [11] PC World, accessed on March 13, 2011, available at: [http://www.pcworld.com/article/198965/hackers\\_exploit\\_windows\\_xp\\_zero\\_day\\_microsoft\\_confirms.html](http://www.pcworld.com/article/198965/hackers_exploit_windows_xp_zero_day_microsoft_confirms.html)
- [12] PC World, accessed on March 13, 2011, available at: [http://www.pcworld.com/businesscenter/article/201474/windows\\_shortcut\\_exploit\\_what\\_you\\_need\\_to\\_know.html](http://www.pcworld.com/businesscenter/article/201474/windows_shortcut_exploit_what_you_need_to_know.html)
- [13] PC World, accessed on March 13, 2011, available at: [http://www.pcworld.com/businesscenter/article/195261/microsoft\\_investigates\\_sharepoint\\_2007\\_zero\\_day.html](http://www.pcworld.com/businesscenter/article/195261/microsoft_investigates_sharepoint_2007_zero_day.html)
- [14] Computerworld, accessed on March 13, 2011, available at: [http://www.computerworld.com/s/article/9180978/Zero\\_day\\_Windows\\_bug\\_problem\\_worse\\_than\\_first\\_thought\\_says\\_expert](http://www.computerworld.com/s/article/9180978/Zero_day_Windows_bug_problem_worse_than_first_thought_says_expert)
- [15] Computerworld, accessed on March 13, 2011, available at: [http://www.computerworld.com/s/article/9178938/Three\\_more\\_Microsoft\\_zero\\_day\\_bugs\\_pop\\_up](http://www.computerworld.com/s/article/9178938/Three_more_Microsoft_zero_day_bugs_pop_up)
- [16] Computerworld, accessed on March 13, 2011, available at: [http://www.computerworld.com/s/article/9179358/Experts\\_predict\\_extensive\\_attacks\\_of\\_Windows\\_zero\\_day?taxonomyId=17&pageNumber=1](http://www.computerworld.com/s/article/9179358/Experts_predict_extensive_attacks_of_Windows_zero_day?taxonomyId=17&pageNumber=1)
- [17] C|NET, accessed on March 13, 2011, available at: [http://news.cnet.com/8301-27080\\_3-10461853-245.html?tag=mncol;9n](http://news.cnet.com/8301-27080_3-10461853-245.html?tag=mncol;9n)
- [18] The Register, accessed on March 13, 2011, available at: [http://www.theregister.co.uk/2010/08/30/apple\\_quicktime\\_critical\\_vuln/](http://www.theregister.co.uk/2010/08/30/apple_quicktime_critical_vuln/)
- [19] Computerworld, accessed on March 13, 2011, available at:

[http://www.computerworld.com/s/article/9174101/Hacker\\_busts\\_IE8\\_on\\_Windows\\_7\\_in\\_2\\_minutes](http://www.computerworld.com/s/article/9174101/Hacker_busts_IE8_on_Windows_7_in_2_minutes)

[20] The Tech Herald, accessed on January 3, 2011, available at:

<http://www.thetechherald.com/article.php/201051/6599/Micro-soft-now-dealing-with-a-second-Zero-Day-vulnerability>

[21] ZDNet, accessed on March 13, 2011, available at:

[http://www.zdnet.com/blog/security/attack-code-posted-for-new-ie-Zero-Day-vulnerability/7859?tag=mantle\\_skin;content](http://www.zdnet.com/blog/security/attack-code-posted-for-new-ie-Zero-Day-vulnerability/7859?tag=mantle_skin;content)

[22] Michael Whitman and Herbert Mattord, *Principles of Information Security*. Course Technology, Cengage Learning, 2009.

[23] Greg Hoglund and Gary McGraw, *Exploiting Software*. Boston: Addison-Wesley, 2004.

[24] Michael Gegick and Laurie Williams, "On the design of more secure software-intensive systems by use of attack patterns," *Information and Software Technology* 49 (2007) pp. 381–397.

[25] Michael Gegick and Laurie Williams, "Matching attack patterns to security vulnerabilities in software-intensive system designs," *ICSE-SESS*, ACM Press, 2005.

# Smart Grid Insecurity: A New Generation of Threats

Summer Olmstead and Dr. Ambareen Siraj

Department of Computer Science

Tennessee Tech University

Cookeville, Tennessee, United States

**Abstract** - *The critical infrastructure powering the nation is currently undergoing a massive collaborative effort to integrate modern technologies with 50 year-old assets derived from 100 year-old designs. Improving the existing electric power infrastructure with smart grid technologies compounds existing threats with new threats. Identification of next generation smart grid security threats is vital for implementing a more secure national power grid. This paper discusses security threats in the information layer of the smart grid conceptual model by mapping threats to the seven domains. This paper also discusses federal- and state-level smart grid security initiatives.*

**Keywords:** cyber security, smart grid, smart grid security

## 1 Introduction

The critical infrastructure powering the nation is currently undergoing a massive, collaborative effort to integrate modern technologies with 50 year-old assets derived from 100 year-old designs. The North American electric power grid is evolving into an interactive, dynamic grid with substantial efficiency, reliability, and security improvements.

Improving the existing electric power infrastructure with smart grid technologies compounds existing threats with new threats. The identification of next generation smart grid security threats is vital for implementing a more secure national power grid as required by federal policy and guidelines. While the flow of electricity was previously a one-way communication from generation to consumer distribution, in response to government smart grid initiatives, the National Institute of Standards and Technology (NIST) redefines the flow of electricity with two-way communications shown in the smart grid conceptual framework and its seven domains (Figure 2) [1]. This paper discusses security threats in the information technology layer of the smart grid conceptual model by mapping threats to its seven domains - bulk generation, customer, distribution, markets, operations, service provider, and transmission. This paper also discusses newer smart grid security initiatives at the federal- and state-level.

Increasing demand on the antiquated grid will eventually exceed the capabilities of the grid. Continuing technology progressions propagate the smart grid development

supported by the 2007 Energy Independence and Security Act and \$4.5 from the American Recovery and Reinvestment Act [7, 9]. National energy demands continue to increase requiring efficient energy production control. Electricity is currently produced in response to demands largely due to the lack of battery technology capable of storing electricity. Alternative sources of power include wind and solar generators.

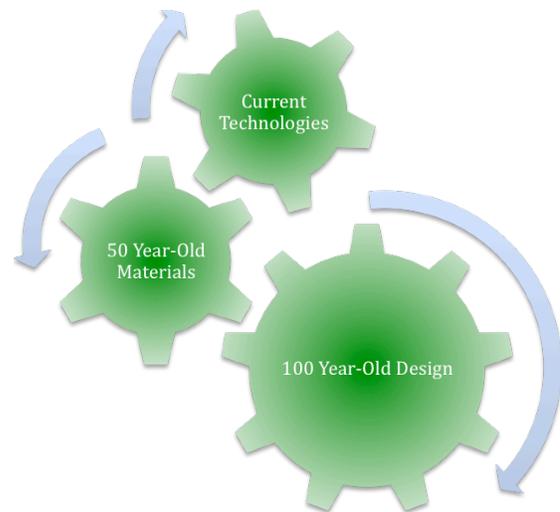


Figure 1. The smarter grid.

## 2 Defining the Grid

An electrical power grid is an electricity network supporting generation, transmission, and distribution operations. The smart grid is the modernization of the national electric power grid. The existence of grid vulnerabilities has been recognized for years, the next generation of smarter grids is being developed with security in mind. The prominent deployment of smart grid technologies brings focus to vulnerability exploitation issues.

The smart grid is a combination of legacy and neoteric systems. Threats are inherited from the legacy systems and introduced with the adoption of new systems. While the smart grid will offer redundancy, vulnerabilities can exist at a single point of failure. Due to the nature of power service disruption threats can come from a variety of sources.

Natural threats rank highest in significant power disturbances with approximately 60% of disturbances contributed to weather [4]. Other general threats universally applicable to the domains include cyber failure, equipment failure, and human error. With increasing computing technology being introduced into the power grid, we expect to see an increase in the number of cyber failure related outages.

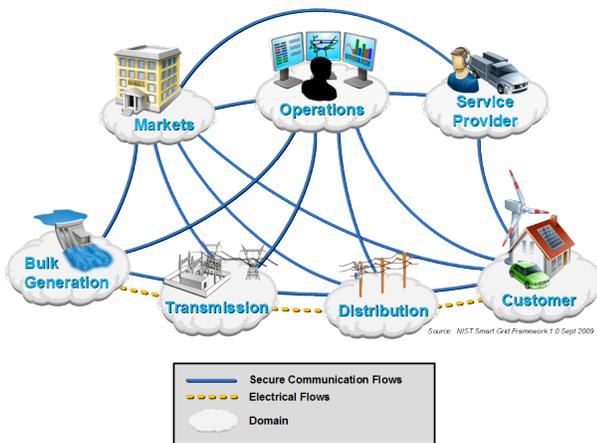


Figure 2. NIST Smart Grid Conceptual Model [1].

### 2.1 Bulk Generation Domain

Large quantities of electricity are generated from energy sources within the bulk generation domain. The classification of energy sources include renewable-variable, renewable non-variable, non-renewable non-variable, and energy storage [1]. Traditional threats to bulk generation include physical threats to power plants. Smart grid technologies that increase networking into the Internet introduce a new generation of threats. One of the emerging renewable energy technologies is photovoltaic (PV) energy. PV energy will continue to be more widely incorporated into the national electric power infrastructure in the bulk generation domain. The cyber security threats, which come from PV energy generation, could impact the software, hardware, and communication flows. In order for PV energy generation to remain profitable and useful in the grid, it must remain available. A denial-of-service (DoS) attack on the communication flows to and from a PV network could prevent the generation of PV.

If the control unit for PV generation is accessible through the Internet, or the operator of the control unit is capable of accessing the Internet through the control unit, then there exists the potential exposure to malicious code. A virus, worm, or other malicious code could be a gateway for a method to attack the PV control unit. For example, a DoS attack gaining control of the unit, or stealing information to

gain permissions high enough to alter the performance and/or communication within the PV generation station. While these concepts are not all encompassing, it is natural to see the absence of potential cyber security threats if the PV generation station was lacking the technology to communicate on a network digitally. Without the technology the PV generation station would simply be generating electricity and pushing it onto the transmission grid.

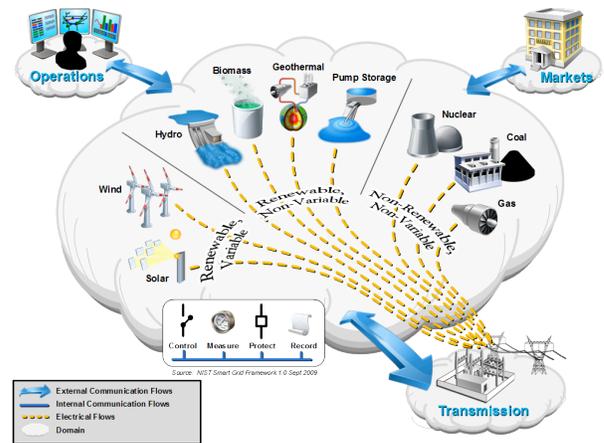


Figure 3. Bulk Generation Domain [1].

### 2.2 Customer Domain

Consumers of electricity in the customer domain include home, commercial, and industrial. The customer domain uses the Advanced Metering Infrastructure (AMI) to connect with the distribution system in two-way communications. The AMI is rich in smart grid technologies such as smart meters [1]. Smart meters provide a higher level of control than the previous generation of meters. The customer domain additionally encompasses plug-in-vehicles (PEV). Utilizing the two-way communications available in the smarter grid, customers may manage, generate, and store electricity. One-way communication usually meant customers managed energy consumption with little information and almost no interaction with the grid. The average home consumer use case for energy management would consist of energy usage verified by the monthly utility bill. Efforts to manage electricity would include unplugging unused appliances, turning off lights, and similar common energy saving techniques. Real-time information was not readily available to accurately control and manage consumption. Smart meters offer the two-way communication required for fine-grained energy consumption. Now, customers are able to utilize pseudo-real time rates, smart appliances, and applications to develop a fine-grain energy-use policy. Customers can generate electricity to sell to the utility company for redistribution.

Legacy threats include service theft and fraud. New threats include the next generation of service theft and fraud. For example, a home customer simulates generating electricity to sell to the local utility company to fraudulently make money. Another example of a new threat would be a case where a home user hacks their smart meter to alter their actual energy consumption to lower their utility bill. A traditional threat to individual customers would be the disruption of service due to downing the only power supply line to the customer. A new threat to the individual customer would be the disruption of service due to a denial-of-service attack targeting the smart meter located at their residence. This new threat offers attackers an anonymous and lost cost method of disrupting power service to utility company customers. The smart grid may be resilient for power outages of a larger scale, but a home customer with one line of power into their home represents a single-point of failure in the grid.

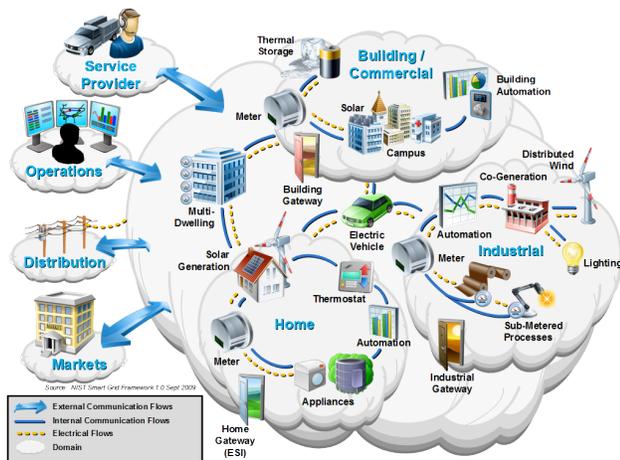


Figure 4. Customer Domain [1].

### 2.3 Distribution Domain

The distribution domain of the smart grid will use the distribution system to supply electricity to customers and to consume electricity from customers. Distribution substations of the distribution system receive electricity from the transmission system. Traditional threats include physical threats to the distribution substations. A new threat can be seen in any number of technologies within the domain, but here we focus on the communication flow between the customer and distribution domains.

Attacking communication flow between any domains could result in severely impacting the availability of power. For example, within the customer domain there exists two companies, A and B respectively who are competitors. With Currently the nature of the power grid being supply with demand, there is enormous motivation for one of the companies to be able to attack a competitor company or its

communication in an effort to dominant supplying power during a peak demand time. Company A could utilize the new technologies in the communication flow between the customer and the distribution company to deny the ability for Company B to distribute power to customers. This enables Company A to supply the customers with power while Company B is out of operation. Company A can utilize a wide array of attacks methods in this regard. While this type of attack is not a new concept in the business world, the cyber technologies being integrated into the grid brings a whole new level of digital competition to the distribution domain.

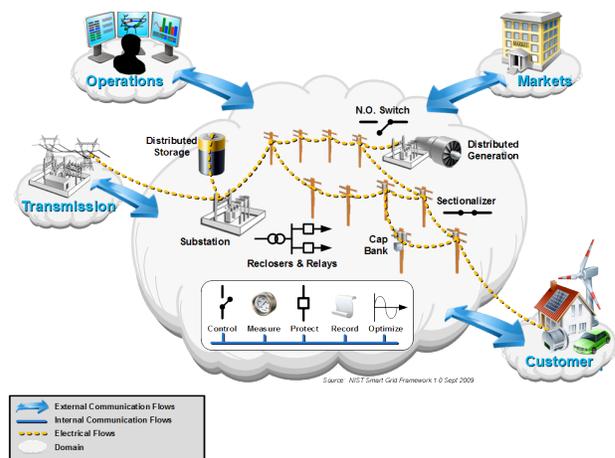


Figure 5. Distribution Domain [1].

### 2.4 Operations Domain

The operations domain plays an essential and unique role by managing the flow of electricity of all domains. The smart grid will enable the movement of responsibility of power system operations from a regulated utility to outsourced service providers [1]. The operation domain includes the following typical applications [1]: stakeholder planning and management, business planning/reporting, human resources, security management, premises, communications network, financial, supply chain/logistics, meter reading/control, customer support, extension planning, maintenance/construction, operational planning, records/assets, and network operations. A sub-domain of the operation domain is network operations. The network operations sub-domain includes applications for analysis, calculations, control, fault management, monitoring, reporting/statistics, and training.

With financial motivation in mind, there are emerging threats in the operations domain. The integration of computing technologies into the financial control increases opportunities for accessing information by an unauthorized user. If a financial application within the operations domain exists on a network, the application is at risk for being

accessed by an authorized user. Methods for attacking the application range from brute force attacks, unauthorized access through social engineering, and exposure to malicious code. An attacker could access the information on the financial application to use in a number of ways. One example could be to sell to competing companies for setting their units of price of electricity. If Company A wants to sell their electricity at a slightly lower rate than Company B to increase their volume of customers, then data from the financial application in the operations domain would be potentially extremely valuable to provide business insight. The nature of cyber attacks makes them inexpensive or free for an attacker to execute from nearly anywhere in the world, with almost complete anonymity. Prior to having access to the financial application over an exposed external network, the attacker would need to physically travel to the operation location and gain physical access to the financial application in order to steal the information.

smarter grids, the customer typically communicated demand by simply using electricity. The added layer of smart grid communication technology will enable retailers to locate and charge electricity rates sold to customers based upon IP addresses located within the customer smart meters. An attacker could use an IP address spoofing network attack to change the IP address of the smart meter being billed. The retailer would incorrectly bill the customer using the spoofed IP address.

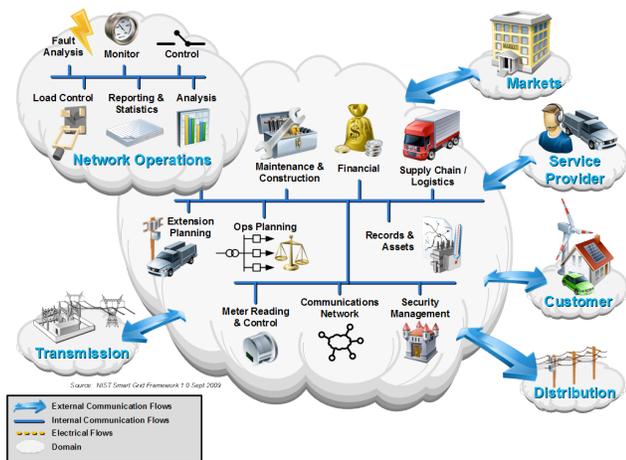


Figure 6. Operations Domain [1].

## 2.5 Markets Domain

Managing the actors in the markets is the responsibility of the markets domain. The markets domain includes the following typical applications [1]: ancillary operations, distributed energy resources (DER) aggregation, market management, retailing, trading, and market operations. Critical to the smart grid balancing the demand of energy from consumers with the response from suppliers is the communications between the bulk generation, transmission, distribution, customer domains and the markets domain [1]. Existing threats to this domain include physical threats and newly introduced threats to the applications within the markets domain could be seen in the external communication flow between retailing within the markets domain and the customer domain. Consider the case where a network attack affects the availability of communications between the markets and consumer domains. Prior to two-way communication available in

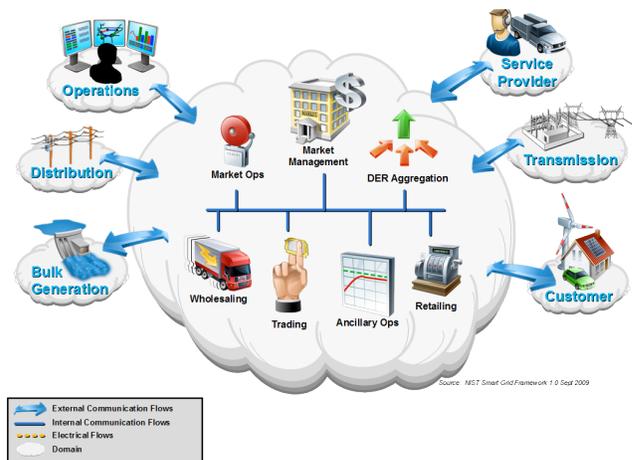


Figure 7. Market Domain [1].

## 2.6 Service Provider

Each domain can interact with third parties. The role of the service provider domain is to manage the operations between third parties and domains. Service providers interact with the operations, markets, and customer domains [1]. One of the areas to receive a lot of attention with regards to cyber security usually rests within the interaction or communications between the service provider and the customer domain. The driving force in the electricity power industry business is the customer – the source of demand and profit for the industry.

An example of a service provider and customer domain interaction can be seen in home management. With the integration and adoption of computing technologies into the power grid and consumer appliances and homes – there will be a need for homes to become “smart.” A customer wanting to take full advantage of the new smart appliances would want a “wired” home – a home capable of communicating with smart meters and their utility company. A service provider could provide the physical computing technologies for a house as well as the software components and applications to make home management user friendly for the consumers. The more networking capabilities a home possesses, particularly wireless networking, the threats from cyber attacks increase. Threats to wireless networks range from DoS attacks to traffic

sniffing. Customers become at risk for having their personal and confidential information stolen or accessed by an unauthorized user. For example, an angry neighbor could hack into your home management software application to shut off your power – or even steal your financial information. Customer education for defending and mitigation of possible cyber security threats is essential and should be part of the responsibility (as part of combined effort from federal, state, and local authorities) of the service provider to the customer.

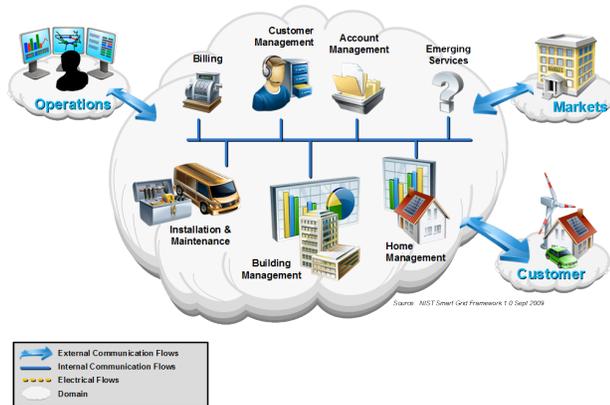


Figure 8. Service Provider Domain [1].

## 2.7 Transmission Domain

The transmission domain plays the vital role of interacting with the bulk generation, operations, markets, and distribution domains to transmit bulk-generated power to the distribution domain and eventually to customers. External flows of communication are present between each of the interacting domains and electrical flows exist from bulk generation through the transmission domain and to distribution. Each external flow of communication represents a possible point of security vulnerability.

A physical attack to the assets transmitting bulk generation power onto the transmission system can be an example of legacy security vulnerability. A newly introduced security vulnerability could be an attack on the control systems. High voltage electric transmission requires highly sophisticated control systems capable of monitoring and balancing supply and demand. If an unbalance of supply from the bulk generation domain and the demand, then both the bulk generation and transmission domain could malfunction and result in a blackout. If increased network connectivity is introduced to the vital transmission control systems in the smart grid initiative, then an attack on the system could disrupt monitoring functions and shut down the section of the transmission grid under control of the control system.

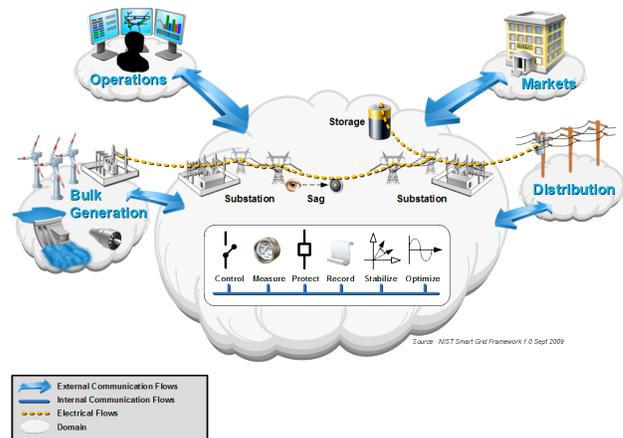


Figure 9. Transmission Domain [1].

## 3 Smart Grid Security Initiatives

### 3.1 Federal

The world is moving towards smart electricity, and the future of energy is smarter electric power grids. Networks are being created between power plants, utility companies, and consumers increasing network connectivity while producing challenges for securing the smart grid. Initiatives for smart grid security can be seen at the federal- and state-levels in the United States.

The national power grid is a critical infrastructure requiring federal security efforts. The government is developing and deploying smart grid security guidelines. The major contribution from the federal government in regards to cyber security is the NIST publication “Guidelines for Smart Grid Cyber Security”[5]. This publication offers a guideline to utilities to customize into standards fitting their operations. Also, the DOE Federal Smart Grid Task Force has been created and tasked with some security aspects concerning the smart grid [8]. Federal-level initiatives can be seen from the Department of Homeland Security (DHS), Department of Energy (DOE), Federal Energy Regulatory Commission (FERC), the National Institute of Standards and Technology (NIST), and the United States Federal Government.

### 3.2 State

The North American electric power grid covers the nation in an elaborate network of assets. While the grid crosses state borders, individual states recognize the importance of the smart grid cyber security. Addressing smart grid development and deployment through the regulation of utility companies within jurisdiction is evident in a variety of initiatives from California [9], Illinois [10], and Maryland [11]. The California State Senate passed Senate

Bill 17 which addresses smart grid law for the state. The Illinois General Assembly passed Senate Bill 1592 and the Maryland General Assembly passed the EmPOWER Maryland Energy Efficiency Act of 2008.

### 3.3 Community

While security requires a level of secrecy, dissemination of information is vital to securing the smart grid. The smart grid community is providing an array of resources from individual, university research, conferences, and groups. Table I lists several smart grid community resources.

TABLE I. SMART GRID COMMUNITY RESOURCES

Resource	URL
Smart Grid Security Blog	<a href="http://smartgridsecurity.blogspot.com">http://smartgridsecurity.blogspot.com</a>
Smart Grid Security Resources	<a href="http://smartgridsecurityresources.csc.tntech.edu">http://smartgridsecurityresources.csc.tntech.edu</a>
Smart Grid Cyber Security Summit	<a href="http://www.smartgridsecuritysummit.com">www.smartgridsecuritysummit.com</a>
NIST Smart Grid Cyber Security Working Group	<a href="http://collaborate.nist.gov/wiki-sggrid/bin/view/SmartGrid/CyberSecurityCTG">http://collaborate.nist.gov/wiki-sggrid/bin/view/SmartGrid/CyberSecurityCTG</a>

## 4 Conclusion

With government financial initiatives and the continuing need for a smarter national electric power grid, the transformation of the electric grid infrastructure will continue indefinitely. The goals of the smart grid rely on overall increased efficiency, reliability, and security. While securing the smart grid is a seemingly overwhelming task, persistent security efforts are imperative for ensuring smart grid benefits are greater than the risks. The loss of confidentiality, integrity, and availability by an attacker is being redefined within the electric power grid. Technologies from hardware enabling local networking to wireless communications are being introduced into our national power grid along with the inherent threats they impose (DoS, man-in-the-middle, spoofing). Through the identification of threats we can move into cyber security defense strategies encompassing prevention, detection, response, and recovery.

Using the NIST Smart Grid Conceptual Framework domains, we have identified security threats through domain mapping. Additionally, we have discussed smart grid security initiatives at the international, federal, and state level. Through the identification of threats from legacy systems and introduced with smart grid technologies, the mitigation process can proceed and the vision of Grid 2030 can become a reality.

## 5 Acknowledgment

Work funded by the Office of Research at Tennessee Tech University.

## 6 References

- [1] National Institute of Standards and Technology, NIST Special Publication 1108: NIST Framework and Roadmap for Smart Grid Interoperability Standards, [www.nist.gov/public\\_affairs/releases/upload/smartgrid\\_interoperability\\_final.pdf](http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf), 2010.
- [2] United States Department of Energy, Technology Providers, [www.oe.energy.gov/DocumentsandMedia/TechnologyProviders.pdf](http://www.oe.energy.gov/DocumentsandMedia/TechnologyProviders.pdf), 2009.
- [3] North American Electric Reliability Corporation, "2009 System Disturbance Report", <http://www.nerc.com/files/disturb09.pdf>, 2009.
- [4] National Institute of Standards and Technology, NIST IR 7628. "Guidelines for Smart Grid Cyber Security," <http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7628>, 2010.
- [5] United States Department of Energy Office of Electricity Delivery and Energy Reliability, "Study of Security Attributes of Smart Grid Systems – Current Cyber Security Issues," [http://www.inl.gov/scada/publications/d/securing\\_the\\_smart\\_grid\\_current\\_issues.pdf](http://www.inl.gov/scada/publications/d/securing_the_smart_grid_current_issues.pdf), 2009.
- [6] United States Energy Information Administration, Energy Independence and Security Act of 2007, [http://energy.senate.gov/public/\\_files/getdoc1.pdf](http://energy.senate.gov/public/_files/getdoc1.pdf), 2007.
- [7] United States Department of Energy, Federal Smart Grid Task Force, [http://www.oe.energy.gov/smartgrid\\_taskforce.htm](http://www.oe.energy.gov/smartgrid_taskforce.htm), 2010.
- [8] United States Government, American Recovery and Reinvestment Act of 2009, [http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=111\\_cong\\_bills&docid=f:h1enr.pdf](http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=111_cong_bills&docid=f:h1enr.pdf), 2009.
- [9] California State Senate, Senate Bill 17, [http://info.sen.ca.gov/pub/09-10/bill/sen/sb\\_0001-0050/sb\\_17\\_bill\\_20091011\\_chaptered.pdf](http://info.sen.ca.gov/pub/09-10/bill/sen/sb_0001-0050/sb_17_bill_20091011_chaptered.pdf), 2010.
- [10] Illinois General Assembly, Senate Bill 1592, <http://www.ilga.gov/legislation/95/SB/PDF/09500SB1592enr.pdf>, 2010.
- [11] Maryland General Assembly, EmPOWER Maryland Energy Efficiency Act of 2008, <http://mlis.state.md.us/2008rs/bills/sb/sb0205t.pdf>, 2008.

# NEW DETECTION TECHNIQUE USING CORRELATION OF NETWORK FLOWS FOR NIDS

P.V.Amoli<sup>1</sup>, A.R.Ghobadi<sup>2</sup>, G.Taherzadeh<sup>2</sup>, R.Karimi<sup>3</sup>, S.Maham<sup>3</sup>

<sup>1</sup> Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, Johor Bahru, Malaysia

<sup>2</sup> Faculty of Information Technology Multimedia University, Selangor, Malaysia

<sup>3</sup> SOHA Sdn. Bhd. Cyberjaya, Selangor, Malaysia

**Abstract**— *Network Intrusion Detection System (NIDS) is a security mechanism to monitor the behavior of the network; in case of any abnormal behavior inside of the network it should alert the Administrator about incoming intrusion. In the world of computer Prevention, Detection, Reaction is 3 layers of actions to an attack. By applying these layers we can increase Confidentiality, Integrity and Availability (CIA) of systems and data.*

*In this paper, The detection technique which was proposed and implemented is about monitoring incoming Network Flows and Packets to detect some of network intrusion such as DOS (Denial of Service) and DDOS (Distributed Denial of Service) attack like (I)Process Table, (I)SSH Process Table, (III)SYN Flood Neptune, (iv)UDP Storm Attack and (v)Smurf.*

**Keywords:** Network Security, Intrusion Detection System, Network Flows, Correlation Algorithm.

## 1. Introduction

Today, regarding the remarkable growth of internet, one of the most important concern of the network based services providers is security. The main goal of security is to avoid any kind of computer attacks. There are three stages for securing a network: (I) Prevention, (I) Detection, and (III) Reaction [1]. (I) The goal of prevention stage is to prevent the system from any kinds of attacks or unsecured states; however, it cannot ensure that the attack will not cross this mechanism, so if prevention mechanism could not stop the attack, the detection level will take the actions. (I) The goal of detection mechanism is to detect the attack and send an alarm to administrator for stopping the attack[2]. Intrusion Detection System (IDS) is one of detection mechanism.

The aim of IDS is to detect activity which is against the policy and it is a necessary component of protection beside of other security mechanism such as access control. In case of any attack IDS raise alerts which could be in real time [3]. IDS collect information from sources to detect the intrusion, there are no perfect IDS because there is no perfect algorithm for detection, and also it is hard to choose the best source for detecting the intrusion.

In this paper we propose and implement a detection technique for NIDS which use correlation algorithm on network flow.

The rest of this paper is organized as follows: Section 2 describe more on previous work, in Section 3 the methodology will be discussed, Section 4 is about experimental result and finally last Section is about conclusion and future work.

## 2. Related Works

Numbers of IDS have represented by some researchers, for example Network Based IDS (NIDS), Host Based IDS (HIDS), Application based IDS and etc. The NIIDS system connects to the “Access Points” (like hubs and switches) to collect and check all of the packets which transferred in the network. The NIDS mechanism focus on the header of the packets, and the detection is based on the data which comes from the headers. Headers contained data like: protocol, packet size, source, destination and etc.

In The NIDS, the two most common techniques of detection are Signature Based and Anomaly based. In Signature based NIDS the system looks for characteristics of known network attacks. This technique can precisely detect illegal accesses which are contained in the signature database which made before. In Anomaly based NIDS the system will adopt the normal state of the network traffic as criteria of anomaly, by this way it can detect novel and unknown network attacks without database of known attacks. However, it makes a lot of detection errors because of the difficulty to define the normal state of the network traffic precisely. [4]

Each NIDS has a unique way to monitor the network traffic such as bytes monitoring, packets monitoring and Network Flows monitoring [5], also some of the NIDS engine check and monitor the log files to suspicious action in the network [6]. Researches' experiments show that the best source to detect most of the network attacks such as DOS and DDOS are through checking network flows [7]. The chance of detection will be more in Flow-based NIDS [5]. To rank or major the performance of any NIDS, complex attacks detection rate should be considered. Detecting complex attack by NIDS could be done by combining several facts [3].

There are several methods to detect complex attacks, for example one of them is Correlation method. Correlation defined as a number or percentage of similarity between two random variables to find and show the relationship. Previously many researchers used correlation algorithm on security mechanism such as IDS to find and detect suspicious activities.

We review related work in the areas of (I) byte, packet and network flows based NIDS, (I) evaluation of previous NIDS on different protocol and finally (III) evolution of correlation mechanism on NIDS.

As mentioned before, NIDS get input from the transmitted data inside of the network. The data could be in the form of Byte, Packet, and Network Flow. The research which done by Lakhina et.al. [8] shows that, if NIDS check Bytes it only can detect 50% of the intrusions (compare to network flow based) and also when the NIDS check Packets it detects 75% of the intrusions (compare to network flow based). In this research it shows the best performance comes from checking Network Flow by NIDS. Two main type of errors in IDS is (I) False Positive (The event which push IDS to make an alarm when no attack has taken place) and (I) False negative (IDS fail to detect an actual attack).

As mentioned before [7][3], today network attacks became more complex, so the NIDS which only check the Bytes could not detect the complex attack, because the data which can be extracted from Bytes is not enough, Also in Packet based NIDS the system get more information about the network but sometimes it is not enough. One of the advantages of using network flows in NIDS is getting full information about the reaction of network and also finding the suspicious behavior inside of the network. Each network flow consist of [9] [10] [11] set of packets from same source to destination and inverse (except UDP protocol which defined as one way communication). Network Flow has summery of one communication which consist of (I) Duration of network flow (I) Protocol (III) Byte and etc...[11][12]. Another advantage of Network Flow is saving all of traffic in one place instead of packet based system which needs high resource to save all of the data about each packet. [8]

Previously several researchers used correlation algorithm for NIDS [1][11][13][12][14]. In 2003 they used log correlation for IDS [1], in this method they used network flow and their performance became high because of using a sliding window but the limitation of their project was about focusing only on the TCP protocol and also, because of lack of information and testing, the optimal window size was not determined very well so the probability of having false alarms were high.

Another research was about Multi-Dimensional Flow Correlation [11], this method used to fine the similarity in flows by correlation algorithm to tracks multiple characteristics, in that case the IDS will not rely on one characteristic only and it can detect complex attack but the Limitation of this method was waiver of ICMP protocol and focusing more on TCP and UDP protocol.

In 2004[13], one of the researches was on detection of DDoS attack based on Using Source IP Address Monitoring. In this system the focus was more on correlating network flow from new IP addresses, the

performance of system was good enough but the problem may occur if the attack come and start from inside, in that case all of the IP addresses will be trusted and known and the attack will not be detected, so by this scenario the system can generate high false negative alarm.

In 2006 [14] one of the proposed method was Alert Correlation Analysis in Intrusion Detection. Each IDS generate several alarm, but some of them are false positive and also some of the alarm those not mean anything(because they are single fact), so by correlation several alarm it is possible to increase rate of detection (because it can also detect complex attacks by having several facts) and also having less false positive. In this method the system will correlate alarms and then check it with the policy to find the suspicious behavior. This method had a high performance but the only problem was checking packet instead of network flow make more false negative alarm.

We believe the results from previous research shows that, the best source to find intrusion inside of the network is network flow, because it is the summery of any reaction inside of network, so it will be much faster to analyze it (compare to check each packet or byte one by one). Data could be categorized in several layers, (i) raw data, (ii) information, (iii) knowledge and etc., Byte and Packet is the raw data about reaction of one network, but when it comes to Network Flow it will be information. So another advantage of using network flow in NIDS is having better result of detection because the data in more visible and understandable compare to packet and Byte (raw network data). Another outcome of previous research shows that today most of intrusions became complex, correlation algorithm help us to simplified those complexity.

### 3. Methodology

Our main goal is to improve rate of detection (intrusions) in this NIDS and also make a fast and light detection technique to speed up the NIDS. In this part we use several approaches in this NIDS to achieve our goal. Each system has 3 stage, (i) input, (ii) process, (iii) output. In this part we will discuss the stages in detail.

#### 3.1 Input (Network Flow)

As mentioned before NIDS is connected to access points and collect all of the packets which are coming through access point. The proposed solution needs network flow so first it should convert the packets to one flow. In this system we focused on TCP and UDP and SMTP protocols. Network Flows in TCP: it consists of packets from one source to one destination during one communication. In TCP network flow the source port and destination port will not change. It starts with SYN packet and it will finish with FIN packet. Figure 1 shows the detail of Network flow in TCP.

Figure 1: Network Flow in TCP

Source	Destination	Source Port	Destination Port	Protocol	Info
A	B	X	Y	TCP	SYN
B	A	Y	X	TCP	SYN/ACK
A	B	X	Y	TCP	ACK
•					
•					
•					
•					
A or B	B or A	X or Y	Y or X	TCP	FIN

Network Flow in UDP: it is exactly like TCP network flow but the communication is from one source to one destination, it means it is one way communication. Figure 2 shows the detail of Network flow in UDP.

Figure 2: Network Flow in UDP

Source	Destination	Source Port	Destination Port	Protocol
A	B	X	Y	UDP
A	B	X	Y	UDP
•				
•				
•				
A	B	X	Y	UDP
A	B	X	Y	UDP

Network Flow in SMTP: in this protocol the sender will send one request to destination then destination machine will answer to the request. Figure 3 shows the detail of Network flow in SMTP.

Figure 3: Network Flow in SMTP.

Source	Destination	Info
A	B	Request
B	A	Reply

During packet collection for each network flow the system will store some information about each network flow, such as Time of First Packet in the Flow (sec), Source IP, Destination IP, Protocol, Size of Flow (byte), source port, Destination Port, Status of Flow, Packet Counter, Time of Last packet in the Flow(sec), and Duration of Flow(sec).

If the network flow didn't finish properly, and no packets (from source and destination of that specific network flow) pass through access points for 60 seconds the system will automatically close the network flow to finalize the status.

### 3.2 Process (Correlation)

In this step correlation engine will find the suspicious relation between network flows according to the policy or signature which defined previously. The policy which defined by us is according to the signature of attacks in DARPA website [15], in this project we tested our system with several DOS/DDOS attacks, such as: (i)Process

Table, (ii)SSH Process Table, (iii)SYN Flood Neptune, (iv)UDP Storm Attack and (v)Smurf.

Correlation Policies of these five DOS/DDOS attacks are mentioned in below table.

Table 1: Correlation Policies for five DOS/DDOS attacks.

DOS/DDOS Attacks	Signature
<b>Process Table</b>	If the system find five unsuccessful flow in one window from one sender to one destination in TCP protocol it will save it as Process Table attack.
<b>SSH Process Table</b>	If the system finds five unsuccessful flow in one window from one sender to one destination in TCP protocol which lead to SSH port it will save it as SSH Process Table attack.
<b>SYN Flood Neptune</b>	If the number of SYN flow from one sender to one destination become more that 5 in one windows the system will save it as a SYN Flood Neptune (the flow start with SYN packet but it will not continue)
<b>UDP Storm Attack</b>	If one flow start from chargen port(UDP) to echo port(UDP), then system will save it as UDP Storm attack because large number of flow will start from receiver.
<b>Smurf</b>	If one machine do not send any 'echo request' but it receive high number of flow with 'echo replies' (ICMP) from several sender in one window the system will save it as Smurf attack.

Correlation engine will collect and check network flows from t0 to t1 (window size). The window size which proposed in this NIDS is 10 seconds so it means if t0 is 0 then t1 will be 10. According to our study on signature of network based attacks in DARPA traffic sample [15], 10 seconds of network flows has enough information to detect all of the network attacks.

Another point which makes this solution unique is smooth movement of window in NIDS. The window will move smoothly, it means the second window will not start from 10, it will starts from 7, it means this time the t0 will be 7 and t1 will be 17. By this technique we minimize the probability of losing suspicious relations of network flows which happened short period of time but in two different windows.

Figure 4 shows the correlation process clearly, the correlation algorithm will check all of the flows inside of one window according to the policy which we determined and explained before, if it find any attack inside of windows it will send the error to administrator to make the reaction otherwise the system will continue its job.

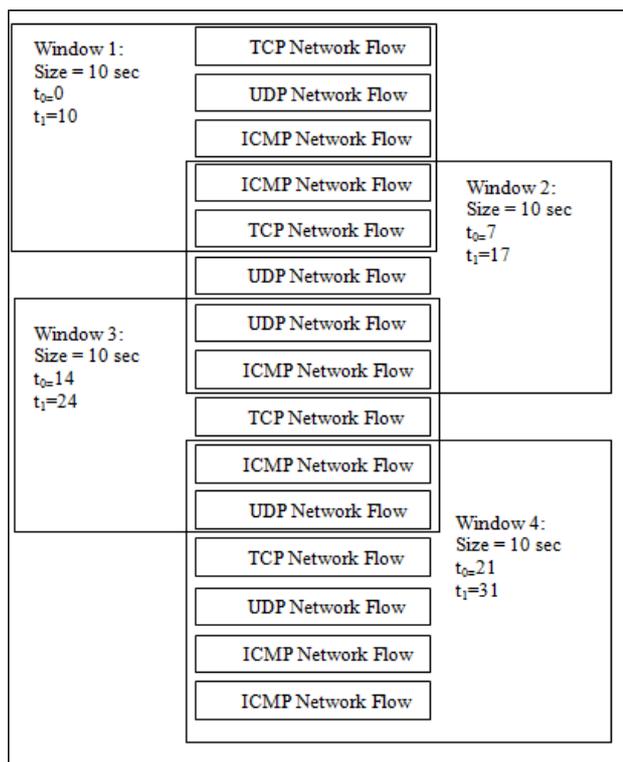


Figure 4: Smooth movement of window

### 3.3 Output (Attack Detection Alarm)

If any attack happen and the NIDS detects it, it will generate the alarm to administrator with full information, for example: (i) Number of attack, (ii) Type of attack, (iii) Attacker IP address, (iv) Victim IP address, (v) How many times the attack happened, (vi) Duration of attack, and (vii) Ports number.

## 4. Experimental Result

Previous solutions on NIDS made several errors because the definition of critical situation was not defined well (weak policy), also most of NIDS do not collect all useful information. But in this proposed solution we tried to collect all of useful information from packets and network flow and also we tried to define attack policy in the best way.

Another study was on obtaining the window size of NIDS in the best way. Based on the testing and result analyzing if the windows size is too small for example two seconds some of the attack may not be detected by NIDS because during attack some of the windows become attack free so the NIDS will not generate any alarm, and if the size of window become so large it will be much harder to detect attack because attack already mixed with normal traffic so it will be hidden to NIDS. So based on the signature of attack the best proposed size for NIDS to detect those four types of attack was having 10 sec window and also moving it forward smoothly.

We implemented our NIDS with Java in Microsoft Windows environment and we tested this system with DARPA traffic sample [15].

## 4.1 Sample of Results

Figure 5 shows the number of TCP flow which opened with SYN packet and never continued, this figure shows that around 12 network flows open in two seconds. As mentioned before the signature of Neptune (SYN flood-DOS attack) attack is having big number of TCP flow which opened by single user and never continued. So the correlation engine will check the window and finding the suspicious relation between network flows. According to the attack policy, this attack is SYN Flood.

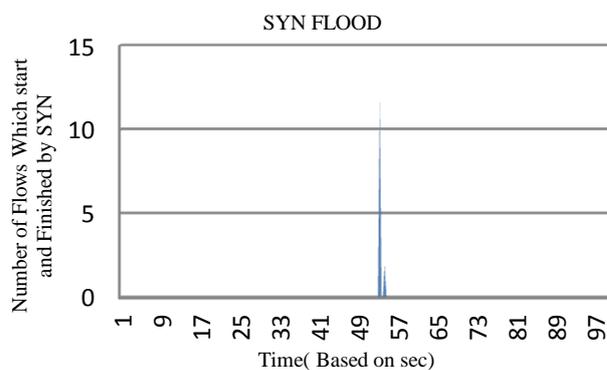


Figure 5: Unsuccessful TCP Flows which Started by SYN Packets

Figure 6 shows huge numbers ICMP flows which opened but during communication the attacker never answer so it will report this reaction as Smurf Attack (DDOS Attack).

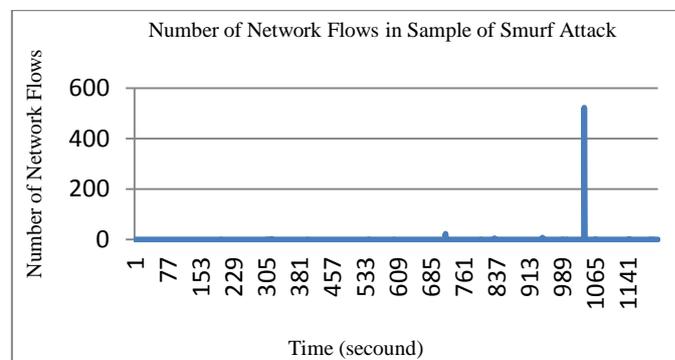


Figure 6: Number of Network Flows in Smurf Attack

## 5. Conclusion

This proposed solution achieved to give a solution for detecting network attacks (five types of DOS/DDOS attack tested), and it was implemented and tested on DARPA traffic sample. The result of testing shows it can detect defined attack with rate of detection 100% and false negative and false positive errors is 0%.

It used network flow as input of system to have the best and useful input and also less complexity in order to achieve faster detection time, because monitoring packets or bytes during detection makes system busy with huge size of data. Correlation algorithm used as engine of NIDS to find critical relation between network flows.

This engine monitoring specific number of network flows in a same time to reduce false errors.

Correlation algorithm which applied inside of detection engine makes this system to be unique because, it monitors every 10 sec of network traffic. The size of window came from the research and also testing, in testing phase it shows if window size in small (for example 2 sec) it may not detect some of the attacks because the distance between network flows in middle of attack may exceed 2 sec and also it should not be big (for example 1 min) because reaction of attack will be invisible or hard to detect.

Another point which makes this technique unique is smooth movement of window which makes system not to lose any suspicious network flows. By these achievements, this project made the rate of detection higher and also decreases false alarms.

This project just applied on DARPA traffic sample and tested with five types of DOS/DDOS attacks. Improvement of this project in future could be little changes to make the system working in more real time environment. Also there could be an upgrade in signature to detect more attacks.

## 6. References

- [1] Abad. C., Taylor. J., Sengul. C., Yurick. W., Zhou, Y., and Rowe, K. (2003). Log Correlation for Intrusion Detection: A Proof of Concept, Proceedings 19th Annual Computer Security Applications Conference, 255-264.
- [2] Molina, J., and Cukier, M. (2009). Evaluating Attack Resiliency for Host Intrusion Detection Systems. Journal of Information Assurance and Security, volume 4, no 1, 001-009.
- [3] Duncombe, D., Mohay, G., Clark, A. (2006). Synapse: Auto-correlation and Dynamic Attack Redirection in an Immunologically-inspired IDS, In Proc. Fourth Australasian Information Security Workshop (Network Security) (AISW 2006), Hobart, Australia. CRPIT, Volume 54. 135-144.
- [4] Waizumi, Y., Kudo, D., Kato, N., Nemoto, Y. (2005). A New Network Anomaly Detection Technique Based on Per-Flow and Per-Service Statistics, In Proceedings CIS IEEE, 252-259.
- [5] Mark, A. L., Crovella, M., Diot, C., (2004). Characterization of Network-Wide Anomalies in Traffic Flows, IMC '04 Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, Taormina, Sicily, Italy, 201-206.
- [6] Abad, C., Taylor, J., Zhou, Y., Sengul, C., Rowe, K., Yurick, W. (2003). Log Correlation for Intrusion Detection: A Proof of Concept, Computer Security Applications Conference, 2003. Proceedings. 19th Annual, IEEE, 255-264.
- [7] Kim, M. S., Kong, H. J., Hong, S.C., Chung S. H, and Hong. J. W. (2004). A Flow-based Method for Abnormal Network Traffic Detection, Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP, Seoul, South Korea, Volume 1, 599-612.
- [8] Lakhina. A., Crovella. M., and Diot. C. (2004). in Proc. Internet Measurement Conference, 201-206.
- [9] J. Rajahalme, A. Conta, B. Carpenter and S. Deering (2004-03). RFC 3697 - IPv6 Flow Label Specification
- [10] J. Quittek, JT. Zseby, B. Claise, and S. Zander (2004-10). RFC 3917 - IPFIX Requirements
- [11] Strayer. W.T., Jones. C., Schwartz. B., Edwards. S., Milliken. W., and Jackson. A. (2007). Efficient Multi-Dimensional Flow correlation, 32nd IEEE Conference on Local Computer Networks, LCN 2007, Dublin. 531-538.
- [12] Giani. A., Souza. I. G. D., Berk. V., and Cybenko. G. (2008). Attribution and Aggregation of Network Flows for Security Analysis, Proceedings of the SPIE Vol. 6201, Sensors and Command, Control, Communications, and Intelligence.
- [13] Peng. T., Leckie. C., and Ramamohanarao. K. (2004). Proactively Detecting Distributed Denial of Service Attacks Using Source IP Address Monitoring, Proceedings of the Third International IFIP-TC6 Networking Conference (Networking 2004), 771-782.
- [14] Shin. M. S., and Jeong. K. J. (2006). Alert Correlation Analysis in Intrusion Detection, in Proc. ADMA, 1049-1056.
- [15] Lincoln Laboratory, Massachusetts Institute of Technology 2008 [http:// www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/attackDB.html](http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/attackDB.html)



# A Generic Attribute-Improved RBAC Model by Using Context-Aware Reasoning

Chen-Chieh Feng and Liang Yu

Department of Geography, National University of Singapore, Singapore

**Abstract** - Traditional role-based access control models (RBAC) are restricted in certain domains and difficult to extend for other applications. The problem is exacerbated with the need to use different RBAC models to manage a variety of resources. Rather than changing the entire RBAC, we propose a generic access control model that takes the advantage of RBAC's reasoning capability and then extends it by adding attributes which significantly improve the inference process. An algorithmic framework is depicted with customizable interfaces that adapts to the requirements of the users. The generic access control model potentially can be adapted to a broader range of user requirements.

**Keywords:** access control model, RBAC, attribute

## 1 Introduction

Access control has been recognized as an important feature of data-centric applications where data are required to be shared under a certain policy [1, 2]. In an information system, the typical use case involves a user executing an operation over an object. The access control engine is expected to give a yes or no answer based on a set of rules. Many fundamental models have been proposed among which the role-based access control (RBAC) model [3] is the most popular mainly because it supports reasoning on the *user-role* and *superRole-subRole* relations.

RBAC model has been improved to meet requirements of various information systems. One type of extension explicitly considers spatial, temporal, and spatiotemporal dimensions. Access control models based on spatial attributes, e.g., SRBAC [4] or GEO-RBAC [5], are used in spatial applications. These models restrict a user's roles according to the spatial region in which the user is located and are extremely useful in wireless location based applications [5]. Temporal role-based access control (Temporal-RBAC) model [6] adds temporal restrictions to the user roles and the hierarchical relations between two roles [7]. The restrictions make it possible to dynamically enable or disable relations between users, roles, and rules, and to automatically disable

the access to the associated element after the life-cycle is over. Spatial-temporal role-based access control (STRBAC) has also been proposed [8] to restrict the roles of both subject and object according to the spatiotemporal information.

A second type of RBAC extension considers workflow in organizations, such as task role-based access control (Task-RBAC) [9]. In Task-RBAC, a task reflects different job assignment of a workflow and is a finer unit than a role. It forms the middle layer between subject and object. Organization role-based access control (ORBAC) [10] suggests that all the assignments and authorizations have to be associated with organizations. ORBAC is further extended to incorporate a mechanism to handle access control in a distributed environment, which was named O2O [11].

Despite of these achievements, existing access control models still suffer from the following shortcomings in an enterprise application:

- 1) The authorization is only performed at an abstract level. It only uses abstract elements such as role, activity and view. A concrete element such as a real user cannot be authorized directly. The authorization process is thus inflexible when we need to authorize a privilege to a concrete element.
- 2) The models described above cannot be easily extended or customized. Every model requires a set of compulsory elements, e.g., ORBAC requires that every role is under one or more organizations while Task-RBAC requires every role is associated with a task. This makes it difficult to consider more elements or remove some elements from a model.
- 3) Different models are not subject to the same basic model, making communications between different access control systems difficult. One may argue that the O2O model [11] solve this problem. However, O2O requires all the organizations to use the ORBAC model. A user from system A should be defined in a virtual organization A-B to gain authorization in system B. Nevertheless one organization always has no knowledge about users from another.

Other than RBAC-based models, recently a more flexible model based on a set attributes that an user could prove to have (e.g., clearance level), termed attribute-based access control (ABAC) [12], has been proposed to either replace RBAC or at least simplifies RBAC and makes it more flexible [13]. Attribute has a simple form so it is easy to be shared by different organizations. A system does not have to authorize a user in advance but authorize by the user's

This work was supported the project *CyberInfrastructure of Center for Environmental Sensing and Modeling @ Singapore-MIT Alliance for Research and Technology (CENSAM @ SMART)*.

Chen-Chieh Feng is with Department of Geography, National University of Singapore: 1 Arts Link, Singapore 117570. (e-mail: geofcc@nus.edu.sg).

Liang Yu was with Department of Geography, National University of Singapore. (e-mail: liangyu.geo@gmail.com).

attributes instead. However, the trade-off for this flexibility is the complexity of cases that must be considered – negotiation between parties must establish trust using elements' attributes and ensure that parties use the same definition for attributes [13]. ABAC also suffers from the lack of support for role-based reasoning.

In this paper, we present a generic access control model that combines the advantages from RBAC and ABAC. We extend the reasoning ability of RBAC by introducing a generic hierarchical relation rather than specific relations such as *user-role*. The notion of attribute from ABAC is used to form *context* which determines the applicable scope of rules. A standard validation algorithm with customizable interfaces is proposed. The customizable interfaces are important feature of the proposed generic model considering most access control requirements are often discovered after the implementation of a system [14].

The remainder of the paper is organized as follows. Section 2 presents the generic model as well as its fundamental definitions and theorems. Section 3 discusses how to add context to the model and use it to supervise the reasoning process. Section 4 proposes a framework of validation algorithm and demonstrates a prototype implemented with Java, and Section 5 concludes the work and lays out the future work.

## 2 Generic attribute-based RBAC model

### 2.1 A simple use case

The simplest access control case can be described as a binary relation between a *subject* and an *object*, where a subject refers to a user and an object refers to various resources. To distinguish one access privilege from another, e.g., read from write, the relation becomes a triple which include one more element, i.e., *operation*. An authorization is then described as a pattern  $p = (sub, opt, obj)$ . The concept of *role* was introduced to group subjects. By assigning privileges to a role, all subjects assuming the role are automatically granted the privileges which have been assigned to the role. A simple example is as follows:

- *Tom* is an *analyst*
- An *analyst* can read *annualReport.xls*
- Inference: *Tom* can read *annualReport.xls*

The statements above contain three essential element types of RBAC:

- Individuals: *Tom*, *analyst*, *read*, and *annualReport.xls*
- Type assumption relation:  $r_1 = \langle analyst, Tom \rangle$ , indicating that *Tom* is an *analyst*
- Approved authorization pattern:  $p_1 = \{subject=analyst, operation=read, object=annualReport.xls\}$

Note that individual elements in  $p_1$  are its attributes and can be denoted as  $p_1.subject = analyst$ ,  $p_1.operation = read$ , and  $p_1.object = annualReport.xls$ .

If *Tom* wants to read the *annualReport.xls*, the validation process needs to decide if pattern  $p_2 = \{subject=Tom, operation=read, object=annualReport.xls\}$  is

approved. Given the type assumption relation  $r_1$ , the pattern  $p_2$  does not have to be defined directly but asserted by  $p_1$ . We note this as  $assert(p_1, p_2) = true$ . The assertion process can be further utilized for cases in which role hierarchy defined as a sub-role inherits from its super-roles all the features. For example, assume that *seniorAnalyst* is a sub-role of *analyst*, denoted as  $r_2 = \langle analyst, seniorAnalyst \rangle$ , and *John* is a *seniorAnalyst*, denoted as  $r_3 = \langle seniorAnalyst, John \rangle$ . The request for permitting *John* to read the *annualReport.xls*, denoted as pattern

- $p_3 = \{subject = John, operation = read, object = annualReport.xls\}$ ,

can also be approved without specifying explicitly the pattern  $p_3$  but again asserted by  $p_1$  ( $assert(p_1, p_3) = true$ ) because the existence of both  $r_2$  and  $r_3$ .

Note that the three patterns  $p_1$ ,  $p_2$ , and  $p_3$  serve different purposes. The  $p_1$  is an approved pattern for authorization while the  $p_2$  and the  $p_3$  are undecided input patterns for validating requests. The three relations carry different semantics. The relation  $r_1$  and  $r_3$  are *type-assumption* relations between a concrete and an abstract element, while the relation  $r_2$  indicates an inheritance relation between two abstract elements.

The relations stated above stands for a partial ordering, termed *hierarchy*, between the first and the second elements, denoted by '>' below.

- $r = \langle e_1, e_2 \rangle \equiv e_1 > e_2$

Using  $r_1 - r_3$  as examples,

- $r_1 = \langle analyst, Tom \rangle \equiv analyst > Tom$
- $r_2 = \langle analyst, seniorAnalyst \rangle \equiv analyst > seniorAnalyst$
- $r_3 = \langle seniorAnalyst, John \rangle \equiv seniorAnalyst > John$

The following axiom follows immediately after  $r$  is defined:

- $e_1 \geq e_2 \equiv (e_1 = e_2) \cup (e_1 > e_2)$

The transitivity axiom of a partial ordering relation can then be used to infer new hierarchical relations. The validation process is to search the approved patterns which assert the input patterns, where the hierarchical relations are used for reasoning. In the above example, the assertion process can be decomposed as:

- $assert(p, p') \equiv (p.subject \geq p'.subject) \cap (p.operation = p'.operation) \cap (p.object = p'.object)$

### 2.2 Improvement and Formalization

The original RBAC model discussed in Section 2.1 has been improved to meet more sophisticated access control requirements. These efforts can be classified into three categories according to their functionalities:

- 1) Elements to elaborate the hierarchical relations. More element types are entitled to have roles, e.g., operation and object in ORBAC are now associated with *Activity* and *View*.
- 2) Elements to constrain the domain. For example, in the Temporal-RBAC, a role is only enabled in a temporal region, so does the user-role assignment. In the ORBAC model, a role is enabled within an organization, so does

a superRole-subRole relation.

- 3) Elements to resolve the conflict. This type of improvement incorporates richer authorization types as well as the solutions for resolving conflicts between them. *Prohibition* and *Obligation* are two exemplar authorization types, in addition to *Permission*, being considered [15]. The type can also be considered as a specific attribute, so the pattern  $p_3$  could be reformatted as

$$p_3 = \{type=permit, subject=analyst, operation=read, object=annualReport.xls\}.$$

The first point indicates that the hierarchical relation can be applied to any entities. With that, new patterns can be inferred by replacing every element with its sub-elements. The assert method can be rephrased as:

$$\begin{aligned} \text{assert}(p, p') &\equiv (p.type \geq p'.type) \cap \\ &(p.subject \geq p'.subject) \cap (p.operation \geq p'.operation) \\ &\cap (p.object \geq p'.object) \end{aligned}$$

Type can also have hierarchies, e.g., *Obligation* can be seen as a sub-type of *Permit*. Based on the analysis above, the essential elements in all the RBAC-based models are depicted in Figure 1.

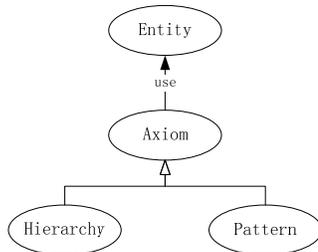


Figure 1. Basic elements in a access control model

**Definition 1.** A generic role-based access control (GRBAC) model is a tuple  $GRBAC = (ES, AS)$ , where  $ES$  is a set of entities and  $AS$  is a set of axioms. Each  $AS$  is also a tuple  $AS = (HS, PS)$ , where  $HS$  is the hierarchy set and  $PS$  is the approved pattern set.

In the generic model, *entity* represents any individuals since it can be divided into different categories in a concrete model for the management purpose. For example, *user*, *dataset*, *service*, *task*, *organization*, *space*, and *time* are all entities.  $HS$  is a set of hierarchical relations such as the  $r_1$  and  $r_2$  that mentioned above.  $PS$  is a set of approved patterns such as  $p_1$ .

**Definition 2.** A hierarchical relation  $hr = \langle e_1, e_2 \rangle$  is a partial order relation between two entities,  $e_1$  and  $e_2$ , for which  $e_1$  is superior to  $e_2$ , denoted as  $e_1 > e_2$ . The  $e_1$  and the  $e_2$  can be considered as attributes to  $hr$ , denoted as  $hr.super = e_1$ ,  $hr.sub = e_2$ .

Two most common hierarchical relations are type assumption between subjects and inheritance of roles. Hierarchical relations can also be established across different entity types. For example, in the hierarchical relation  $r_4$  below, a role (*analyst*) is associated to a task (*analysis*) as its subordinate entities:

- $r_4 = \langle analysis, analyst \rangle \equiv analysis > analyst$ , if
- $p_4 = \{subject=analysis, operation=read, object=annualReport.xls\}$

The  $\text{assert}(p_4, p_3)$  is thus evaluated to be true because

- $analysis > analyst > seniorAnalyst > John$

The partial order relation can also be applied to the pattern. The fact that an approved pattern asserts an input pattern means it has a same or broader semantics.

- $\text{assert}(p, p') \equiv p \geq p'$

**Definition 3:** Pattern  $p = \{a_1, a_2, \dots, a_n\}$  is a collection of attributes. Each attribute is a tuple  $a_i = \langle k_i, v_i \rangle$ , where  $k_i$  is the key and  $v_i$  is the value. It can be noted as  $p.k_i = v_i$ .

Note that both *hierarchy* and *pattern* are expressed by *attribute*, which is a key-value pair  $\langle k, v \rangle$ . The key is used to uniquely identify the attribute type and the typical key is an attribute name denoted by a string. The value can be simply an entity or a collection of entities, which means that an attribute can have sub-attributes.

An entity in GRBAC can belong to different entity types. This design choice allows the model to maintain maximum flexibility for defining a pattern. For example, a *service* could be an *object* in a pattern while as a *subject* in another. It also enables the authorization over concrete entities rather than just abstract ones.

Attributes of hierarchical relation and pattern can also be extended. To understand how a pattern is extended, consider the following example with temporal attributes. Assuming two input patterns  $ip_1$  and  $ip_2$  and an approved pattern  $ap_1$  are specified:

- $ip_1 = \{type=permit, subject=Tom, operation=read, object=map1, time='13^{th} Jan 2009'\}$
- $ip_2 = \{type=permit, subject=Tom, operation=read, object=map1, time='13^{th} Jan 2010'\}$
- $ap_1 = \{type=permit, subject=analyst, operation=read, object=SpatialData, time='2009'\}$

Given that *Tom* plays role *analyst* and *map1* is considered as *SpatialData*,  $ip_1$  is asserted by  $ap_1$  while  $ip_2$  is not, because  $ip_1$ 's lifecycle is within the time period of  $ap_1$  but  $ip_2$ 's is not. By using the hierarchical relation ' $2009' \geq '13^{th} Jan 2009'$ ' establishes while ' $2009' \geq '13^{th} Jan 2010'$ ' does not. The rule can be described as:

$$\begin{aligned} ap \geq ip &\equiv (ap.type \geq ip.type) \cap (ap.subject \geq ip.subject) \\ &\cap (ap.operation \geq ip.operation) \cap \\ &(ap.object \geq ip.object) \cap (ap.time \geq ip.time) \end{aligned}$$

As discussed earlier in this section, different access control models can be applied in the same environment and they have to communicate with each other. There is a possibility that an attribute defined in one model does not exist in a second model. For example,

- $ap_2 = (type=permit, subject=analyst, operation=read, object=SpatialData)$

This happens when a user from a Temporal-RBAC system wants to access a regular RBAC system. Since each attribute is used as a restriction, the pattern  $ap_2$  should not be limited in any temporal region. Apparently,  $\text{assert}(ap_2, ip_1)$  and  $\text{assert}(ap_2, ip_2)$  can both be established.

**Theorem 1:** If  $ap = \{a_1, a_2, \dots, a_n\}$ ,  $ap \succcurlyeq ip \equiv \forall a_i = \langle k_i, v_i \rangle, (ip.k_i \neq \Phi) \cap (v_i \succcurlyeq ip.k_i)$ , where  $ap$  is an approved pattern,  $ip$  is an input pattern,  $a$  is an attribute,  $k$  is a key,  $v$  is the value of a key, and  $\Phi$  is null. The input pattern can have more attributes but every attribute in the authorization pattern should be found in the input pattern. For example,

- $ap_3 = \{subject=analyst, operation=read, object=SpatialData, organization=Group1\}$

The authorization pattern  $ap_3$  does not assert  $ip_1$  or  $ip_2$  because they both lack the attribute *organization*. In our generic model, a pattern is not required to have minimum number of attributes, but sometimes the pattern is meaningless without sufficient attributes. Using the following five patterns as an example:

- 1)  $p = \{type=permit, subject=Administrator\}$
- 2)  $p = \{type=permit, operation=search\}$
- 3)  $p = \{type=permit, object=SpatialData\}$
- 4)  $p = \{subject=analyst, operation=read, object=SpatialData\}$
- 5)  $p = \{type=permit\}$

The pattern 1 can be understood as the administrator is permitted to do everything; pattern 2 means an access request will be permitted as long as the operation is *search*; pattern 3 means the *SpatialData* is publicly open to any access. However, the pattern 4 and 5 are not acceptable because the former lacks the authorization type while the later does not have any other attributes besides a type. Thus, we demand that a pattern must have at least two attributes with one of them indicating the authorization type. The other one must be associated with an entity.

In this section the generic RBAC model has been defined with the minimum elements and a flexible authorization model. The validation is decomposed to a set of reasoning processes. The generic model can be extended to support more elaborated authorization rules, which would be discussed in the next section.

### 3 Context-aware Reasoning

Attributes of an authorization pattern described in Section 2 are not always independent from each other. Certain attributes may have special status as they affect the applicability of a pattern or a hierarchy, and thus the rest of the attributes. Termed *context* in GRBAC, these attributes demand special treatment because the authorization process is significantly affected by restricting the elements it utilizes. In general, additional attributes require additional reasoning process when comparing two patterns. In addition, an attribute as a context can affect the reasoning process in other ways:

- 1) A context can change the hierarchies established for the input. Using the following patterns and the hierarchical rule as an example,
  - $ap_1 = \{type=permit, subject=analyst, operation=read, object=SpatialData\}$
  - $ip_1 = \{type=permit, subject=Tom, operation=read, object=SpatialData\}$

- $hr_1 = \{super=analyst, sub=Tom, organization=Group2\}$
- $ap_2 = \{type=permit, subject=analyst, operation=read, object=SpatialData, organization=Group1\}$
- $ip_2 = \{type=permit, subject=Tom, operation=read, object=SpatialData, organization=Group1\}$

For  $ap_1$  and  $ip_1$ , the hierarchy  $hr_1$  can be established and  $assert(ap_1, ip_1)$  evaluates to true. However, the addition of context  $organization = Group1$  in  $ap_2$  and  $ip_2$  causes the assertion  $assert(ap_2, ip_2)$  fails because  $hr_1$  is defined in another organization (*Group2*). The reasoning process using the transitive relation is also affected because all the antecedent hierarchical relations are required to be applicable to the input.

- 2) A context can change the entities applicable to the input. For example, a user created with a three-day lifespan will be invalidated after three days. A role associated with one organization means it is not applicable to another.

Below formal definitions of the context and its associated rules are given.

**Definition 4:** *Context* is a specific condition reflected by a collection of attributes,  $ct = \{a_1, a_2, \dots, a_n\}$ .

Similar to *pattern*, the attributes of a context are also referred to by their keys. In a validation process, the reasoning process is controlled by the context attributes, which means 1) context attributes have a higher priority and 2) context attributes affect the applicability of common attributes.

The validation process is to compare the context from approved pattern and the context from the input pattern to see if the latter is contained by the former one. A context is attached to an axiom as an attribute. For example,

- $ip_1 = \{type=permit, subject=analyst, operation=read, object=SpatialData, context=\{organization=Group2\}\}$

The *organization* attribute has been moved to the context, which means that this attribute will affect the reasoning through other attributes. The *context* attribute is not compulsory and an axiom without a context means it can be applied in any contexts. The partial operator ' $\succcurlyeq$ ' for hierarchical relation also applies to context. To assert that one context is superior to another, we need to compare each of its attributes are superior to those of another.

**Theorem 2:** If  $ct_1 = \{a_1, a_2, \dots, a_n\}$ ,  $ct_1 \succcurlyeq ct_2 \equiv \forall a_i = \langle k_i, v_i \rangle, (ct_2.k_i \neq \Phi) \cap (v_i \succcurlyeq ct_2.k_i)$ .

A context can have sub-contexts. The context affects the way of reasoning on each single attribute. A hierarchical relation is used for reasoning only if it is applicable in a specific context. Here the expression  $e_1 \succcurlyeq e_2[ct]$  denotes a hierarchical relation between elements  $e_1$  and  $e_2$  is established under a specific context  $ct$ .

**Theorem 3:**  $e_1 \succcurlyeq e_2[ct] \equiv \exists hr = (e_1, e_2) \in HS, (hr.context \succcurlyeq ct)$ . It means a conditional partial order relation.

**Theorem 4:** If  $e_1 \succcurlyeq e_2[ct]$  and  $e_2 \succcurlyeq e_3[ct]$ , then  $e_1 \succcurlyeq e_3[ct]$ .

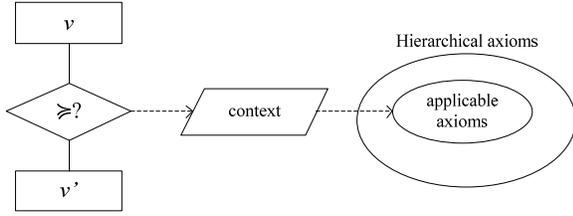


Figure 2. Use context to assert if the hierarchical relation between two attribute values ( $v$  and  $v'$ ).

As shown in Figure 2, the context defines a subset of the whole axiom set that becomes the axiom set applicable for reasoning. Theorem 4 implies that the context-aware hierarchical relation is a transitive relation. With theorem 3 and 4, new hierarchical relations under context  $ct$  can be inferred. In Section II we standardize the validation without considering the context based reasoning. Here the validation process is improved as follows:

**Theorem 5:** If  $ap = \{a_1, a_2, \dots, a_n\}$ ,  $assert(ap, ip) \equiv \forall a_i = \langle k_i, v_i \rangle, (ip.k_i \neq \Phi) \cap (v_i \geq ip.k_i[ip.context])$ .

We use the following example to demonstrate the decomposed process. The example contains two patterns (i.e.,  $ap_1$  and  $ip_1$ ) that have organization, space, and time as the attributes and several hierarchical relations between authorization type ( $hr_1$ ), role ( $hr_2$ ), operation ( $hr_3$ ), resource ( $hr_4$ ), and organization ( $hr_5$ ). All hierarchical relations except  $hr_1$  have a time attribute to indicate their life time.

- $ap_1 = \{type=obligation, subject=analyst, operation=access, object=spatialData, context=\{organization=Group1, time=T_1\}\}$
- $ip_1 = \{type=permit, subject=Tom, operation=read, object=map1, context=\{organization=Group2, time=T_2\}\}$
- $hr_1 = \{super=obligation, sub=permit\}$
- $hr_2 = \{super=analyst, sub=Tom, context=\{time=T_3\}\}$
- $hr_3 = \{super=access, sub=read, context=\{time=T_4\}\}$
- $hr_4 = \{super=spatialData, sub=map1, context=\{time=T_5\}\}$
- $hr_5 = \{super=Group1, sub=Group2, context=\{time=T_6\}\}$

The assertion  $assert(ap_1, ip_1)$ , according to Theorem 5, can be initially decomposed into five sub-processes:

- $assert(ap_1, ip_1) \equiv (1) (obligation \geq permit[ct_{ip1}]) \cap (2) (analyst \geq Tom [ct_{ip1}]) \cap (3) (access \geq read [ct_{ip1}]) \cap (4) (spatialData \geq map1 [ct_{ip1}]) \cap (5) (ct_{ap1} \geq ct_{ip1} [ct_{ip1}])$

- (1) The hierarchical relations between authorization types are global and not subject to any context. Thus, sub-process 1 always returns *true*.
- (2) The hierarchy of role assignment is subject to a context time. To make it less complicated, their hierarchies are explicitly defined as  $hr_2$ ,  $hr_3$ , and  $hr_4$ . Thus the sub-processes 2, 3, and 4 can be replaced by asserting the hierarchical relations between their contexts, i.e.,
  - (6)  $(ct_{hr2} \geq ct_{ip1} [ct_{ip1}]) \cap (7) (ct_{hr3} \geq ct_{ip1} [ct_{ip1}]) \cap (8) (ct_{hr4} \geq ct_{ip1} [ct_{ip1}])$ .

- (3) Because that the role assignment is only subject to the attribute time, the sub-processes 6, 7 and 8 can be replaced by asserting the hierarchical relations between the time expressions, i.e.,

- (9)  $(T_3 \geq T_2) \cap (10) (T_4 \geq T_2) \cap (11) (T_5 \geq T_2)$ .

The comparison of two time values is not related to any context, thus the context expressions  $[ct_{ip1}]$  for sub-processes (9), (10), and (11) are removed. Note that the example is a trivial case as the context values (time) become the attributes for comparison. In other cases, a context may have sub-context as stated in Theorem 2. An example is shown in the sub-process (12) below.

- (4) According to the Theorem 2, the sub-process 5 can be decomposed as

- (12)  $(Group1 \geq Group2 [ct_{ip1}]) \cap (13) (T_1 \geq T_2)$ .

Given the hierarchical relation  $hr_4$ , the sub-process (12) can be replaced by

- (14)  $(T_6 \geq T_2)$

Again, the context restriction in sub-process 13 and 14 are removed because they are not related to any context. Finally, the assertion process can be rephrased as a series of comparison between time expressions.

- $assert(ap_1, ip_1) \equiv (T_3 \geq T_2) \cap (T_4 \geq T_2) \cap (T_5 \geq T_2) \cap (T_1 \geq T_2) \cap (T_6 \geq T_2)$

In existing research the time-related computation has been well investigated. They are thus not discussed in this paper. Readers interested in these computations are referred to works in [6-8]. This use case does not employ complicated rules such as restrictions of time, space, organization. It also eliminates the complexity of reasoning process for those hierarchical relations that have not been explicitly defined but can be inferred. It can be implemented according to theorem 4, which would be demonstrated in the next section.

## 4 Validation algorithm and prototype

Based on the rules defined in the last section, this section demonstrates how GRBAC can be implemented and used in a practical environment. The demonstration focuses on searching the authorization patterns that assert an input pattern. The execution workflow is depicted in Figure 3.

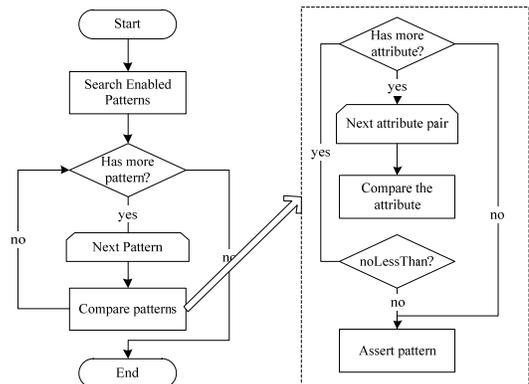


Figure 3. Execution Workflow for Validation

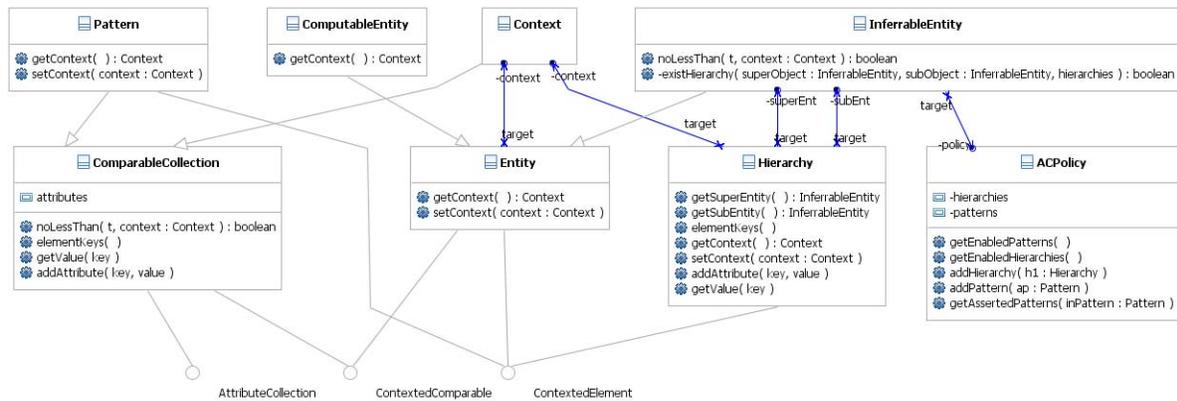


Figure 4. Class Diagram of the Prototype

The algorithm can be roughly divided into three steps:

- 1) Identify all the enabled authorization patterns.
- 2) For each candidate authorization pattern, decide if it asserts the input pattern by comparing each pair of attributes with the same key.
- 3) Return all the passed authorization patterns as the result.

Step 2 plays a crucial role in authorizing resources and thus warrants further explanation specifically on how the search for existing hierarchical relations and the inference of hierarchical relations through reasoning are completed.

In the prototype, the second step is implemented in a way that the whole process can be decomposed into more dedicated units, which in turns are open for customization and then change the behaviours of the access control system.

Figure 4 shows the UML diagram of the core classes in our prototype which is implemented with Java. Several new elements are introduced in the prototype systems in addition to those defined in Section 2 and 3:

- 1) ACPolicy is introduced as a rule system for access control, which consists of *hierarchies* and *patterns*.
- 2) Three interfaces and one abstract class are introduced to handle attributes. Interface ContextedElement indicates that an element is subject to a context object; Interface ContextedComparable indicates an object is comparable with another under a certain context; Interface AttributeCollection is implemented by Context and Pattern which means an object can be treated as a *key-value* collection; Abstract class ComparableCollection implements the algorithm of comparing two collections.
- 3) The Entity class is extended by two sub-classes InferrableEntity and ComputableEntity. The sub-class InferrableEntity implements a comparing method (noLessThan) that compares two entities based on a set of hierarchical relations. It takes the hierarchical rules (*hierarchies*) in an ACPolicy as input so that its instances only need to customize contexts. The ComputableEntity refers to the entities which can be compared by algorithms. These entities usually do not have any context.

```

Class: ACPolicy
public Collection<Pattern> getAssertedPatterns(Pattern inPattern) {
    Collection<Pattern> patterns = this.getEnabledPatterns();
    Collection<Pattern> result = new ArrayList<Pattern>();
    for (Pattern pattern : patterns) {
        if (pattern.noLessThan(inPattern, inPattern.getContext())) {
            result.add(pattern);
        }
    }
    return result;
}

Class: ComparableCollection
public boolean noLessThan(Object another, Context context) {
    Set<Object> keys = this.elementKeys();
    if (another instanceof AttributeCollection) {
        boolean noLessThan = true;
        AttributeCollection attrCollection = (AttributeCollection)
another;
        for (Object key : keys) {
            ContextedComparable thisValue = (ContextedComparable) this
                .getValue(key);
            ContextedComparable anotherValue = (ContextedComparable)
attrCollection.getValue(key);
            if (!thisValue.noLessThan(anotherValue, context)) {
                noLessThan = false;
                break;
            }
        }
        return noLessThan;
    }
    return false;
}

Class: InferrableEntity
public boolean noLessThan(Object t, Context context) {
    Entity entity = (Entity) t;
    if (this.equals(entity)) {
        return true;
    }
    Collection<Hierarchy> hierarchies =
this.policy.getEnabledHierarchies();
    Collection<Hierarchy> applicableHierarchies = new
ArrayList<Hierarchy>();
    for (Hierarchy hierarchy : hierarchies) {
        if (hierarchy.getContext().noLessThan(context, context)) {
            applicableHierarchies.add(hierarchy);
        }
    }
    return existHierarchy(this, (InferrableEntity) entity,
hierarchies);
}

private boolean existHierarchy(InferrableEntity superObject,
InferrableEntity subObject, Collection<Hierarchy> hierarchies) {
    for (Hierarchy hierarchy : hierarchies) {
        if (hierarchy.getSuperEntity().equals(superObject)) {
            if (hierarchy.getSubEntity().equals(subObject)) {
                return true;
            }
        }
        else if (existHierarchy(hierarchy.getSubEntity(),
subObject,
hierarchies)) {
            return true;
        }
    }
}
return false;
}

```

Figure 5. Methods for the Validation

Figure 5 list three important methods for the validation. The validation starts from the `getAssertedPatterns` method of `ACPolicy`. It gets all the enabled patterns in the policy and iteratively asserts if each of them assumes a `noLessThan` relation with the input pattern under its context.

The `noLessThan` is an abstract method defined in interface `ContextedComparable`. It has two implementations. The first is in the class `ComparableCollection`, where it is decomposed into iterative calls to all the `noLessThan` methods of its attribute values and compared with the values of the same attribute keys from another collection. The second implementation is in the class `InferrableEntity`, where it uses the hierarchical relations defined in the policy to decide if an entity is in a superior position comparing to another one. The `existHierarchy` method performs the reasoning on the hierarchical relations to infer both direct and indirect hierarchical relations between two entities.

Users can customize the classes to handle more elaborate entities to meet their systematic requirements. The `noLessThan` method can be overridden to support more complicated comparison algorithm. The most convenient way is to make them sub-classes of either `InferrableEntity` or `ComputableEntity` and reuse the algorithms. The rules can be customized by adding customized context or re-implementing the `noLessThan` method. Entities such as *Role*, *View*, *Dataset*, *Organization*, should extend the `InferrableEntity` and customized the contexts for their instances. The hierarchical relations between entities should also be defined. For custom entities, such as customized temporal or spatial representations, the `ComputableEntity` should be extended and the `noLessThan` method should be customized. For example, for the use case we discussed in the last section, a simple temporal entity can be created as a subclass of `ComputableEntity` which contains the starting and ending date. The algorithm logic used in `noLessThan` method could be:

- $T_1 \succcurlyeq T_2 \equiv T_2.startingDate \succcurlyeq T_1.startingDate \cap T_1.endingDate \succcurlyeq T_2.endingDate$

Similarly, spatial algorithms can be reused for spatial-related authorization, e.g., to create a geometry class and reuse the algorithm to assert if a spatial region contains another one.

## 5 Conclusion and future work

We have developed a generic RBAC (GRBAC) model based on context-based reasoning. The traditional role-based architecture is replaced by more generic *hierarchy* and the authorization expression is no longer subject to a fixed sequence. A partial order relation pins the foundation of reasoning and is applied to all elements. The reasoning process is supervised by *context* which is composed of a set of attributes. A prototype has been implemented according to a couple of theorems which can be reused to accommodate more attributes to solve domain-specific access control problems. Various access control functions can be integrated under this framework. Compared to the former works, GRBAC focuses on the flexibility rather than specific domain

models. Yet, more remain to be done to adapt it well to a real information system especially in a distributed environment, such as the exchange of authorization information within a distributed environment, the development of advanced rule editing tools, and investigating of the efficiency problem.

## 6 References

- [1] Barth, A., et al., *Privacy and utility in business processes*. 20th IEEE Computer Security Foundations Symposium. 2007. 279-291.
- [2] Basin, D. *Model driven security*. in *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*. 2006.
- [3] Ferraiolo, D.E., J.A. Cugini, and D.R. Kuhn. *Role-based access control (RBAC): features and motivations*. in *Proceedings of 11th Annual Computer Security Applications Conference, 11-15 Dec. 1995*. 1995. Los Alamitos, CA, USA: IEEE Comput. Soc. Press.
- [4] Hansen, F., V. Oleshchuk, and Ieee. *Spatial role-based access control model for wireless networks*, in *2003 Ieee 58th Vehicular Technology Conference, Vols1-5, Proceedings*. 2004. p. 2093-2097.
- [5] Damiani, M.L., et al., *GEO-RBAC: A spatially aware RBAC*. *Acm Transactions on Information and System Security*, 2007. **10**(1).
- [6] Bertino, E., P.A. Bonatti, and E. Ferrari, *TRBAC: a temporal role-based access control model*. *ACM Transactions on Information and Systems Security*, 2001. **4**(Copyright 2002, IEEE): p. 191-223.
- [7] Joshi, J.B.D., et al., *A generalized temporal role-based access control model*. *Ieee Transactions on Knowledge and Data Engineering*, 2005. **17**(1): p. 4-23.
- [8] Ray, I. and M. Toahchoodee. *A spatio-temporal role-based access control model*, in *Data and Applications Security XXI, Proceedings*, S. Barker and G.J. Ahn, Editors. 2007, Springer-Verlag Berlin: Berlin. p. 211-226.
- [9] Oh, S. and S. Park, *Task-role-based access control model*. *Information Systems*, 2003. **28**(6): p. 533-562.
- [10] El Kalam, A.A., et al., *Organization based access control*. *Ieee 4th International Workshop on Policies for Distributed Systems and Networks, Proceedings*. 2003. 120-131.
- [11] Cuppens, F., N. Cuppens-Bouahia, and C. Coma, *O2O: Virtual Private Organizations to manage security policy interoperability*, in *Information Systems Security, Proceedings*, A. Bagchi and V. Atluri, Editors. 2006. p. 101-115.
- [12] Karp, A., H. Haury, and M. Davis, *From ABAC to ZBAC: The Evolution of Access Control Models*. *Proceedings of the 5th International Conference on Information Warfare and Security*, ed. E.L. Armistead. 2010, Nr Reading: Academic Conferences Ltd. 202-211.
- [13] Kuhn, D.R., E.J. Coyne, and T.R. Weil, *Adding Attributes to Role-Based Access Control*. *Computer*, 2010. **43**(6): p. 79-81.
- [14] Devanbu, P.T. and S. Stubblebine, *Software engineering for security: a roadmap*, in *Proceedings of the Conference on The Future of Software Engineering*. 2000, ACM: Limerick, Ireland. p. 227-239.
- [15] Cuppens, F., N. Cuppens-Bouahia, and M.B. Ghorbel, *High Level Conflict Management Strategies in Advanced Access Control Models*. *Electronic Notes in Theoretical Computer Science*, 2007. **186**(Compendex): p. 3-26.

# A Fuzzy Clustering Algorithm for Fingerprint Enhancement

C. Obimbo<sup>1</sup> and W. Wang<sup>2</sup>

<sup>1</sup>School of Computer Science, University of Guelph, Guelph, ON, Canada

<sup>2</sup>i365 A Seagate Company, Toronto, ON, Canada

**Abstract** - Fingerprint Identification and Verification are important tools for both forensics (when dealing with crime evidence) and authentication when used as a biometric-identifier, say for access purposes. Although fingerprint authentication is considered an effective method, the quality of the verification is highly dependent upon sample quality. In the case of crimes, since most people would not leave clear footprints, it is important to have an effective enhancing method that would be able to assist narrow down the matching specimen while reducing the false-positives. We use a fuzzy clustering algorithm to enhance fingerprints, and report the encouraging results found.

**Keywords:** Fingerprint enhancement.

## 1 Introduction

In recent years, automatic fingerprint identification and classification has become one of most important biometric technologies and has drawn a substantial amount of attention. This is enhanced by the fact that the increase in collection of biometric data at airports and border crossing renders the amount of data to be compared with as tremendous, so it behooves us to find faster, more efficient algorithms and better described fingerprint samples to be matched.

Fingerprint recognition mainly entails the extraction of patterns of ridges and furrows. More than a hundred characteristics and relationships, which are called minute details, have been identified. Among them, the most commonly used features in fingerprint identification are “ridge ending” and “ridge bifurcation”. Ridge ending is the point at which a ridge ends, Fig. 1(a), whereas ridge bifurcation is the point at which a ridge splits into two ridges Fig. 1(b).

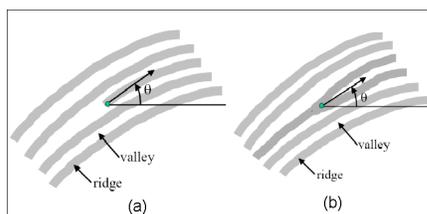


Figure 1 (a) ridge ending (b) ridge bifurcation

Today many algorithms and methods have been proposed in the field of fingerprint image segmentation [1,2,3] and enhancement [4-9]. In order to segment fingerprint area from background area, Asker M. Bazen [11] introduced three pixel features: Coherence, Local Mean and Local Variance, which were also used in our project. They applied an optimized linear classifier, which was trained to determine the weight coefficients of linear combination of these features. The advantage of their algorithm is that whenever the weight coefficients are set, the whole segmentation processing can run under low computational complexity. However for input images from different database, in order to determine the feature coefficients, sample images must be carefully selected to train the system. Hong, L. [12] presented a one-nearest neighbor classifier to classify each block in fingerprint image into recoverable or unrecoverable area based on the frequency of ridge and furrow. Hong, L. [13] also introduced a voting algorithm based on the filtered fingerprint image using a bank of Gabor filters to generate the coarse-level ridge map and unrecoverable region mask.

In order to overcome the drawback of TGF mentioned above, Jianwei, Yang [14] designed a modified Gabor Filter (MGF), which represents the frequency by a band pass filter associated with a bank of low pass filters. Kamei, T. [15] designed a ridge frequency filter and ridge direction filter in the Fourier domain. An energy function for selecting image features is defined by intensities of images obtained with the two filters and a measure of smoothness of the features. By using the image features to minimize the energy function, the enhanced image can be produced from the filtered images.

Zhao, Hao and Li [16] use supervised Support Vector Machines (SVMs) to classify patterns and select typical patterns to train the classifier. They first partition the image into 12 times 12 blocks and the low gray variance background blocks were segmented by the contrast.

In this Paper, we use classic FCM Clustering algorithm with predefined three pixel features to get the fingerprint area and exclude the background area. Because fingerprint image can be viewed as oriented texture, we calculate the local ridge orientation based on some kinds of gradient operators and use the so-called “Local Orientation Image” to present the input fingerprint image. By applying low pass filter on the “Local Orientation Image”, most of the noises can be eliminated or weakened.

We use the modified anisotropic filter [11] to enhance the ridge and furrow information according to the local orientation. Finally we employ information fusion based threshold segmentation methods to crisp partition the enhanced image and get the binary image. Experiment results show that our method is able to separate good fingerprint areas from background and unrecoverable areas and can efficiently enhance the ridge and furrow information to increase the quality of input image significantly.

## 2 Enhancement & Segmentation Algorithm

The implemented system, according to the requirement of fingerprint image enhancement and segmentation, receives the input fingerprint image and then outputs the enhanced fingerprint image and binary image through the four main parts, Input Image Processing, Fuzzy Clustering, Image Enhancement, and Image Segmentation [9].

The following subsections describe them more in detail.

### 2.1 Input Image Processing

Input Image Processing module takes the fingerprint image as input and outputs the normalized fingerprint image with pre-defined mean and variance of gray level. It contains two main steps: Pre-Processing step and Normalization step.

#### 2.1.1 Image Pre-Processing

When starting to process the input fingerprint image, many fingerprint images are normally found to have fixed black shadows and gray shadows (Fig. 3.1(a)), which are probably introduced during the digitalization and may influence the quality of fuzzy clustering and image enhancement significantly. Since these unwanted elements almost appears

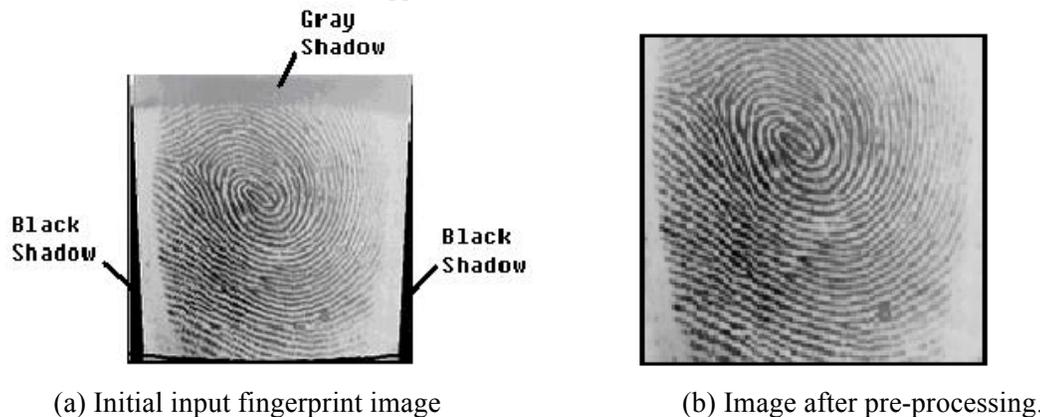


Figure 2.1 Image Pre-processing.

at the same position, we simply cut off (crop) these shadows by choosing the area from (21, 60) to (340, 347) of the original image as the new input image (see Fig. 3.1(b)). If the initial input image does not have these unwanted elements, the image pre-processing module can be bypassed.

#### 2.1.2 Image Normalization

An input fingerprint image is normalized so that it has a pre-specified mean and variance gray-level. After normalization, all the input images should have the standardized intensity value, from which we can easily control the intensity level of enhancement results. The formulas used for normalization are shown as follows:

$$I'(i, j) = \begin{cases} M_0 + \sqrt{VAR_0(I(i, j) - M)^2 / VAR}, & \text{if } I(i, j) > M \\ M_0 - \sqrt{VAR_0(I(i, j) - M)^2 / VAR}, & \text{otherwise} \end{cases}$$

$$M_0 = VAR_0 = 100 \quad (1)$$

where  $M$  represents the mean gray value of input image (2);

$$M = \frac{1}{m \cdot n} \sum_{y=0}^{n-1} \sum_{x=0}^{m-1} I(x, y) \quad (2)$$

and  $VAR$ , the variance gray value of input image, as calculated by the formula:

$$VAR = \frac{1}{m \cdot n} \sum_{y=0}^{n-1} \sum_{x=0}^{m-1} (I(x, y) - M)^2 \quad (3)$$

where  $I(x, y)$  is the original grey value of pixel  $(x, y)$ , and  $I'(x, y)$  is the normalized grey value of pixel  $(x, y)$ , and  $m$  and  $n$  are the width and height of the input image.

Normally the normalized image will have relatively low contrast level of gray value.

#### 2.1.3 The Summary of Input Image Processing Algorithm

- (1) Calculate the  $M$  (mean value) of input image using Equation 2.
- (2) Calculate the  $VAR$  (variance value) of input image using Equation 3.
- (3) For each pixel in the original input image, calculate the normalized gray value using Equation 1.
- (4) Output the new gray values as the normalized image.

## 2.2 Fuzzy Clustering

In order to select the fingerprint area and exclude the background area or the highly damaged area, in this project, we apply classic FCM algorithm on the input normalized image with pre-defined pixel feature vector. These features are also used by Bazen, A.M and Gerez, S.H [10] to build a linear neural network. The advantage of fuzzy clustering is that it does not need training data and experiment result shows that it does have the ability to separate the fingerprint area from the background area. In this part, there are two steps: Image Histogram Equation and Fuzzy Clustering.

Since our fuzzy clustering algorithm heavily relies on the local gray level change and low contrast image will influence the result of clustering, so we first use classic histogram equation to increase the contrast of those images before applying FCM algorithm on the normalized image.

The feature vector we used in our project contains three pixel features: Local Coherence, Local Mean and Local Variance. Local Coherence gives a measure of how well the gradients are pointing in the same direction within the pre-defined kernel ( $25 \times 25$ ) centered by current pixel. Since a fingerprint mainly consists of parallel line patterns, the local coherence in the fingerprint area will be considerable higher than that in the background area. The formula of Local Coherence is defined as follows:

$$Coh = \frac{\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}}{G_{xx} + G_{yy}} \quad G_{xx} = \sum_g G_x^2 \quad G_{yy} = \sum_g G_y^2 \quad (4)$$

$G_{xy}$  is the sum of the product of  $G_x$  and  $G_y$ .  $G_x$  and  $G_y$  represent local gradients on the horizontal orientation and the vertical orientation. In this project, we use Sobel Kernel ( $3 \times 3$ ) to calculate these gradients.  $\sum_g$  means that we use Gaussian kernel ( $25 \times 25$ ,  $\sigma = 6$ ) centered at pixel  $(x, y)$  to calculate the Local Coherence value of that pixel. By using Gaussian smooth kernel to get the local average features, it is proved to be very useful to reduce the influence of noise. We also use this way to calculate the Local Mean and Local Variance.

Since the local coherence in the fingerprint area will be considerable higher than that in the background area, using this characteristic we can easily tell which cluster center represents the fingerprint area and which one represents the background area.

## 2.3 Image Enhancement

In this part, we use modified anisotropic filter to enhance the ridge and furrow information in the fingerprint area based on the local orientation image. Image Enhancement module takes the normalized image as input and output the enhanced fingerprint image. It contains four steps: Local Orientation Estimation, Image Smoothing by Low Pass Filter, Image Enhancement by Anisotropic Filter and Image Histogram Equation.

## 2.4 Segmentation

During fingerprint enhancement, the noise information are suppressed and the ridges and furrows information enhanced. In order to get rid of the noise in binary image without influence the ridges and furrows information, information is introduced to the fusion based threshold algorithm to get the binary result of the enhanced image.

## 3 Experimental Results

Fingerprint images with size =  $360 \times 364$  and 8-bit gray level BMP pictures were used as test files. 120 files were tested. The fingerprint images were divided into 3 main categories: *Good Quality Images*, *Poor Quality Images*, and *Extremely Dense Ridge Images*.

The *Good Quality Images* had relatively clear ridge and furrow structure and in the fingerprint area and there are not many noises (for example smears, scars and other unwanted elements) For *Good Quality Images*, they can also be classified as two different kinds of images: *Energy Balanced Images* (ridge and furrow contains almost the same amount of energy) and *Dense Ridge Images*. In the former the algorithm maintains the balanced status of the current image and get excellent enhancement and segmentation result

The *Dense Ridge Image* are ones in which although the ridge-furrow information is clear, the ridges occupy more energy than the furrows. So in these images, the average gray levels are usually low and the furrows, which are in high intensity, can be barely seen. 9. The experiment results show that our algorithm can, not only clearly enhance and segment these images, but also can balance the energy distribution due to the information fusion based threshold segmentation

Normally *Poor Quality Images* contain many areas in which there are blurred ridge and furrows. Also most of the *Poor Quality Images* have large number of noise (such as smears and scars) in the fingerprint area. Again our algorithm shows that it can handle these situations well including selecting relatively good fingerprint area, re-build the ridge and furrow structures in blurred areas and extenuate the influence of noises. One limitation to the proposed algorithm is that it is still dependent on the quality of the input image. If the input image is very poor and fuzzy then the algorithm fails to give satisfactory results.

Table 1 below shows the results from a statistical analysis of the experiments performed. According to the statistical analysis data given in the Table 1, we know that our algorithm work well on 90.5% of all testing fingerprint images. And only get about 1.7% really bad segmentation results. We also learn from the statistic data that 88.7% of all testing fingerprint images got correct classification and 11.3% of them include some kinds of misclassification.

**Table 1: Statistical Analysis of the Performance of the Algorithm**

	Database 1		Database2		Database3		Database4	
Total	40		40		40		120	
	Num	Ratio	Num	Ratio	Num	Ratio	Num	Ratio
Excellent	16	40%	25	62.5%	19	48.6%	60	50%
Good	18	45%	14	35%	14	40%	46	40%
Normal	4	10%	1	2.5%	4	11.4%	9	7.8%
Bad	2	5%	0	0%	0	0%	2	1.7%
Misclassified	5	12.5%	5	12.5%	3	8.6%	13	11.3%

## 4 Conclusions

The paper proposes a system that performs fingerprint enhancement and segmentation, which is a pre-processing step for automatic fingerprint identification. A Fuzzy Clustering algorithm is used to separate the fingerprint area and background area. Further more, the fuzzy clustering algorithm can also get rid of those unclear or highly damaged area and keep the relatively good fingerprint area.

Combining with the fuzzy membership degree, a Local Orientation Image is used to represent the directions of ridges and furrows in the fingerprint area. By using low pass filter to smooth this Local orientation image, most of the noise, such as scars, smears and other flaws can be eliminated or weakened. Applying modified anisotropic filter on the fingerprint area, the filter will turn itself to the direction according to the local orientation and only enhance the information of ridges and furrows. So on the other hand, the modified anisotropic filter is able to suppress the influence of noise efficiently. Experimental results show some fundamental success in the use of the algorithm to pre-process fingerprints. However, some shortcomings of the algorithm were found (which are also common in all known algorithms). Poor quality images, the fuzzy clustering may select some unconnected areas as the fingerprint area.

## 5 References

- [1] Guoqiang Ma; Juan Liu; , "Fingerprint Image Segmentation Method Based on Gray Vision Parameter," Information Engineering (ICIE), 2010 WASE International Conference on , vol.1, no., pp.154-157, 14-15 Aug. 2010.
- [2] J. Y. Kang, C. L. Gong & W. J. Zhang. Fingerprint Image Segmentation Using Modified Fuzzy c-means Algorithm. J. Biomedical Science and Engineering Dec 2009:2. pp 656 - 660. Also available at <http://www.SciRProg/journal/jbise> [viewed March 3, 2011].
- [3] Greenberg, S., Aladjem, M. and Kogan, D., "Fingerprint Image Enhancement Using Filtering Techniques" (*Real-Time Imaging*), Vol.8, pp.227 – 236, 2002.
- [4] Lei Zhang; Mei Xie; , "Realization of a new-style fingerprint recognition system based on DSP," IT in Medicine and Education, 2008. ITME 2008. IEEE International Symposium on , vol., no., pp.1107-1111, 12-14 Dec. 2008.
- [5] F. González, O. Villegas, V. Sánchez, H. Domínguez. *Fingerprint Recognition Using Open Algorithms in Frequency and Spatial Domain*. Proceedings of 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference CERMA 2010, October 2010 Cuernavaca, Morelos, México, pp 469 - 474.
- [6] S. Jirachaweng, Z. Hou, W-Y Yau, V. Areekul. "Residual Orientation Modeling For Fingerprint Enhancement And Singular Point Detection." J. Pattern Recognition archive Volume 44 Issue 2, pp 431 – 442. Feb, 2011.
- [7] Almansa, A. and Lindeberg, T., "Fingerprint enhancement by shape adaptation of scale-space operators with automatic scale selection" (*IEEE Transactions on Image Processing*), Vol.9, No.12, pp.2027 – 2042, 2000.
- [8] Jiajia Lei; Hatem, H.; Long Zhou; Xinge You; Wang, P.S.P.; Duanquan Xu; , "Fingerprint enhancement based on non-separable wavelet," Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on , vol., no., pp.313-317, 7-9 July 2010.
- [9] C. Obimbo, W. Wang, J. Tian, G. Grewal. Fingerprint Enhancement and Segmentation. Proceedings of the Conference on Artificial Neural Networks in Engineering, St. Louis, MO, USA. November 5-8, 2006, pp 451 - 458.
- [10] P. Sutthiwichaiyorn. *Iterative Fingerprint Enhancement with Matched Filtering and Quality Diffusion in Spatial-Frequency Domain*. International Conference on Pattern Recognition. 1257 – 1260.
- [11] Bazen, A.M. and Gerez, S.H., "Segmentation of Fingerprint Images" (*ProRISC Workshop on Circuits, Systems and Signal Processing*), pp.276 – 280.
- [12] Hong, L., Wan, Y. and Jain, A., "Fingerprint Image Enhancement: Algorithm and performance evaluation" (*IEEE Transactions on Pattern Analysis and Machine Intelligence*), Vol.20, No.8, pp.777 – 789, 1998.
- [13] Hong, L., Jain, A., Pankanti, S., Bolle, R., "Fingerprint Enhancement" (*IEEE Workshop on the Application of Computer Vision*), pp.202 – 207.
- [14] Jianwei, Yang., Lifeng, Liu. and Tianzi, Jiang., "A Modified Gabor Filter Design Method for Fingerprint Image Enhancement" *Pattern Recognition Letters* v24:12, 2003, pp 1805 - 1817.
- [15] Kamei, T. and Mizoguchi, M., "Image filter design for fingerprint enhancement" (Proceedings International Symposium on Computer Vision), pp.109 – 114, 1995.
- [16] Shijun Zhao, Xiaowei Hao, Xiaodong Li, "Segmentation of Fingerprint Images Using Support Vector Machines," iita, vol. 2, pp.423-427, 2<sup>nd</sup> International Symposium on Intelligent Information Technology Application, 2008.

# Methods of Speeding Up Secret Computations With Insecure Auxiliary Computer

Yerzhan N. Seitkulov

Electrical and Computer Engineering Department,  
Binghamton University, Binghamton, NY, USA.

E-mail: yseitkul@binghamton.edu

Tel.: (607)7442316

**Abstract** - Currently, the problem of speeding up secret computations with the help of an auxiliary computer changed and was enriched by numerous problems of computational mathematics, where a solution means an approximate solution. The main goal of this paper is to demonstrate the different methods of computing approximate solutions of some equations with help of an auxiliary computer. To show methods, we chose the certain classes of algebraic and differential equations because in most cases modern computing problems are reduced to solving such systems of equations (differential equations, linear programming, etc.).

**Keywords:** Speeding up secret computation, SASC, secure outsourcing.

## 1 Introduction

The problem of speeding up secret computations with the help of an auxiliary (external) computer in the theory of information security first emerged in [1]. The idea of using auxiliary computers in solving problems with secret parameters is also considered and developed by many cryptographers [2-17]. The main results of these studies were obtained in RSA. Currently, the problems about an auxiliary computer changed and was enriched by numerous problems of computational mathematics, where a solution means an approximate solution. Such problems arise in the economy, military and other spheres. Note that many of today's computing problems require large computational resources and therefore they can only be solved on supercomputer or using the capabilities of the largest computing systems such as grid technology, etc.

Detailed description of the problem is contained in [1]. Briefly: secret computation with help of an auxiliary computer (server) is used when a client needs to execute a task but does not have the appropriate computation power to perform it. A problem is that the client may wish that some computation input be kept secret from the server. Moreover, the involved helpers (server) may be dishonest or corrupted and thus the task owner has to check that the returned result is correct.

The main goal of this paper is to demonstrate the different methods of computing approximate solutions of

some equations with help of auxiliary computer. To show methods, we chose the certain classes of algebraic and differential equations because in most cases modern computing problems are reduced to solving such systems of equations (differential equations, linear programming, etc.).

So according to generally accepted terminology Client means an entity who wishes to obtain an approximate solution to some problem with secret parameters. And Server means an insecure auxiliary computer (supercomputer, grid, etc.).

## 2 Methods of speeding up secret computations

When writing a protocol for each method, we try to follow the usual requirements for reliability protocol. These requirements are:

- I. Protection protocol from active attacks.
- II. The correctness of protocol. Meaning that Client calculates a simple task, and Server calculates a difficult task.
- III. Practicality.

### 2.1 Linear equations with a secret right-hand side

**Target LE.** Let  $H$  be Banach space and  $L$  be linear operator with domain in  $D(L) \subseteq H$ . Client needs to obtain a solution of the equation  $Lx = f$ , where right-hand side  $f$  and solution  $x$  are secrets.

**Protocol LE.**

- I. Client takes a random element  $w \in D(L)$  and calculates  $Lw$ . Next, Client calculates a difference between  $f$  and  $Lw$ :

$$f - Lw = g.$$

Now Client sends the equation  $Ly = g$  to Server, and keeps the element  $w$  as secret.

- II. Server finds a solution of the equation

$$Ly = g,$$

and returns the approximate solution  $y$  to Client.

**III.** Client finds an approximate solution of

$$Ly = f$$

by the formula  $x = y + w$ .

**Example LE.** Let Client needs to obtain a solution of the equation

$$\begin{cases} x'' + q(t)x = f(t), \\ x(0) = 0, x(1) = 0 \end{cases} \quad (1)$$

where right-hand side  $f(t)$  and a solution  $x(t)$  are secrets.

**Protocol Example LE.**

**I.** Client takes any twice continuously differentiable function  $r(t)$ . Let

$$w(t) \equiv t(t-1)r(t).$$

Now Client calculates

$$w''(t) + q(t)w(t) \equiv g_1(t).$$

And let

$$f(t) - g_1(t) \equiv g(t)$$

Now Client sends the function  $g(t)$  to Server.

**II.** Server finds a solution of the equation:

$$\begin{cases} y''(t) + q(t)y = g(t) \\ y(0) = 0, y(1) = 0, \end{cases}$$

and returns the approximate solution

$$y = (y[0]), \dots, y[n-1]$$

to Client, where

$$y[i] = y\left(\frac{i}{n-1}\right), i = 0, \dots, n-1.$$

**III.** Client obtains a solution of (1) by the algorithm:

*for*( $j = 0; j < n; j++$ )

$$x[j] = y[j] + w[j];$$

## 2.2 Boundary value problem with the secret boundary conditions

**Target BVP.** Let Client needs to obtain solution of the equation

$$\begin{cases} y''(t) + q(t)y(t) = f(t) \\ y(0) = s_1, y(1) = s_2, \end{cases} \quad (2)$$

where the boundary conditions  $s_1$  and  $s_2$  are secrets. The equation  $y''(t) + q(t)y(t) = f(t)$  is not secret.

**Protocol BVP.**

**I.** Client sends the equation  $y''(t) + q(t)y(t) = f(t)$  to Server.

**II.** Server calculates any fundamental system of solutions  $y_1(t)$  and  $y_2(t)$  of the homogeneous equation:

$$y''(t) + q(t)y(t) = 0,$$

and finds any particular solution  $y_3(t)$  of the equation

$$y''(t) + q(t)y = f(t).$$

And returns approximate solution

$$y_k = (y_k[0]), \dots, y_k[n-1], k = 1, 2, 3.$$

to Client, where

$$y_k[i] = y_k\left(\frac{i}{n-1}\right), i = 0, \dots, n-1.$$

**III.** Client finds the numbers  $c_1$  and  $c_2$  from the system of algebraic equations:

$$\begin{cases} c_1 y_1[0] + c_2 y_2[0] + y_3[0] = s_1, \\ c_1 y_1[n-1] + c_2 y_2[n-1] + y_3[n-1] = s_2 \end{cases}$$

and then finds approximate solution

$$y = (y[0]), \dots, y[n-1]$$

of the equation (2) by the algorithm:

*for*( $j = 0; j < n; j++$ )

$$y[j] = c_1 y_1[j] + c_2 y_2[j] + y_3[j];$$

## 2.3 Nonlinear equations reducible to a linear form

Sometimes nonlinear equations can be reduced to a linear form, for example, by changing variables. There are many such examples. Therefore, if a nonlinear equation is reduced to a linear equation, it can solve the above-mentioned methods.

## 2.4 Initial value problem with a secret parameter

Now we will show how we can solve the nonlinear equations without the possibility of reducing to a linear equation.

**Target IVP.** Let Client needs to obtain a value  $y(1)$  of the solution  $y(t)$  of the nonlinear initial value problem:

$$\begin{cases} -y'' + y^3 = at^3 \\ y(0) = y'(0) = 0, \end{cases}$$

where the parameter  $a$  is secret.

**Protocol IVP.**

**I.** Client takes a random number  $\beta \neq 0$  and computes the product  $a\beta^6 \equiv c$ . Next, Client picks a random number  $m > 2\beta^{-1}$ , and sends the numbers  $c$  and  $m$  to Server. Client keeps the  $\beta$  as secret.

**II.** Server finds a solution on interval  $[0, m]$  of the equation:

$$\begin{cases} -v''(\eta) + v^3(\eta) = c\eta^3 \\ v(0) = v'(0) = 0, \end{cases}$$

and returns the solution  $v(\eta)$  to Client.

**III.** Client calculates  $y(1)$  by the formula:

$$y(1) = \beta^{-1}v(\beta^{-1}).$$

### 2.5 Calculating the value of the function of a secret argument

**Target VFSA.** Let Client needs to obtain the value of the holomorphic function  $f(z)$  at the secret argument  $z = a$ . The function  $f(z)$  is not secret,  $a$  is a secret argument.

We need the next theorem from the complex analysis: Suppose  $U$  is an open subset of the complex plane  $C$ ,  $f : U \rightarrow C$  is a holomorphic function and the closed disk  $D = \{z : |z - z_0| \leq r\}$  is completely contained in  $U$ . Let  $\partial D$  be the circle forming the boundary of  $D$ . Then for every  $a$  in the interior of  $D$ :

$$f(a) = \frac{1}{2\pi i} \oint_{\partial D} \frac{f(z)}{z - a} dz \tag{3}$$

where the contour integral is taken counter-clockwise.

Now the formula (3) can be approximately rewritten in terms of integral sum:

$$f(a) \approx \frac{1}{2\pi i} \sum_{k=1}^n \frac{f(z_k)}{z_k - a} (z_{k+1} - z_k) \tag{4}$$

where  $z_1, \dots, z_n$  denote numbers chosen on  $\partial D$ , and  $z_1 = z_{n+1}$ . Thus, we see that the value of a secret argument can be approximately calculated by the formula (4).

#### Protocol VFSA.

**I.** Client chooses the numbers  $z_1, \dots, z_n$  on the boundary of  $\partial D$  and transmits them to Server.

**II.** Server calculates  $f(z_k)$  for each  $k = 1, \dots, n$ , and sends the results to Client.

**III.** Client finds the approximate value of the function  $f$  at the secret argument  $a$  by the formula (4).

### 2.6 Nonlinear equation with a secret right-hand side

**Target NLE.** Let Client needs to obtain a solution of the nonlinear equation:

$$\begin{cases} -y'' + y^3 = f(t) \\ y(0) = y(1) = 0, \end{cases} \tag{5}$$

where the function  $f$  is secret.

To solve this problem, we can use the method 2.5 described above. So, let  $F(t, \lambda)$  be a function of two variables:

$$F(t, \lambda), 0 \leq t \leq 1, \lambda \in \Omega,$$

where  $\Omega$  is an open area in the complex plane. Further, suppose that:

1)  $F(t, \lambda)$  is holomorphic in  $\Omega$  for each  $t \in [0, 1]$ ;

2)  $F(t, \lambda_0) = f(t)$  for some  $\lambda_0$ .

Now we form the equation:

$$\begin{cases} -y'' + y^3 = F(t, \lambda) \\ y(0) = y(1) = 0. \end{cases} \tag{6}$$

Under certain conditions on  $F(t, \lambda)$ , the solution  $y(t, \lambda)$  of the system (6) as a function of  $\lambda$  is an analytic function. Therefore, we have the following formula:

$$y(t) \equiv y(t, \lambda_0) = \frac{1}{2\pi i} \oint_{\partial D} \frac{y(t, \lambda)}{\lambda - \lambda_0} d\lambda.$$

#### Protocol NLE.

**I.** Client chooses the numbers  $\lambda_1, \dots, \lambda_n$  on the boundary of  $\partial D$ ,  $D = \{z : |z - z_0| \leq r\} \subset \Omega$  and transmits them to Server.

**II.** Server finds a solution  $y_j(t, \lambda_j)$  of the equation

$$\begin{cases} -y_j'' + y_j^3 = F(t, \lambda_j) \\ y_j(0) = y_j(1) = 0, \end{cases}$$

for each  $j = 1, \dots, n$ , and sends the results to Client.

**III.** Client finds the approximate solution of the equation (5) by the formula:

$$\begin{aligned} y(t) \equiv y(t, \lambda_0) &= \frac{1}{2\pi i} \oint_{\partial D} \frac{y(t, \lambda)}{\lambda - \lambda_0} d\lambda \approx \\ &\approx \frac{1}{2\pi i} \sum_{k=1}^n \frac{y_k(t, \lambda_k)}{\lambda_k - \lambda_0} (\lambda_{k+1} - \lambda_k) \end{aligned}$$

where  $\lambda_{n+1} = \lambda_1$ .

### 3 Conclusions

Note that all methods have been shown in the conceptual level. It can be seen that all the methods satisfy usual requirements of reliability of cryptographic protocols. Moreover, the secret parameters in all methods are protected 100 percent, and methods of cryptanalysis are not applicable. Methods 2.1 – 2.4 are very practical, but the methods 2.5 and 2.6 is not quite so easy in practice (because  $n$  is a very large number). Nevertheless, the belief is that the methods for nonlinear problems will find their use in the future.

Further, we note that Client in all methods can check the results of calculations obtained from Server, except method 2.5. Therefore, a passive attack for method 2.5 is possible. However, if we assume that Client knows the inverse function  $f^{-1}$ , then Client can check the result obtained from Server in method 2.5.

### 4 References

- [1] T. Matsumoto, K. Kato and H. Imai. "Speeding up secret computations with insecure auxiliary devices." In S. Goldwasser (Ed.) *Advances in Cryptology - Crypto 88*, LNCS 403, Springer-Verlag, pp. 497-506, 1990.
- [2] S. M. Hong, J. B. Shin, H. Lee-Kwang and H. Yoon. "A New Approach to Server-Aided Secret Computation", *Proceedings of ICISC98*, Seoul, Korea, pp. 33-45, December, 1998.
- [3] T. Cao, X. Mao, and D. Lin. "Security analysis of a server-aided rsa key generation protocol". In K. Chen, R. H. Deng, X. Lai, and J. Zhou, editors, *ISPEC*, volume 3903 of *Lecture Notes in Computer Science*, pp. 314-320. Springer, 2006.
- [4] P. Pfitzmann, M. Waidner. "Attacks on Protocols for Server-Aided RSA Computation". In: R. Rueppel (Ed.) *Advances in Cryptology – Eurocrypt 92*, LNCS 658, Springer-Verlag 1993, 153-162
- [5] A. Ernvall, K. Nyberg. "On Server-Aided Computation for RSA Protocols with Private Key Splitting". In S. Knapskog, editor, *Proceedings of Nordsec 2003*. Department of Telematics, NTNU, 2003.
- [6] P. Beguin and J.J. Quisquater. "Fast server-aided RSA signatures secure against active attacks", in *Crypto'95*, pp. 57-69, 1995.
- [7] C.H. Lim and P.J. Lee, "Security and performance of server-aided RSA computation protocols", in *Crypto'95*, pp. 70-83, 1995.
- [8] J. Burns and C.J. Mitchell, "Parameter selection for server-aided RSA computation schemes", *IEEE Trans. on Computers*, Vol. 43, No. 2, pp. 163-174, 1994.
- [9] D. Boneh, N. Modadugu, and M. Kim. "Generating rsa keys on a hand-held using an untrusted server". In *INDOCRYPT 00: Proceedings of the First International Conference on Progress in Cryptology*, pp. 271-282, London, UK, Springer-Verlag, 2000.
- [10] J. Burns and C. J. Mitchell. "Parameter selection for server-aided rsa computation schemes". *IEEE Trans. Comput.*, 43(2):163-174, 1994.
- [11] S. Kawamura and A. Shimbo, "Fast server-aided secret computation protocols for modular exponentiation", *IEEE JSAC*, Vol. 11, No. 5, pp. 778-784, 1993.
- [12] R. Akimana, O. Markowitch, and Y. Roggeman. "Grids confidential outsourcing of string matching". *The 6th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems*, 2007.
- [13] R. Akimana, O. Markowitch, and Y. Roggeman. "Secure outsourcing of dna sequences comparisons in a grid environment".
- [14] Atallah, Pantazopoulos, Rice, and Spafford. "Secure outsourcing of scientific computations". In *Advances in Computers*, ed. by Marshall C. Yovits, Academic Press, volume 54. 2001.
- [15] Moez Ben MBarka. "Secure computation outsourcing"; <http://www.fisoft1.com/sources/outsourcing.pdf>
- [16] Y. Chen, R. Safavi-Naini, J. Baek, and X. Chen. "Server-aided rsa key generation against collusion attack". In M. Burmester and A. Yasinsac, editors, *MADNES*, volume 4074 of *Lecture Notes in Computer Science*, pages 27–37. Springer, 2005.
- [17] M. Dijk, D. Clarke, B. Gassend, G. E. Suh, and S. Devadas. "Speeding up exponentiation using an untrusted computational resource". *Des. Codes Cryptography*, 39(2):253–273, 2006.

# Implementation and Applications of a Fingerprint Encoding System

I-Fu Lin and Tzong-An Su

Dept. of Information Engineering and Computer Science  
Feng Chia University  
Taichung Taiwan  
tasu@fcu.edu.tw

**Abstract-** For the fingerprint verification system, preventing from fake fingerprint attacks is a very important security issue. The fingerprint is one of the most common biometrics used for security today. Most research works focus on the living finger identification technology. But the cost of living fingerprint verification system is still higher than that of the traditional one. In our research, we utilized a sequential verification method called the sequential fingerprint verification to enhance the resistance of fake fingerprint attacks on the traditional fingerprint verification system. We can also extend this method to use fingerprints from a group of people to control the access to secret information. The experiment result showed that our method can delay the break-in from attacks of fake fingerprints. Some new applications based on our method are also discussed.

## I. INTRODUCTION

The use of fingerprints in various applications has been seen for thousands of years. People use fingerprints for signatures on art works, to authenticate a document, for criminal identification and most recently, for access control. Most of these applications involve so called fingerprint verification in which input fingerprints from some person are used to match with fingerprints pre-stored in a database.

Among those applications, using fingerprints in access control to ensure the system or information security is the most popular one in recent years, for example, many PCs have been installed fingerprint reader to verify authorized users. Lots of researches have been focused on supporting this type of application. Previous research in fingerprint verification area can be divided into fingerprint recognition and quality estimation [1, 2, 3, 4, 5], security issues on fingerprint verification system [6, 7, 8, 9, 10], and implementations of commercial products and applications [12].

In this paper, we will discuss the security issues on fingerprint verification systems. Generally, a fingerprint verification system can be divided into two parts: front-end and back-end [6]. On the back-end side, the security issues could be seen as general information security issues, such as operation system security, database security and communication security. Researchers have proposed various security technologies to work with the fingerprint verification system on the back-end side to protect the fingerprint data, such as image encryption [11]. In [12], the authors distributed the fingerprint minutiae into fingerprint verification system

and the smart card. This method could prevent the whole fingerprint features from stolen at the back-end side. Thus, our focus will be on the front-end side.

On the front-end side, the most important security issue is how to prevent the system from fake fingerprint replica attacks. [7] introduced different types of fingerprint readers, including optical, capacitive, ultrasonic and thermal sensors. [6] showed that different types of fingerprint readers had their own drawbacks. The authors indicated that there was 80% success rate by using fake fingerprint replicas to pass the fingerprint verification on the traditional optical fingerprint reader. Although the capacitive reader could filter off some isolative fingerprint replicas, [8] pointed out that attackers could use gelatin fingerprint replicas with moisture and resistance characteristics similar to a real human finger to fool the capacitive reader. [9, 10] showed the approach to make the fingerprint replicas and the conclusion is that it is not difficult. With all the discussion above, we should be able to realize the natural vulnerabilities of traditional fingerprint readers.

The design of traditional fingerprint reader mostly uses 1-1 or 1-N approach, i.e., one input fingerprint matches another stored fingerprint or matches among several stored fingerprints. This design feature actually adds even more vulnerability to the replica attacks. Some people suggested use multiple input fingerprints to match with stored ones in order or without order. Unfortunately, we haven't seen any real implementation due to the reader's simple architecture and weak capability including limited memory size.. Also, to change the "key" fingerprint normally involves a re-input of another fingerprint. It is both time-consuming and confusing.

Although the traditional fingerprint reader is subject to the issues stated above, it still enjoys high market acceptance rate. One of the reasons is due to its low manufacturing cost. The other reason is that even the high end and expensive biometrically enhanced reader can not completely eliminate the replica attack. Therefore, how to reduce or even eliminate the replica attack and add more power to the traditional fingerprint reader becomes an important work and deserves paying more attention.

## II. METHODOLOGY

In our research, we design and implement a sequential (match with order) multiple fingerprint verification reader based on the following requirements:

- The number of input fingerprints used in the matching process is determined by the user.
- The set of fingerprints used as input is determined by the user.
- The order of fingerprints used in matching process is determined by the user.
- The reader should allow fingerprints from multiple persons as an input in a matching process.
- The reader should allow users to change their secret finger sequences dynamically and easily.

To fulfill these requirements, we use a mapping scheme between fingerprints and a set of codes which are symbols. We associate each registered fingerprint with a unique code, i.e., encoding all the registered fingerprints. And then users can use this encoding scheme to create his secret finger sequence. The following describes details of these two steps:

*A. Fingerprint Encoding*

The simplest method to encode fingerprints is to map different fingerprints to different symbols like numerals or alphabets. In Figure1, we see that one user has enrolled his five different fingerprints in the reader system. There are two kinds of associated codes for this example in Figure1. In the simplest case, we can associate a single digit with a fingerprint and each digit can only be associated with one fingerprint. The other case is that we can associate any character in a set of characters to a fingerprint. The restriction is that the intersection of any two such mapping sets must be empty. That is, any character in the mapping set of a fingerprint should be able to uniquely identify the specific fingerprint.

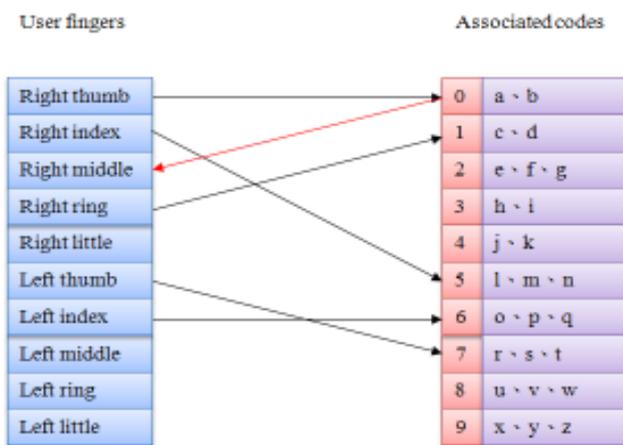


Figure1. Fingerprint Encoding of an Individual User

As an example, consider Figure 1. We can associate right thumb with digit ‘0’. On the other hand, we can associate it with the set of character ‘a’ and ‘b’. In the first case, the right

thumb fingerprint could be represented as the numeral ‘0’. In the second case, it can be represented as the alphabet ‘a’ or ‘b’.

As the system requirements state, our reader should allow fingerprints from multiple persons to be registered. To handle this requirement, after multiple persons registering their fingerprints, each person should follow the method described above to establish his own mapping between his fingerprints and symbol sets. Note that users do not need to register all their ten fingerprints. Also, the mappings of two persons are not related, e.g., they even can have the same mapping although it is not recommended.

*B. Creating the secret fingerprint sequence*

After encoding fingerprints appropriately, users could begin to create their secret fingerprint sequence. We would divide this step into two cases based on verifications to be done on single person or on a group of persons.

1) *Verifications involved only one single person:* To create a secret fingerprint sequence, a user can choose some from those registered fingerprints and input to the reader the corresponding mapping codes. For the example in Figure 1, if the user inputs the code sequence as ‘1751’, the system would sequentially verify the user’s right ring, left thumb, right index and right ring fingerprints. For another example, if the user sets sequential codes as ‘lab17’ the system would sequentially verify the user’s right index, right thumb, right thumb, right ring and left thumb fingerprints. In our design, we allow users to create multiple secret fingerprint sequences to avoid situations such as finger injured.

2) *Verifications Based on a Group of Persons:* Group sequential fingerprint verification allows using several persons’ fingerprints together for verifications to protect sensitive information. Since the verification involves all members in the group, for each code in the sequence, we need to have a mechanism to tell which person it belongs. The way we do it is to give each member an ID and associates each code in the sequence with the ID of some member in the group.

For example, if there are three members in the group, we can assign their user IDs as ‘A’, ‘B’ and ‘C’ respectively. Based on the mapping in Figure 1, if the group sequential code is ‘5A1B6B0C5A7A6B7B1C’ the system would sequentially verify A’s right index fingerprint, B’s right ring fingerprint, B’s left index fingerprint, C’s right thumb fingerprint, A’s right index fingerprint, A’s left thumb fingerprint, B’s left index fingerprint, B’s left thumb fingerprint, C’s right ring fingerprint. By the way, if the members wanted to change their fingerprints in the group verification, they could simply change their associated codes.

IV. IMPLEMENTATION

In this section, we introduce the design and implementation of the system. We introduce the devices used first.

*A. Equipment*

To prove that our method can apply to most of fingerprint verification systems, we chose a traditional fingerprint

verification module, which contained some basic functions on fingerprint verification system.



Fig.2 the A04-WM100 optical fingerprint developing module

1) *Hardware and middleware units:* We chose the A04-WM100 fingerprint developing module with the optical reader, which was developed by the East Wind Technologies in Taiwan, as shown in Fig.2. This module contains 32 bit DSP (ADSP BF531) 396 MHz processor, the Dual RS232 port communication interface, and a 24-pin digital signal input from CMOS sensor for the fingerprint interface. The supporting are 9600, 19200, 38400, 57600 and 115200 bps for RS232/RS485, and the power consumption was 5V D.C., 220mA, when it is operating with the optical sensor. The matching Speed for 1-to-1 verification is 0.5 second, and the memory size can fit 500 users' data. This module can be upgraded by applying middleware updates to extend the environment of software development. To achieve our objective of low cost implementation, we did not change the structure of the hardware and the middleware.

2) *Software and the packet structure:* The A04-WM100 fingerprint developing module uses Visual Basic and Visual C languages in its development; therefore, we chose Visual Basic as our implementation language. There is one exemplification program associated with the module, which included some basic functions such as the enrollment, verification, user data management, image display, image saving and other parameter setting. Some of the codes were written in the dynamic-link library (DLL) files as to protect the kernel technology.

Like other embedded systems, this module also needs the fixed packets to communicate with the main board. The packet structure shown in Fig.3, contains the header, data body and a checksum. The header was used to identify the packet, the data body would take the data, and the checksum is used to check if the packet is legal before the transmission.

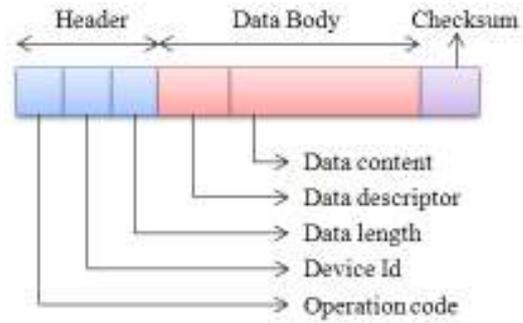


Fig.3 the packet structure for A04-WM100 module

*B. Data structure:*

For the single user case, the user ID, password and role were used for individual identification and verification. The role category includes administrator and regular user. The fingerprint encoding data is used to record the fingerprint encoding information mentioned in section III, the sequential codes are used to do the sequential fingerprint verification, and the group associated data contains the groups in which the user participate.

For the group case, the group ID is used for identification of the group. The group password is used to verify the members in the group. To join the group, qualified users have to get the group password from the administrator. The group data also record the total number of members in the group and the total number of fingerprints. The last piece of data is the group sequential code. Based on our design, the group sequential code contains members' IDs and their associated fingerprint codes. The associated codes help members easily change their fingerprints.

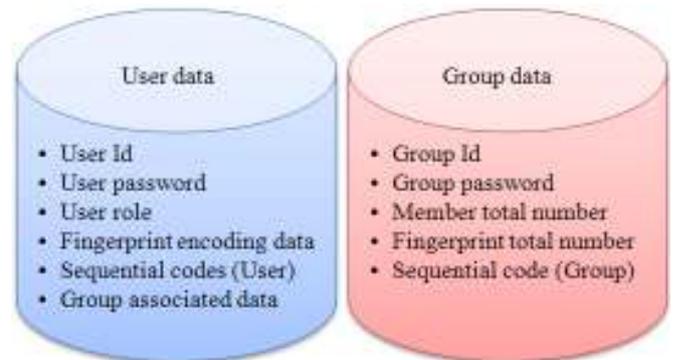


Fig.4 the structure of single user data and group data

*C. System design:*

The followings are the functionalities we implemented on the A04-WM100 hardware to create a low cost fingerprint verification system.

1) *Enrollment for the single user case:* This functionality supports users to enroll their information to the system. Information can be useful fingerprints, their user IDs, user passwords, fingerprint names and their associated codes. We

use the reserve fingerprint space in the hardware to store such information. Like most fingerprint verification systems, this module only has space for three fingerprints. In other words, each user could only register 3 fingerprints.

2) *Data viewing for the single user case*: This function allows users to view their own individual data, when they forgot them. The user data is protected by the user password, so no one can see these personal data except the user. The data included user ID, fingerprint encoding data and the sequential code.

3) *Verification for the single user case*: When the user data is established, the user can prepare to do the sequential verification. To prevent attackers to collect and analyze the error information, such as the length of the sequential code, we designed the system in such a way that it will ask users whether they want to continue to match the next fingerprint or not after matching each fingerprint in the sequential verification process. When users think that they have matched all fingerprints, they need to tell the system that the verification process is done. The threshold of the sequential verification i.e., the number of fingerprints used in the verification can vary depended on the number of fingerprint storage space that the hardware provides.

4) *Enrollment for the single group case*: This function provides two services, the enrollment of group data and the participation of group members. First, the system will request the user to fill in the group ID and the group password, then the system will check whether the group exists or not. If the group exists, the system will begin the process of the member participation and check the group password. After confirming that the user has the permission to join this group, the system will check whether the user and the data are legal or not. The user should fill in the user ID, user password, user role, the group fingerprint number, the participated fingerprint and its associated code. Only after passing all verifications, the user can take part in the group. Due to the hardware resource limitation, the maximal length of our group sequential code is 9. If you get more resources, you could extend the length.

5) *Data viewing for the group members*: This function is similar to user data viewing. If a user wants to view the group data, he has to fill in the group password. Only legal group members and the administrator can see the group data. The data included group ID, the number of total members, the number of total participated fingerprints, and the group sequential code.

6) *Verification for the group and the authorization*: The principle of the group sequential verification was similar to the single user verification. Because the length of group sequential code is longer than that of the single user, we can use the success number of fingerprint matching to grant authorization.

### III. ADVANTAGES

We discuss the advantages of our design methodology below.

1) *To reduce the consumption of storages*: Many fingerprint verification system actually store fingerprints for each secret fingerprint sequence. In some situations, such as finger injury,

we have to have the backup fingerprint sequence. Thus, physically storing each fingerprint sequence will consume lots of memory space especially in the group verification case. In our design, we only store one actual copy of each fingerprint and any fingerprint sequence can be represented as a sequence of mapping codes. This would greatly reduce the consumption of memory.

2) *To make the modification or the creation of the secret fingerprint sequence easier*: Without using a coding scheme to represent fingerprints internally such as those traditional verification readers, to create or modify the secret fingerprint sequence has to re-input those fingerprints. This is a very time-consuming and cumbersome work. In our design, after all fingerprints are registered, to create or modify fingerprint sequence is just a setup or change of some internal codes. This works even better in the group verification case e.g., when a member of the group can not use some of his fingers and the fingerprint sequence has to be changed.

3) *To reduce the success rate of fingerprint replicas attacks*: Obviously, attackers have to know both the mapping scheme and sequential codes before they can use prepared fingerprint replicas to pass the verification. It is much more difficult than the traditional verification readers. We suggest storing these two pieces of data separately to reduce the risks of been stolen. Even in the circumstance that one of the two pieces of data has been stolen, we still can easily change the other piece of information to stop the attacking process and gain more time to fix the system vulnerability.

4) *To create more applications*: Based on our method, we can create more applications. One example is the so called "challenge-response" application. Say there is an administrator of a lab in a university. To control the access to the lab, only those authorized personnel are allowed to enter the lab. The administrator could setup a table containing symbols to be mapped by fingerprints. He then can ask all authorized persons to register their fingerprints to the security system by using the table he created. When every person's fingerprint mapping is created completely, the administrator could set a sequential code on the system that everyone could see. The authorized persons could pass the verification because they know the fingerprint mapping. For unauthorized persons, they still couldn't pass the verification even they could see the sequential code. Also, the administrator could change the sequential code as often as he needs to without notifying anyone.

### IV. PERFORMANCE COMPARISON

In this section, we compare the traditional fingerprint verification method with our approach under the fingerprint replica attacks. We assumed that attackers already got the entire fingerprint replicas. For the simplest case where most of the PCs allow using just one registered fingerprint to login, there is no protection at all because the attacker can just try ten times in the worse case to gain access to the PC. Some PCs using combination of two fingerprints are also not able to provide enough protection because the attacker can try only 45 combinations of two fingerprints from ten fingerprints to crack

the login protection. Assume the verification reader matches the pattern without order.

Next, we discuss the expression (1), which was used to calculate the worst case attacking time for the sequential fingerprint verification. Let  $S$  be the number of fingerprints which are used to construct the sequential codes.  $V_t$  is the time to perform one verification and  $m$  is the number of persons in the group involved in the verification process. Expression (1) shows the total attacking time.

$$(10m)^S S V_t. \quad (1)$$

In the following, we estimate the attacking time by varying the  $S$  value, i.e., the number of fingerprints involved in the verification process. We would separate the discussion into the single user case and the group case.

1) *Estimation of the single user verification:* We set  $S$  value from 1 to 10,  $m$  is 1, and the  $V_t$  is 0.5 second which we measure from our implementation. The estimation results are shown in Figure 2.

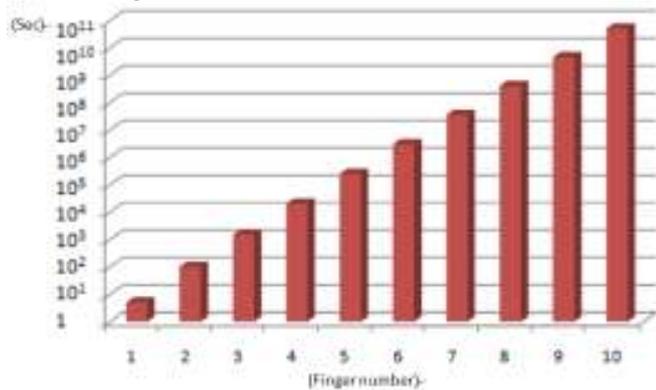


Figure 2. Estimation results for single user

Figure 2 shows the total attacking time of the sequential method would extend when the length of sequential code increases. By using 4 fingerprints, the attacking time is about 5.5 hours, and by using 8 fingerprints in line with sequential code the total attacking time would extend to about 12.6 years.

2) *Estimation for the group verification:* In this case, we would use the same values for the parameters as in case 1. The number of participated members  $m$  is set to 3. The results are shown in Figure 3. If we choose the number of fingerprints as 9, the attacking time would be extended to about 2808647.2 years. By the way, if we increased the number of participated members, the attacking time would be higher.

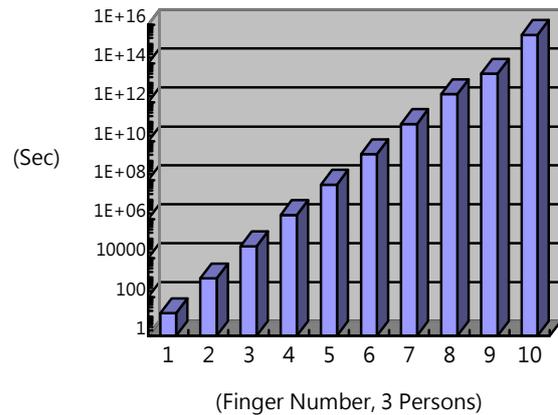


Figure 3. Estimation results for the group

With these estimation results, we could see that the sequential fingerprint verification method could extend the attacking time. In other word, we could gain more time to detect the attacks and fix the system's vulnerabilities.

## V. CONCLUSIONS

In this paper, we present a design to improve the vulnerability of traditional fingerprint verification systems against fingerprint replica attacks. This design can raise the protection capability of a cheap traditional fingerprint verification system to the comparable level of a much more expensive biometric system. It also provides the feature of easy creation or modification of the secret fingerprint sequences for users to quickly change their secret sequence to prevent or stop attacks. This design can be totally developed in the software, so we can reduce the research cost on the hardware and middleware. Furthermore, this method will not be limited by the hardware, such as the types or fingerprint sensors. Therefore it can be applied to any verification hardware and any place where entity authentications are required.

## REFERENCES

- [1] Jain, A.K.; Yi Chen; Demirkus, M., "Pores and Ridges: High-Resolution Fingerprint Matching Using Level 3 Features", *Transactions on IEEE Pattern Analysis and Machine Intelligence* Volume 29, Issue 1, Jan. 2007.
- [2] Wang Yuan; Yao Lixiu; Zhou Fuqiang, "A Real Time Fingerprint Recognition System Based On Novel Fingerprint Matching Strategy", *Electronic Measurement and Instruments, 2007. ICEMI '07. 8th Conference on International Aug. 16 2007-July 18 2007*.
- [3] Alonso-Fernandez, F.; Fierrez, J.; Ortega-Garcia, J.; Gonzalez-Rodriguez, J.; Fronthaler, H.; Kollreider, K.; Bigun, J., "A Comparative Study of Fingerprint Image-Quality Estimation Methods", *Information Forensics and Security, Transactions on IEEE* Volume 2, Issue 4, Dec. 2007.
- [4] Modi, S.K.; Elliott, S.J.; Whetstone, J.; Hakil Kim, "Impact of Age Groups on Fingerprint Recognition Performance", *Automatic Identification Advanced Technologies, 2007 Workshop on IEEE 7-8 June 2007*.
- [5] Kukulka, E.; Elliott, S.; Hakil Kim; San Martin, C., "The Impact of Fingerprint Force on Image Quality and the Detection of Minutiae",

- Electro/Information Technology, 2007 Conference on IEEE International* 17-20 May 2007.
- [6] Sean Palka, Booz Allen Hamilton, "Fingerprint Readers: Vulnerabilities to Front- and Back- end Attacks", *Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. Conference on First IEEE International*, 27-29 Sept. 2007.
  - [7] Lin Cheng Sung, Biometric Technology, *Iridian*.
  - [8] Joe Grand, "Can You Really Trust Hardware? Exploring Security Problems in Hardware Devices", *The Black Hat Briefings 2005*, Grand Idea Studio, Inc.
  - [9] A. Wiche, T. Söndrol, O. K. Olsen, F. Skarderud, "Attacking Fingerprint Sensors", *Gjøvik University College*, 15th December 2004.
  - [10] T. Matsumoto, H. Matsumoto, K. Yamada, S. Hoshino, Impact of Artificial "Gummy" Fingers on Fingerprint Systems, *prepared for Proceedings of SPE Vol. #4677, Optical Security and Counterfeit Deterrence Techniques IV*, Thursday-Friday 24-25, January 2002.
  - [11] M. Kiviharju, "Hacking Fingerprint Scanners, Or Why Microsoft Fingerprint Scanner Is Not a Security Feature", Black Hat Europe, 2006.
  - [12] Hanna Choi; Sungju Lee; Daesung Moon; Yongwha Chung; Sungbum Pan, "Secret Distribution for Secure Fingerprint Verification", *Convergence and Hybrid Information Technology, 2008. Conference on ICHIT '08. International* 28-30 Aug. 2008.
  - [13] Tipwai, P.; Thipaksorn, P.; Madarasmi, S., "A Fingerprint Matching Scheme Based on Gradient Difference Compatibility", *Computational Intelligence in Image and Signal Processing, 2007. CIISP 2007. Symposium on IEEE* 1-5 April 2007.
  - [14] *Atmel Corporation*, "Thermal Fingerprint Sensor with 0.4 mm x 14 mm (0.02" x 0.55") Sensing Area and Digital Output (On-chip ADC)", 2007.
  - [15] *East Wind Technologies, Inc.*, "Hardware Specification (A04-WM100 optical fingerprint developing module)".
  - [16] *East Wind Technologies, Inc.*, "User Manual (A04-WM100 optical fingerprint developing module)".
  - [17] *East Wind Technologies, Inc.*, "Programmer Guide (A04-WM100 optical fingerprint developing module)".

# Automatic Mission-Critical Data Discovery Based On Content: Experimental Results

Jonathan White  
University of Arkansas  
Fayetteville, AR  
jlw09@uark.edu  
white@harding.edu

Brajendra Panda  
University of Arkansas  
Fayetteville, AR  
bpanda@uark.edu

**Abstract**—In this work, we present results from a system that was designed to automatically identify critical data based upon content. This identification data can then be used for computer security purposes: known critical data can have extra safeguards extended to it and users that are attempting to access critical data illicitly can also be better identified. The process works by using a SVM-based filtering system that has been trained with expert knowledge. The system itself was described in a previous work; the contribution of this work is in the many experimental results which are presented. The results show that the proposed system will be a valuable tool in the computer security engineer's toolkit.

## I. INTRODUCTION

Information systems typically contain a shifting collection of data and as the amount of data that is stored increases, it is not a valid assumption that those called on to protect this data will be familiar with what data is and is not critical. Intrusion detection systems are designed to protect wide swaths of data as opposed to a focused approach and these factors lead us to propose a new method that can be used to automatically identify data items that are critical. In our research, we have identified two approaches to this process. One involves identifying critical data items by content and context, which is explored in this work. The second method involves examining the usage of the data system and detecting which data items influence a large number of other data items and at what frequency this influence occurs. The latter method has already been designed and tested. To the best of our knowledge, little other publically available previous work has been done in this area.

The advantage of this approach is that it is a flexible and resource efficient technique and it solves the problem of identifying critical data items while also avoiding the pitfalls of using only static lists of critical data. Also, as data systems change constantly, the proposed algorithm can be invoked to react to this dynamically changing environment. By identifying the critical data items, system administrators can deploy tighter and more focused access policies, more detailed monitoring systems, and other focused insider detection tools, such as honeypots and honeynets. Also, once this data is identified, it can be used for detect a potential insider threat, which is the primary thrust of this work.

The rest of the paper is organized as follows. Section 2 identifies related work. Section 3 details our approach, defines the necessary definitions of our work, and shows some results of our feasibility study. Section 4 details how we've used this system to perform insider threat detection, which is a completely novel contribution of this work. Section 5 concludes the paper with an evaluation of the results and future areas of improvements.

## II. BACKGROUND

In the following paragraphs, we briefly identify some of the past work done in the areas of insider threats and automatic identification. We show how we will be able to use and expand upon these ideas to form our new system.

A malicious insider is defined by CERT [5] as a contractor, current or former employee, or business partner who has or had authorized access to an organization's network, system, or data and intentionally exceeds or misuses that access in a manner that negatively affects the confidentiality, integrity or availability of the organization's information systems [4]. The impact of insider crime can truly be devastating; in one recent case an employee stole blueprints on a new and classified process worth an estimated \$100 million and sold them to a Taiwanese competitor with the hope of obtaining future employment with that organization [4].

In general, database systems are well equipped to face external threats [6]. However, insiders pose a significant threat and have unique opportunities over others (including the system administrators and security engineers) when it comes to committing an electronic crime [10]. Insiders, by design, can bypass physical and technical security measures designed to prevent unauthorized access; the data must be made available to them in order for their business function to work properly. They are able to exploit flaws in the system because of their expert knowledge; they work with these critical data items everyday. This unique opportunity that is afforded to malicious insiders of familiarity and their knowledge of the methods required in order to access the important data is what makes identifying what is and is not critical in the data system such an important task in order to mitigate this risk. Because of this risk, the motivation is present to secure mission-critical data

and in order to do so it must be identified, preferable automatically [7].

There are commonalities among other automatic identification and text classification schemes that are applicable to automatically identifying critical data. A few of these systems will be examined here; good explanations of these systems can be found in [3], [4], [9], and [12].

Spam is a familiar object that is identified automatically. While there are many available spam filters, most follow a process that takes the incoming messages, parses them into smaller tokens based on the content of the message and other metadata such as the sender IP address, and then uses these tokens in a process that calculates a score or percentage that indicates the likelihood that a given message is in fact spam [8]. The process used in identifying whether or not a message is spam is often based on expert knowledge and machine learning techniques such as naïve Bayesian filtering or SVM based filters that classify the message based on the frequency that a given token appeared in message that was classified as a spam in the past. Some spam systems initially pass all messages through a list of rules that are applicable to all spam messages, which include rules such as a high frequency of a certain phrase like "SALE" indicates spam or a metadata indication of spam such as a recipient list with over 1000 recipients.

### III. PROPOSED SYSTEM

Briefly, the proposed system works by passing a document consisting of all transactions executed by a user through an SVM filter. The filter has been trained by an expert to have classifications on what is and is not critical. If the logs are representative of the actual system usage, it is a good indicator of normal operating procedures. Each user has a document consisting of all the transactions they have executed.

After the formation of the document, a score is calculated by using an SVM-based filtering process (defined in detail in a previous work). Then, once the average score is calculated for each user, questionable (or all) transactions are compared to this known historical data to find how 'close' these transactions are to the known good behavior. To capture the idea of 'closeness', we introduce the notation of a distance measure. The goal of this distance measure is to capture how far from normal behavior questionable requests appear. Queries that are too far from what is considered 'close' require more security procedures to occur. Actions that are within some threshold of normalcy are allowed with no further examination [15]. If a user suddenly attempts to access highly critical data when they have never done so before, this is a strong indication of an insider threat, and the query should not be allowed. If these transactions have already been seen, there is no need to re-filter the data; the score can be obtained automatically by performing a database lookup.

The summed score that is assigned by the SVM filter is then divided by the average criticality score. This results in a value that can be used to determine how far from

normal this action appears. An insider threat is detected if, after performing the above procedure, the following holds:

$$Dist(T_u) \geq D_{thresh}$$

where  $D_{thresh}$  is the threshold distance that is set to the maximum distance a suspicious transaction(s) can be before it is thought to be malicious or at least suspicious and requiring further intrusion detection actions. These security procedures might include notifying the DBA about the illicit behavior by triggering an alarm with the relevant information such as the time, username, and database objects accessed attached to the message, immediately disconnecting and rolling back the changes that the user made, or execution of some damage confinement and repair mechanisms that are built into the DBMS. These security precautions are dependent on the application and will be different for each use [11].

The proposed system works by building a profile of a user's normal behavior by using historical information; specifically database logs. We have chosen to use database logs as the primary information sources as they are often maintained over long periods of time and several other intrusion detection systems rely on accurate logs in order to build a user profile [Joachims02]. Furthermore, once the criticality level for each user has been established, we have proposed a method to further drill down and ultimately assign a score for each individual data item by way of an associative process that will be described.

The process proceeds as follows. The transactional logs are used to classify the queries that are run against it. The attributes that are used in the SVM classification are the items in the read set, the items in the write set, and the particular user that executed the transaction. Since the order of the read and written items is important, these elements are concatenated to form one discrete item, however, other layers of information are also included in the information that is processed by the SVM filter. Then, the logs are scanned for all transactions by the particular user, and these transactions are then used to form a document which consists of the ordered transactions as well as the name of the user. This textual document is then processed by the SVM. These textual transactions are different from normal textual documents in that they only consist of the transactional information that was found in the logs. This allows more rapid processing as the amount of uninformative text is greatly reduced.

Then, each user-level transactional document is assigned a criticality score. As each of the transactions is executed by a particular user, the score that is assigned to the document also indicates the level of critical data that a user executes. This score is stored by the system.

The next step of the proposed system includes using the above data to ultimately classify each data item (or column in a database environment) as either critical or non-critical with a classification score for each data item. As we already have a classification score for each user with the associated queries that they have run, a vector is made for

each data item that consists of the scores for each user (or class of users) that is present in the system. The values for each of these elements will be based upon the classification scores for each of the users that used those items; for this work five labels are used that indicate highly critical, critical, low critical, average, and below average criticality. Then, these vectors are categorized using the SVM filter once more, ultimately resulting in a score for each data item.

At this point, each data item has been assigned a score based upon both its content and its usage. These two values are summed to give a relative score for each data item, which orders the items that are most critical in the system. This identification, as was shown in section 1, is important if security engineers are to focus preventative measures on those items that are the most critical in the system.

These criticality scores can then be used for intrusion detection purposes. Running transactions are scanned, and tokenized. If a certain transaction greatly exceeds the criticality of data that is normally accessed, this is a sign of potential misuse, and further security procedures should occur. The proposed insider threat detection component of the automatic identification system would not replace existing intrusion detection systems; rather it would complement and add to them. The system can be run as an autonomous subsystem separated from the DBMS on a dedicated machine or the mechanism can be implemented internally in the DDMS using triggers. However, in the case of the latter, the performance of the database may be degraded as the execution of database triggers is normally a high resource consuming task.

As more transactions are encountered in the log, the lists of data used with the associated read/write frequency will change. While the transactional read/write sequences reveal much of the flow of the transactional, a potential area of improvement would involve passing information to the SVM filter about how the transactional sequences themselves are used with an associated ordering. These extra pieces of information will allow the SVM to have more input information to work with in classifying, which will help in the instances when the number of transactions is sparse in the historical logs.

At this stage, each user that was encountered in the historical database logs has been assigned a criticality score based on the SVM clustering. The SVM was trained with expert knowledge on those items in the database that are critical in terms of content without regards to how those data items are used.

As a future work, we also plan to use the time of day that the transaction was executed and perhaps the physical location of the user when they executed the transaction. This temporal and spatial information could be relevant as more industries spread across the globe and are widely distributed.

#### A. 4.7 Item Level Classification

From the above discussion, each user has been assigned a criticality score that is based upon the content of data that they access and this score is stored in a database. The score indicates the typical level of critical data that the user accesses during normal operating procedures. A need also exists to assign a criticality score to each data item based upon content. Having a score assigned to each data item (or column in a database environment) would allow the security engineers to focus on protecting the data, as opposed to catching human insider threats. This classification score would allow the engineers the ability to be much more proactive, instead of reactive, and ultimately greatly increase system security.

As mentioned previously, an expert in the system must train the SVM-based filter on what is critical and what is not. We take a similar approach with the method of item level classification based upon content. For example, if the security engineer indicates that the social security number field is critical based upon content, if we find another data item that is used in a similar manner by highly critical users as the SSN, this is an indication that these two data items are both critical. We already have the criticality usage data as was previously described. This indicates which users typically access critical data and which do not.

Then, this data is processed again by the SVM-based filtering system. If the system has been trained with some examples of known critical data items, the SVM filter will cluster these unknown data items into a cluster, with an associated error margin  $\epsilon$ . If  $\epsilon < \tau$  where  $\tau$  is the error margin threshold, this is an indication that the unknown item is used in a similar manner to the known critical items, providing an indication that this item is also critical. This extra layer of processing ultimately results in more knowledge about the system, which is beneficial when trying to identify critical data more effectively.

A score for each data item is stored in the system. As the system changes, it could be the case that the usage of the system changes and a formerly critical item is no longer viewed as being critical. When this is the case, the SVM procedure can be re-run with feedback from the users and new scores assigned. This process is implemented in pseudo code below:

#### Item Level Classification Algorithm

Inputs: List of users  $C$  with associated criticality score  $C_p$ , list of data items  $I$ , transaction log  $T$ , training samples  $R$ .  
Output: Criticality score  $K$  for each  $I$ .

```

for each  $I_n$ ,  $n=0$  to  $I_{max}$ 
for each  $C_p$ ,  $p=0$  to  $C_{max}$ 
  if(( $C_p$  read  $I_n$  in  $T$ ) or ( $C_p$  wrote  $I_n$  in  $T$ ))
     $V_n[p] = C_p[\text{crit-score}]$ 

```

```

else
  Vn[p] = 0
end for
Kn = SVM-Filter(Vn[0..Cmax], R)
return Kn
end for

```

The function SVM-Filter( $V_n[0..C_{max}]$ , R) uses the same training examples for each data item and works as described in section 2. After each of the criticality scores K have been calculated, the security engineer can then determine which data items are the most critical. When the security engineer decides where to focus limited security resources, these criticality scores can be used as a proactive guide, which is a valuable and novel tool, which is one the primary contributions of this work.

#### IV. EXPERIMENTAL RESULTS

The above algorithm was then implemented. The database log generation program that was used earlier was again used to generate a historical database log. For the following experiment, it was assumed that there were 100 users in the system and that the log consisted of 2000 valid transactions with each user executing an equal amount of transactions. Each transaction consisted of a random amount of reads and writes that varied between zero and twenty. The data items read/wrote were chosen at random from a pool of 300 data items.

##### A. Insider Threat Detection using Automatic Identification

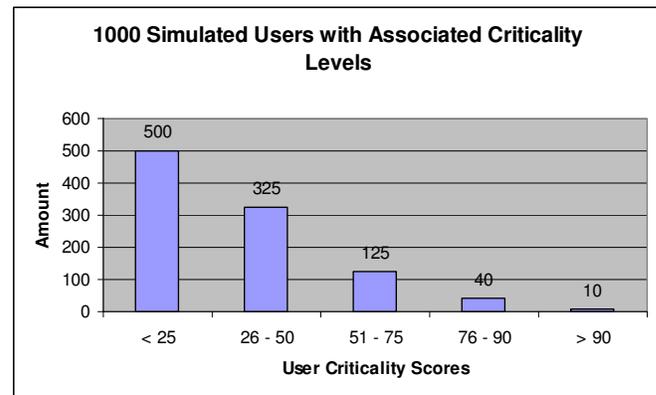
Once the average score is calculated for each user, questionable (or all) transactions are compared to this known historical data to find how 'close' these transactions are to the known good behavior. Several insider threat detection schemes use a similar idea involving a distance measure. The goal of this distance measure is to capture how far from normal behavior questionable requests appear. Queries that are too far from what is considered 'close' require more security procedures to occur. Actions that are within some threshold of normalcy are allowed with no further examination [Ha07]. If a user suddenly attempts to access highly critical data when they have never done so before, this is a strong indication of an insider threat, and the query should not be allowed.

##### B. Insider Threat Detection Example

We now show how we have evaluated and tested the insider threat detection model at the user level. There are generally only a small amount of high level users with large amounts of low level users. We have designed the experiment with this in mind and have designed the models with this type of skew.

We simulated 1000 users that will be tracked by the insider threat prediction model. Each user was assigned

a criticality score that was in one of five groupings. As Figure 1 shows below, 1% of the 1000 users are grouped together at the highest criticality level while 50% are at the lowest criticality level. We then assumed that approximately 1% of the total users were malicious; 5 in the first grouping were assumed malicious, 3 in the second grouping and one each in the 3 most critical groupings would attempt to access critical data with malicious motives.



**Figure 1: Experimental Distribution of Criticality Scores for 1000 Simulated Users**

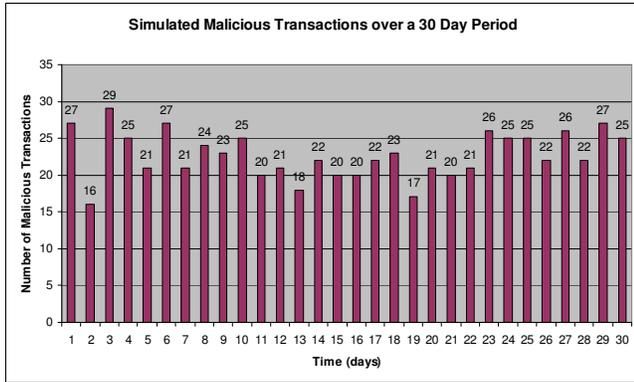
We then simulated each user executing various transactions against the system. We monitored these transactions for a period of 30 days. During each day, each of the 1000 users would be allowed to execute a random amount of transactions that varied between 1 and 10. During this period of 30 days, the malicious users would attempt to access critical data only during a fraction of the transactions that were processed by the system. We allowed this percentage to vary between 25% and 50% of the transactions that they executed each day.

We assumed that each of the malicious transactions were just greater than the various distance thresholds described below. We wanted to show how the various distance thresholds affected the amount of critical data that could be maliciously affected before alerts were generated. Finally, we used four different distance threshold schemes to test with:

- 10% threshold of max score for grouping.
- 10% threshold of current criticality score for the malicious user.
- Stricter on low level users, easier on high level users:
  - For the first two groupings the threshold was set to 5% of the max score for the group while the rest were set to 10%.
  - For the first two groupings the threshold was set to 5% of the current criticality score of the user while the others were set to 10% of their current criticality score.
- Stricter on high level users, easier on low level users:

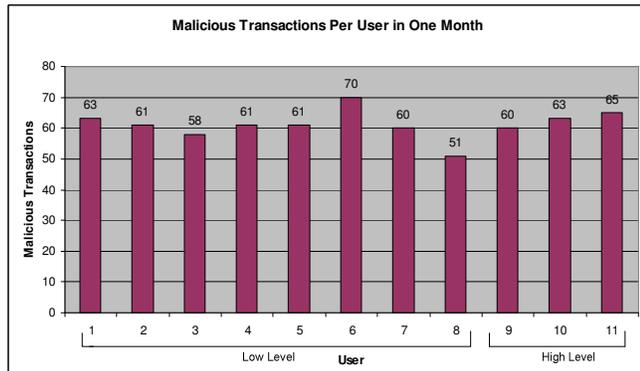
- a. For the last three groupings the threshold was set to 5% of the max score for the group while the other two groups were set to 10%.
- b. For the last three groupings the threshold was set to 5% of the current criticality score of the user while the others were set to 10% of the current criticality score.

The number of malicious transactions run against the system is shown below in Figure 2. The number of malicious transactions averaged around .29% of the total transactions executed each day, which were approximately 5525. For this simulation, the criticality scores of the 5 users in the first grouping were 23, 7, 19, 11, and 3, for the second grouping the scores were 29, 34, and 41 and for the following three groupings the criticality scores of the malicious users were 53, 82, and 94 respectively.



**Figure 2: Number of Malicious Transactions in Simulation**

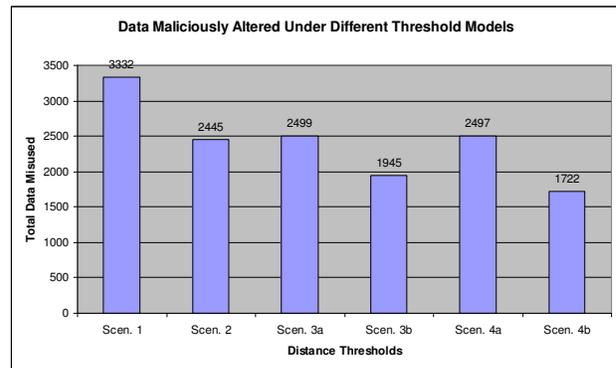
Each of these 11 users executed a varying amount of malicious transactions over the simulated month. These results are below in Figure 3.



**Figure 3: Total Number of Malicious Transactions Executed By Each User Over a Period of One Month.**

We then analyzed how much critical data would have been maliciously used under the above simulated

conditions with the different thresholds. As we defined the malicious transactions to be just over the distance thresholds, the amount of data maliciously used per user is equal to the number of malicious transactions executed by that user times the threshold for that particular scenario. As the distance thresholds allow different amounts of critical data to be used before raising a warning, the amount of critical data maliciously used would be different under the various scenarios. These values are then summed to arrive at a total value for the data maliciously used under that distance threshold. The results are presented below in Figure 4.



**Figure 4: Results of Simulations with Varying Distance Thresholds.**

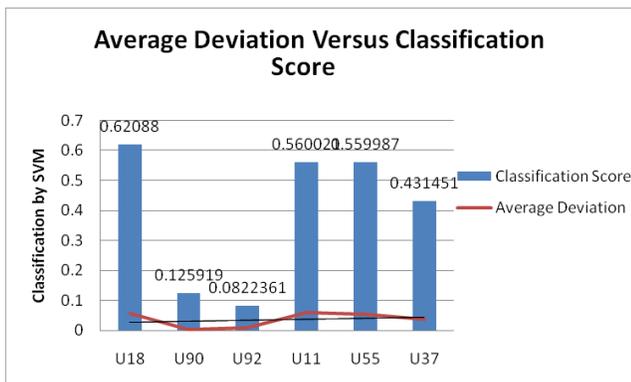
The results showed that the distance threshold did have a large impact on the system, even when the other variables were accounted for. In scenario 1 in Figure 4 above, all the users were given the same threshold, which was a function of what grouping they were in. This allowed the most data to be maliciously used as in some cases the users in each grouping were at the low end, so this threshold effectively gave them the maximum score of the group. Making the distance threshold metric user specific provided better results as scenarios 2, 3b, and 4b showed.

It is also better to be stricter on the high level users. They present the most risk, and when malicious actions are taken by them, on average they cause much more damage than malicious actions by low level users. This difference is showed in the disparity between scenarios 3a and 3b as well as 4a to 4b. Having a non-linear threshold that is a function of both the current criticality score of the user and the general level that they operate at provided the best results.

Statistical analysis was then performed on the results obtained in the previous figures. We wanted to show why we believed that the results of the classification were apt. Tests were performed with different SVM training characteristics; in all four different tests with different training setups were run against the system. As described in [Joachims02], the primary means of testing the validity of the system is to use settings to subtract from each element in the input data the mean of each element in that row; resulting in a row mean of zero; or by dividing each element

in the input vector by the standard deviation of each element in the row; giving the row a variance of one. In some circumstances, if the variables are heterogeneous in scale, both operations can be performed. If the system was working as claimed, then the average deviation of the scores from the mean in the various scenarios would be fairly constant and close to zero, meaning that the SVM system had enough training examples to accurately and properly classify unknown examples. As our control, we used the scores as originally calculated. We then reran the system with these adjustments as proposed in [Joachims02] and graphed the resulting deviations.

The primary difference in each of the SVM results was the SVM initial training setup; the user's actions in the log were all the same regardless of the setup. In order to show these results graphically, five users were chosen at random and examined under each scenario. The user's criticality scores ranged from zero (low criticality) to one (high criticality). The average deviation, if the classification worked properly, should be fairly constant regardless of the classification score, if the classification is working well regardless of the training data. These results are presented in Figure 5 and Figure 6.

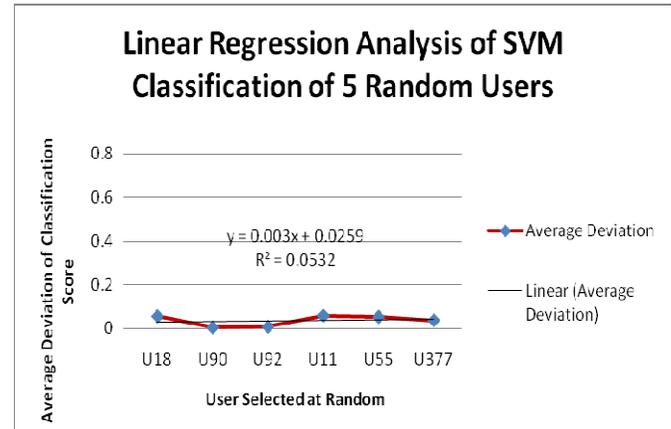


**Figure 5: Average Deviation from the Classification Score with Four Different SVM Training Setups**

Figure 5 shows the five users selected at random as the columns with their associated classification scores. The five users selected at random had scores that ranged from .62 to .08, which represented a wide classification range of scores. However, under the four different SVM training scenarios, the average deviation was fairly constant. This is shown in the red line. The average deviation over all five random users was approximately 2.5% under the four different training scenario setups. The average deviation is close to zero, lending credence that the system is capable of classifying users/data items effectively under various training samples. In all cases, ten samples were used in classification, as was used previously.

Linear regression analysis was then performed to calculate the correlation between the classification score and

the average deviation under four different training setups. This was done to confirm the heterogeneity of the results as suggested in [Joachims99]. These results are shown in Figure 6. The regression analysis was performed on the data as collected in Figure 5.



**Figure 6: Linear Regression on the Deviation**

The regression was fitted with a linear approximation, and the R squared value was calculated. The trend line and R squared value are shown in Figure 6. The trend line revealed the average error in classification as 2.59%. The R squared value was .05, which is close to zero. This is a further indication that the classification score and the average deviation are not related, affirming that the classification scheme is working equally well for highly critical users and lower level users.

This analysis affirms the claims that were made previously on the automatic identification by content methodology. The SVM-based filter, combined with training and expert knowledge to design the static rule set results in a system that can be used to detect the data items that are critical and the users in the system that have a high criticality value. This knowledge can then be used to better inform and guide the security procedures that must occur. This extra knowledge allows these security efforts to be much more focused and effective, which was one of the primary deterrents to the usage of most intrusion detection systems available in the market today.

## V. CONCLUSIONS

In this paper, we have shown how we have developed a system that can automatically identify critical data based on content. The purpose of this work was to present several experimental results as opposed to the system definition itself; these results were presented above. Our system is intended to complement existing intrusion detection systems to help fight the threat of malicious insiders abusing critical data items. We have developed a quantitative framework that applies very well to determining

which data items are critical based upon training and classification by an SVM based filtering methodology. The advantage of our approach is that it is a flexible and resource efficient technique that can be applied to any system that maintains a log of the transactions that are operated on the database. While our approach is aimed to be used by system administrators in a defensive mode of operation, it is also applicable to individuals who wish to use it in an offensive mode in order to efficiently target the areas of the data system where malicious actions will cause the most damage and disruption to the enemy.

We have also proposed using our automatic detection system in order to identify potential insider threats. We have developed a novel algorithm that scans the historical logs to develop a historical indicator of the criticality level that a user typically works with during a given day. Future transactions are monitored, and if the transaction greatly exceeds the criticality level that a user has typically used, this is a strong indication of misuse. This application will enable better identification and mitigation of insider threats.

Our proposal shows great promise to better reveal data item misuse and intrusions into databases. Our methodology is a new and novel system, using several existing concepts in different ways and using other concepts of our own invention. We have experienced several good results with our theoretical models, and we anticipate further positive results as the system grows to encompass more system metrics and as it is applied to real world trials.

#### REFERENCES

- [1] T. Joachims, "Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning", B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.
- [2] T. Joachims, "Optimizing Search Engines Using Clickthrough Data", Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002.
- [3] B. Yang, J. Sun, T. Wang, C. Zheng, "Effective multi-label active learning for text classification", In Proceedings of KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 917-926, New York, NY, USA, 2009.
- [4] D. Sculley, G. Wachman, "Relaxed online SVMs for spam filtering", In Proceedings of the Thirtieth Annual ACM SIGIR Conference, 2007.
- [5] D. Cappelli, A. Moore, T. Shimeall, R. Trzeciak, "Common Sense Guide to Prevention and Detection of Insider Threats", Carnegie Mellon University, 2008.
- [6] Insider Threat Integrated Process Team, Department of Defense (DoD-IPT), 2000. "DoD Insider Threat Mitigation" U.S. Department of Defense, 2000.
- [7] R. Anderson, T. Bozek, T. Logstaff, W. Meitzler, M. Skroch, K. Wyk, Research on mitigating the insider threat to information sys., RAND Corporation Report CF-163, 2000.
- [8] M. Whitman, "Enemy at the Gate: Threats to Information Security". In Communications of the ACM, Vol. 46, No. 8., 2003.
- [9] I. Abbadi, M. Alawneh, "Preventing Insider Information Leakage for Enterprises", In Proceedings of the 2008 Second International Conference on Emerging Security Information, Systems and Technologies, p.99-106, 2008.
- [10] J. White and B. Panda, "Automatic Identification of Critical Data Items in a Database to Mitigate the Effects of Malicious Insiders", in Proc. of the Fifth International Conference on Information Systems Security (ICISS 2009), Kolkata, India, LNCS vol. 5905, pp. 208 – 221, 2009.
- [11] D. Ha, S. Upadhyaya, H. Ngo, S. Pramanik, R. Chinchani, S. Mathew, "Insider Threat Analysis Using Information Centric Modeling" P. Craiger and S. Sheno (Eds.), In Advances in Digital Forensics III, Springer, Boston, 2007.
- [12] Q. Wang, Y. Guan, X. Wang, "A novel feature selection method based on category information analysis for class prejudging in text classification", In Proceedings of the International Journal of Computer Science and Network Security, 6(1): 113–119, 2006.
- [13] J. White, B. Panda, "Implementing PII Honeytokens to Mitigate Against the Threat of Malicious Insiders", in Proc. of the IEEE International Conference on Intelligence and Security Informatics (ISI 2009), Dallas, Texas, pp. 233, 2009.
- [14] L. Zhang, J. Zhu, T. Yao, "An evaluation of statistical spam filtering techniques", ACM Transactions on Asian Language Information Processing (TALIP), pp. 243–269, 2004.
- [15] J. White, B. Panda, Q. Yaseen, K. Nguyen, W. Li, "Detecting Malicious Insider Threats using a Null Affinity Temporal Three Dimensional Matrix Relation", in Proc. of the 7th Intl. Workshop on Security in Info. Sys. (WOSIS 2009), Milan, pp. 93 – 102, 2009.

# Analysis of Current Snapshot Options

SrinivasaRao Seelam  
East Carolina University  
Greenville, NC 27858  
Seelams05@students.ecu.edu

Chengcheng Li, Ph.D.  
Information and Computer Technology Program  
Department of Technology Systems  
East Carolina University  
Greenville, NC 27858  
liche@ecu.edu

**Abstract**—Data storage is essential to host databases, applications and web services. Storage Area Network (SAN) architecture has several benefits over Direct Attached Storage (DAS) devices in terms of efficient storage utilization and greater application performance. SAN architecture uses snapshot technology for Disaster Recovery (DR) and Business Continuity (BC). In addition, snapshots are typically performed to a cheaper disk system such as serial ATA. Traditionally, within the same SAN, if data on an original virtual disk is lost, snapshot provides a quicker recovery and can reduce application downtime tremendously. Conversely, it is cumbersome to perform snapshot operations on DAS devices. With the invention of Fibre Channel over Internet Protocol (FCIP) technology, snapshots can be replicated and copied between SANs that are placed at different physical locations and provides greater DR capabilities. Currently, all storage-level snapshots are disk-based. This paper explains the benefits of integrating tape-based appliances with SAN architecture for local and remote snapshots; especially, in terms of cost savings and efficiency.

**Key Words:** DR, Snapshot, Tape-based snapshot

## I. INTRODUCTION

The exponential growth of data, coupled with the high performance requirements of the Enterprise led to the development of the Storage Area Network (SAN) architecture. Fibre channel based storage area networks are becoming commonplace in today's enterprise data centers. SAN technology is gaining more popularity every year due to its high availability, reliability, performance, ease of storage management, and faster performance [13].

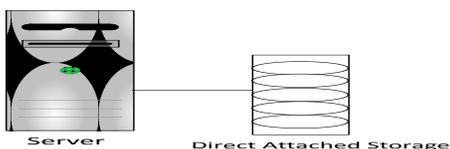


Figure 1. DAS architecture.

In SAN architecture, servers are connected to a centralized disk array, and it is much easier to perform all the disk administration using a single interface. In addition, a Virtual Disk also called a VDISK and a LUN is created and then presented to an individual server. As is shown in the following diagram, typically, a SAN is comprised of disks, FC Switch, servers and a tape library.

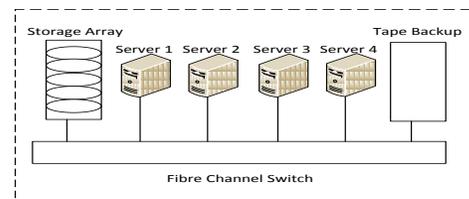


Figure 2. Typical SAN Architecture.

Fibre Channel (FC), Serial ATA (SATA), and Fibre Channel attached Serial ATA (FATA) are the most commonly used disk drives in a storage system. FC drives are used for VDISKS that require high performance, ATA drives are used for archival data and virtual tape libraries, and FATA drives are used for applications that require moderate performance [10].

It is essential to be able to backup and restore VDISKS to survive from a disaster. SAN-based backups are called Snapshots and they can be used for Disaster Recovery (DR) and Business Continuity (BC) operations. Snapshots are an exact replica of a VDISK and can be mounted for a quicker recovery if needed. For instance, if a VDISK is mounted on a server and accidentally deleted or has a virus or malware, it can be recovered as long as a snapshot is available on SAN for the particular VDISK. The data on the recovered VDISK is as current as the time when the snapshot was performed. Snapshots can use copy-on-write (COW) or redirect-on-write (ROW) methods [16]. The actual updating process of snapshot differs

between COW and ROW methods. Nonetheless, anything that is tied to a particular host carries the limitations of a DAS. Snapshots performed at a host-level can be slower due to other operations simultaneous running on host [3]. Although both Block-level and File-level snapshot are popular, Block-level snapshots yields better performance compared to file-level snapshots [16]. Typically, snapshots are performed on SATA and FATA disks. Nonetheless, if there is a technology that connects SANs at two different physical locations, it will provide better BC and DR.

With the invention of Fibre Channel over Internet Protocol (FCIP) protocol, FC devices can take advantage of Wide Area Network (WAN) technology and communicate with SANs located at other physical locations [12]. This provides greater flexibility when performing snapshots from one SAN to another to achieve DR and BC requirements. It is also possible to replicate data from one SAN to multiple SANs that are located at different physical locations [1]. However, currently, regardless of the protocol and method used for a snapshot on a SAN, snapshots are still created on disk. Now is the time to think about new and novel approaches that will offer significant benefit to companies.

In this paper, the option of creating snapshots on tapes in SAN architecture will be evaluated thoroughly. In the following recommended design, tape libraries are installed at each SAN location use FC connection and connected to the FC fabric. SAN management software that is capable of performing snapshots to tapes will be used. In addition, differential, incremental and full backups are performed from the snapshots located on the tapes to another set of tapes without reading data from the disk array.

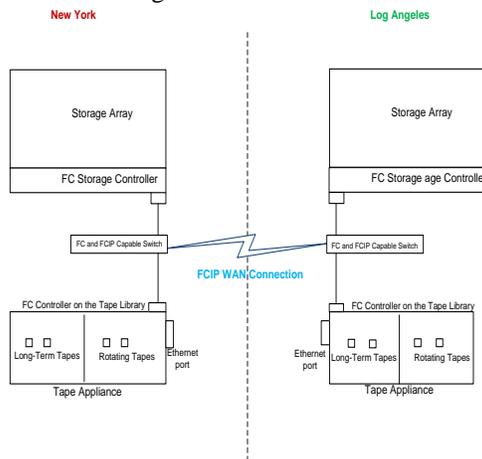


Figure 3. Proposed Design.

## II. BACKGROUND

A snapshot is a backup copy of a virtual disk (VDISK). As explained previously, snapshots are very important to be able to provide the required BC and DR operations. The following diagram explains how snapshots can be made at different time intervals.



Figure 4. Snapshot process.

As illustrated in figure 4, if there is a data corruption or loss of “Home\_Directories\_Accounting” after 4:16 PM on Monday, the system administrator has a choice of restoring it from either the 12:30 PM or the 4:15 PM snapshot. However, the ability to perform snapshots is not native to the storage hardware, but requires additional software.

SAN-based Snapshot software products rely on array-based operations to make snapshots; so, host I/O or CPU resources are not involved in vdisk preparation. Eliminating the host from this snapshot process provides a much speedier solution [3]. More importantly, these software products write the snapshot to disk. In other words, upon completion of the snapshot process, both the source vdisk and snapshot reside on the disk. This paper introduces the idea of using magnetic tape media as a snapshot-destination device, and its advantages and disadvantages compared to disk. A typical Storage Array Network consists of a Storage Array, FC Switch and Tape Library. The following section explains each of these components in greater detail.

### A. Storage Array

The storage array is comprised of the storage controllers and high speed disks. Companies like EMC are constantly innovating, for example, its product Virtual Matrix Architecture (VMA) can integrate industry-standard components with EMC hardware [5]. VMA can integrate Intel Quad-core Xeon and other high-end processors with the standard EMC storage array and the memory installed in the individual components can increase the total size of the total global memory. This model can handle hundreds of

thousands of terabytes of storage and tens of millions of input/output per second (IOPS) [5]. No longer is it uncommon to have Enterprise Flash, Fibre Channel, SAS drives, and iSCSI modules in the same storage array. Fully Automated Storage Tier (FAST) is a model based on such combinational hardware.

More advanced storage technologies, such as, Fully Automated Storage Tier (FAST) are available from storage vendors such as EMC now. FAST is a great automation technology when it comes to data relocation capabilities. It can automate the movement of data across multiple storage tiers based upon business models, predictive models, and real-time access patterns. Amazingly, it can relocate data non-disruptively to different storage tiers and RAID protections, including ultra-high performing enterprise flash drives, Fibre Channel disk drives and high-capacity SAS disk drives [5].

**B. FC Switch**

The Fibre Channel (FC) switch creates the FC fabric by connecting FC-attached devices. A properly designed Fibre Channel environment is crucial in ensuring peak performance for the entire SAN. When deciding on a specific FC switch, reliability, performance, interoperability, migration support, usability, scalability and manageability are very important factors to consider [9].

For a given FC switch, it is important to consider facts such as speed, throughput, line rate, encoding, Retimers in the module, and transmitter training [14]. Following table are all important to analyze on an FC Switch.

Table I  
Comparison of 4, 8 and 16 GFC Switches

Speed	Throughput (MBps)	Line Rate (Gbps)	Encoding	Retimers in the module	Tx Training
4 GFC	400	4.25	8b/10b	No	No
8 GFC	800	8.5	8b/10b	No	No
16 GFC	1600	14.025	64b/66b	Yes	Yes

As table 1 suggests, 16 GFC uses 64b/66b encoding; therefore, its efficiency is at 97%. 8 GFC and 4 GFC use 8b/10b; therefore, their efficiency is at 80%. When there is a requirement to connect multiple fabrics, trunk connections

can be setup between switches. Great care is required when designing the trunk connections [4].

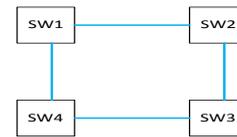


Figure 5. FC Switches with Trunk connections.

Trunk setup can have an impact on overall FC fabric performance. Improper trunk configuration between switches can cause deadlock and traffic fairness issues [4].

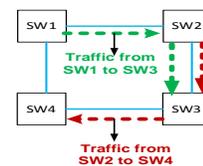


Figure 6. Deadlock and fairness issues in FC switches.

As it is shown in Figure 6, if there is continuous FC traffic from all the nodes on SW1 going to SW3, and if there is continuous FC traffic from all the nodes on SW2 going to SW4, no additional FC data can be forwarded on the FC fabric. In other words, if the buffers of all the switches installed in the loop are filled, then additional individual frames cannot be forwarded, and this situation is called a deadlock [4]. Deadlock can cause unfair traffic distribution. In order to avoid deadlock and traffic fairness issues, the following FC HUB model can be used.

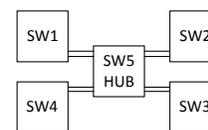


Figure 7. FC HUB Model.

In this model, there are two connections used for each trunk. SW5 acts as an intelligent hub. Having two trunk connections helps to allocate the traffic better as one connection can handle inward traffic and the other connection can handle outward traffic. If the FC fabrics are located at different physical locations, there is a need for some kind of FC over IP (FCIP) connection. Cisco MDS series switches are examples of FCIP hardware that can connect

Fibre Channel fabrics with IP networks [12]. Security is maintained by creating FC Zones. Devices that need to communicate with each other need to be placed in a separate zone [14].

C. Tape Library

Typically, tapes are used for backup and archival purposes. Although tape technology suffers with slower read times due to high seek time, they are widely used for data archival purposes [8]. There are several technologies of tapes available including DLT, SDLT, LTO and others. Linear Tape-Open (LTO) has gained some popularity when it comes to the actual amount of data that they can hold.

Table II  
LTO comparisons

LTO Technology	LTO1	LTO2	LTO3	LTO4	LTO5
Speed in MB/s	20	40	80	120	140
Data Capacity in GB	100	200	400	800	1500

Tape libraries are comprised of multiple tape drives and several media/tape slots; tape libraries are equipped to perform backups of several terabytes of data without any manual intervention.

III. COMPARISON

In this section, different snapshots i.e. disk-based and tape-based snapshots will be compared and analyzed in great detail. Moreover, the hardware environment for both methods is kept similar for better comparison. All the calculations and assessments are made assuming the following hardware is used.

16 Gbps Fibre Channel Switch, 4 Gbps Disk Controller, 118MB/Sec buffer to media speed, 600 GB FC disk drives with 4 Gbps interface, Linux Server with 4 Gbps Host Bus Adapter, 1 Gbps Local Area Network (LAN) bandwidth, 4 Gbps Fibre Channel Based Tape Library, Four LTO-5 Tape Drives with 140 MB/s or 1.12 Gbps bandwidth for each drive.

A. Disk-based Snapshots

In this section, an option of using disk as a snapshot target is being analyzed.

1) Disk-based snapshots within the same Storage Array:

The following diagram lists all the devices that are involved in the snapshot process with their corresponding bandwidth limitations.

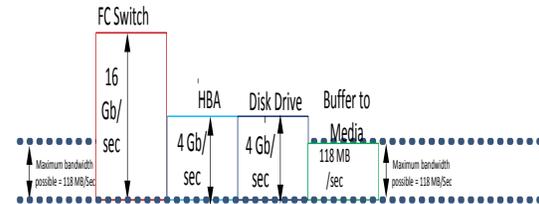


Figure 8. Analysis of maximum bandwidth possible.

As it is shown, certain devices involved in the snapshot process can handle 16 Gb/sec, and 4 Gb/Sec. The data that has to go through 16 Gb/Sec and 4 Gb/Sec also has to go through 118 MB/Sec. In other words, there is a bandwidth bottleneck of 118 Mb/Sec. Therefore, the maximum bandwidth possible for the snapshot operation is 118 MB/Sec. In order to perform a snapshot with the given configuration to make a 10 TB or 10240000MB, it takes 86780 (10240000/118) seconds, or 1446 minutes or 24.2 hours approximately under ideal conditions.

2) Disk-based snapshots between Storage Arrays Connected using FCIP Link:

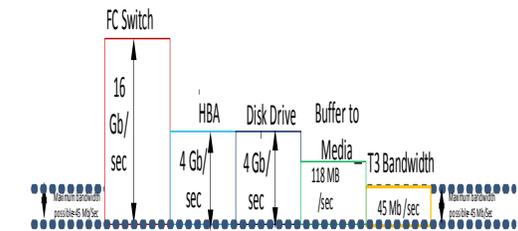


Figure 9. Analysis of maximum bandwidth possible with T3.

If the snapshot has to be performed to a remote SAN located at a different physical location connected by a T3 WAN connection, the best rate at which the data can be transferred is 45 Mbps. Therefore, it can take 533 (10240000\*8/45) hours approximately under ideal conditions.

B. Tape-based Snapshots

Traditionally, tapes are used for incremental, differential, and full backups. Tapes are taken offsite for data protection and to be able to recover data in case of a disaster [8]. Incremental or differential backups are performed on daily basis. Full backups are performed on weekly, monthly and yearly basis. The purpose of this paper is to promote the idea of using tapes for creating snapshots of VDISKS as opposed to disk. In addition, this paper assumes that FC attached tape library is installed and properly configured as explained in Figure 3, proposed design.

1) Tape-based snapshots within the same Storage Array:

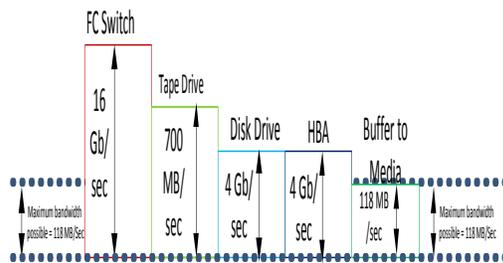


Figure 10. Analysis of maximum bandwidth possible with Tape.

As it can be seen from the above diagram, the maximum possible bandwidth for this model is 118 MB/Sec. Therefore, it will take 24.2 hours approximately to make a 10 TB snapshot within the same disk array.

2) Tape-based snapshots between Storage Arrays Connected using FCIP Link:

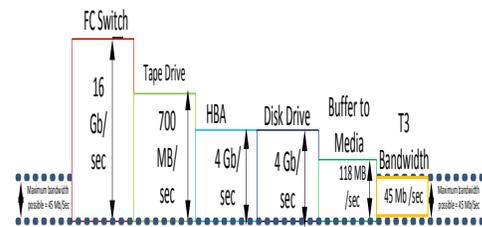


Figure 11. Maximum bandwidth possible with T3.

If the snapshot has to be performed to a remote disk or tape that resides on a SAN that is at a different physical location using T3 WAN connection, it will take approximately 533 hours.

IV. DISCUSSION

Present-day tape libraries can connect to the FC fabric directly and do not need FC to SCSI interface. If the snapshots are done to tape, incremental, differential, full backups can be performed from the snapshot residing on a tape to another tape. It will improve overall backup performance. In disk-based snapshot method, making a snapshot of a vdisk from one SAN to another SAN can take several weeks and utilizes much of the available WAN bandwidth. Conversely, if the snapshot is performed to a tape, and if the snapshot tapes can be shipped from one location to another using a secure postal or courier service, and then use WAN-based FCIP connection for the snapshot changes to replicate, it will save the WAN bandwidth for important applications. The Cost of each Gigabyte of disk storage is three times as expensive as tape storage [11]. If the vdisk sizes are growing exponentially, snapshot sizes also increase. More disks means, more cabinets, controllers, cooling, electricity, space, personnel and maintenance cost. If there is a natural disaster or array failure the disk-based snapshots will be gone. However, if tape-based snapshots are used, these data recovery from a snapshot are not going to be a problem.

Table III Overall comparison

Element	Disk-based snapshots	Tape-based snapshots
Time takes for a 10 TB snapshot within the array.	24.2 hours	24.2 hours
Time takes for a 10 TB snapshot between arrays connected using T3.	533 hours	Between 23.5 and 200 hours.
Cost for each Gigabyte of storage.	3 times more cost.	3 times less cost.
Rotational backup impact on the SAN?	100% impact.	0% impact.
WAN usage inefficiency for initial snapshot	100% inefficiency.	0% inefficiency
Are cooling, storage, electricity, personnel, additional space, and maintenance costs going to increase with snapshots?	Yes. Going to be very expensive.	No.
Can the disk array failures and natural disasters damage snapshots?	Yes. Can be damaged.	No.

## V. CONCLUSION

Based on the evidence presented in the discussion section of improved speed, increased flexibility and lower costs, it is obvious that the storage area network industry needs to start looking at integrating tape libraries into SAN fabrics and develop software to be able to write VDISK snapshots directly to tapes.

## REFERENCES

- [1] Ahmad, N.; Sidek, R.M.; Klaib, M.F.J.; Jayan, T.L.; , "A Novel Algorithm of Managing Replication and Transaction through Read-one-Write-All Monitoring Synchronization Transaction System (ROWA-MSTS)," *Network Applications Protocols and Services (NETAPPS), 2010 Second International Conference on* , vol., no., pp.20-25, 22-23 Sept. 2010  
doi: 10.1109/NETAPPS.2010.11  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5636026&isnumber=5635594> (references)
- [2] Cai, Y.; Fang, L.; Ratemo, R.; Liu, J.; Gross, K.; Kozma, M.; , "A test case for 3Gbps serial attached SCSI (SAS)," *Test Conference, 2005. Proceedings. ITC 2005. IEEE International* , vol., no., pp.9 pp.-660, 8-8 Nov. 2005  
doi: 10.1109/TEST.2005.1584027  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1584027&isnumber=33431>
- [3] Chang-Soo Kim; Bak, Y.; Dong-Jae Kang; Young-Ho Kim; Hag-Young Kim; Myoung-Jun Kim; , "A method for enhancing the snapshot performance in SAN volume manager," *Advanced Communication Technology, 2004. The 6th International Conference on* , vol.2, no., pp. 945- 948, 2004  
doi: 10.1109/ICACT.2004.1293007  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1293007&isnumber=28787>
- [4] Cherkasova, L.; Kotov, V.; Rokicki, T.; , "Designing fibre channel fabrics," *Computer Design: VLSI in Computers and Processors, 1995. ICCD '95. Proceedings., 1995 IEEE International Conference on* , vol., no., pp.346-351, 2-4 Oct 1995  
doi: 10.1109/ICCD.1995.528832  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=528832&isnumber=11610>
- [5] EMC announces new data storage architecture. (2009). Worldwide Computer Products News., n/a. Retrieved from <http://search.proquest.com/docview/224500376?accountid=10639>
- [6] EMC Clariion Specification Sheet. EMC Corporation. <http://www.emc.com/collateral/hardware/specification-sheet/c1146-clariion-cx3-20-ss.pdf>
- [7] HP StorageWorks XP 24000/XP 20000 Disk Array. [http://h18006.www1.hp.com/products/quickspecs/12711\\_div/12711\\_div.pdf](http://h18006.www1.hp.com/products/quickspecs/12711_div/12711_div.pdf)
- [8] Jr, S. J. S. (2003). Making the case for both disk and tape. *Network Computing*, 14(4), 71. Retrieved from <http://search.proquest.com/docview/215441784?accountid=10639>
- [9] KeyLabs completes multi-vendor fibre channel switch comparison. (1999). *PR Newswire* , 1. Retrieved from <http://search.proquest.com/docview/449516393?accountid=10639>
- [10] Milligan, C.; Selkirk, S.; , "Online storage virtualization: the key to managing the data explosion," *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on* , vol., no., pp. 3052- 3060, 7-10 Jan. 2002  
doi: 10.1109/HICSS.2002.994288  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=994288&isnumber=21442>
- [11] Moore, R.; D'Aoust, J.; McDonald, R; David.; Disk and Tape Storage Cost Models. [http://www.cs.ucsb.edu/~chong/290N/dt\\_cost.pdf](http://www.cs.ucsb.edu/~chong/290N/dt_cost.pdf)
- [12] Nikolaidis, I.; , "IP SANs, a guide to iSCSI, iFCP, and FCIP protocols for storage area networks [Book Review]," *Network, IEEE* , vol.16, no.2, pp.5, Mar/Apr 2002  
doi: 10.1109/MNET.2002.993214  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=993214&isnumber=21419>
- [13] Sarkar, P.; Voruganti, K.; Meth, K.; Biran, O.; Satran, J.; , "Internet Protocol storage area networks," *IBM Systems Journal* , vol.42, no.2, pp.218-231, 2003  
doi: 10.1147/sj.422.0218  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5386862&isnumber=5386848>
- [14] The Benefits and Application of 16 Gbps Fibre Channel. Brocade Corporation. [http://www.brocade.com/downloads/documents/technical\\_briefs/Application\\_Benefits\\_16GFC\\_GA-TB-313-00.pdf](http://www.brocade.com/downloads/documents/technical_briefs/Application_Benefits_16GFC_GA-TB-313-00.pdf)
- [15] Xianbo Zhang; Dingshan He; Du, D.H.C.; Yingping Lu; , "Object Placement in Parallel Tape Storage Systems," *Parallel Processing, 2006. ICPP 2006. International Conference on* , vol., no., pp.101-108, 14-18 Aug. 2006  
doi: 10.1109/ICPP.2006.55  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1690610&isnumber=35641>
- [16] Weijun Xiao; Qing Yang; Jin Ren; Changsheng Xie; Huaiyang Li; , "Design and Analysis of Block-Level Snapshots for Data Protection and Recovery," *Computers, IEEE Transactions on* , vol.58, no.12, pp.1615-1625, Dec. 2009  
doi: 10.1109/TC.2009.107  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5184809&isnumber=>

# System-Manipulation using Windows-Messaging-Hooks

Peter Schartner  
System Security Group  
Klagenfurt University  
9020 Klagenfurt, Austria  
Email: peter.schartner@aau.at

Martin Foisner  
BSc-Student  
Klagenfurt University  
9020 Klagenfurt, Austria  
Email: mfoisner@gmx.net

**Abstract**—Recording and replaying keystrokes and/or mouse-moves in form of macros in order to speed up complex or annoying user-interactions is common practice. This process uses so called hooks (entry points for additional software) to integrate special event processors into the operating systems' event processing path. Since there is no check, if these additional event processing methods are malicious or not, this opens the door for well known attack scenarios like password-sniffing. But even worse, the new message processor may drop system messages or insert new (forged) messages into the event queue. This paper first describes the background of Windows messaging hooks and new attack scenarios based on dropping and inserting event-messages. Additionally we provide details of a proof-of-concept implementation, which manipulates the certificate management of Mozilla Firefox, and give some discussion on other attacks based on Windows messages and possible countermeasures.

## I. INTRODUCTION

The messaging-system is an important instrument to provide the communication between the operating system and a window-based application or in-between window-based applications. For every event there is (or should be) a message which describes what happened. For example, if a user minimizes a window, the system sends the `WM_SIZE` message with an additional parameter (`SIZE_MINIMIZED`) – which indicates that the window has been minimized – to the window or more precisely to the windows message handler. This message-handler receives the message and acts according to the message.

But messages can also be sent by other applications. Think of recording and replaying keystrokes and/or mouse-moves in form of macros in order to speed up complex or annoying user-interactions is common practice. This process and some system analysis tools like Microsofts spy++ [1] use so called hooks (entry points for additional software) to integrate special event processors into the operating systems' event processing path. Since there is no check, if these additional event processing methods are malicious or not, this opens the door for attack scenarios like password-sniffers. Additionally, messages of the system may be dropped, delayed, or manipulated and new (forged) messages may be inserted into the event queue.

The remainder of this paper is structured as follows: first we briefly describe the background of Windows messaging

hooks and present new attack scenarios based on dropping and inserting event-messages. After that we provide details of a proof-of-concept implementation which manipulates the certificate management of Mozilla Firefox 3.6.10. The paper concludes with some discussion on other attacks based on Windows messages and possible countermeasures. Note that for this paper the question “How to install the malware at the victims PC” is out of scope. For simplicity, we assume that the malware has been installed already (e.g. by the user who installed some trojan horse).

## II. RELATED WORK

One possible attack, the so called “shatter attack”, was described in a paper of Chris Paget [2], [3] in August 2002. With the `WM_SETTEXT` message a malicious code is copied into a text-field of an application and is hence transferred into its memory. After inserting the malicious code, the `WM_TIMER` message was used to jump to the address of the malicious code to get it running. Countermeasures (“Shatter-proofing Windows”) against this and other message-based attacks have been discussed in 2005 by Close, Karp, and Stiegler [4], until now, none of them has been integrated into current operating systems.

## III. ATTACK METHOD

The windows-based graphical user interface of the Windows Operating System family is based on sending and processing messages related to (user) events and actions. Details to the concept of messages can be found in the Microsoft Developer Network (MSDN) [5]. In order to provide some entry point for extensions of the message processing, windows uses so-called hooks, ... *point[s] in the system message-handling mechanism where an application can install a subroutine to monitor the message traffic in the system and process certain types of messages before they reach the target window procedure* (see [6] for details on implementing and using hooks).

Unfortunately, using hooks is not restricted to privileged users or the operating system only. So, any program can install its own message-handler. Like many other useful programs (e.g. macro-recorders), some malware (e.g. keyboard-sniffers)

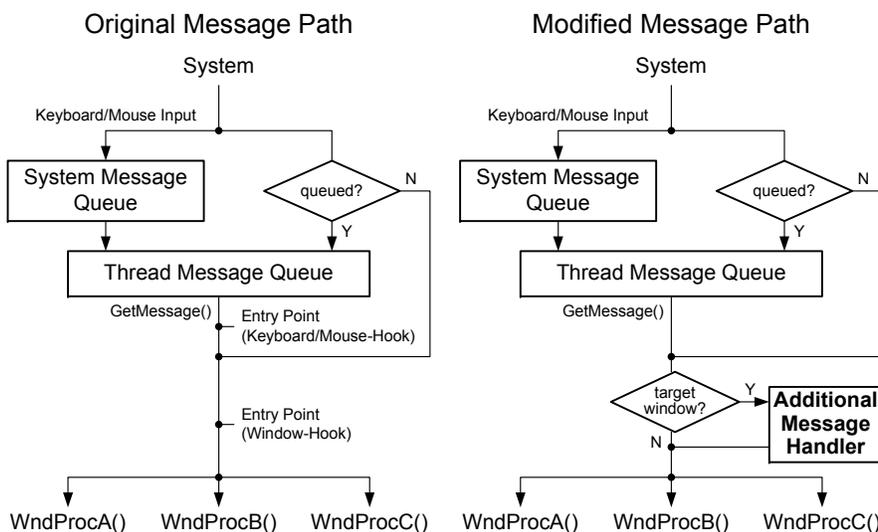


Fig. 1. Original (left) and Modified Message Path (right)

do so, too. The concept described below installs a message-handler in order to manipulate the behavior of security-critical user-dialogues. Examples include the management of the (Windows) firewall, anti-virus software, user-accounts, certificate-storage.

Figure 1 shows the standard Windows message handling on the left side. Messages related to specific events (triggered by hardware, software or the user) may be queued or unqueued. When inserting the a new message-handler at the right hook, all messages in the system can now be controlled by this message handler (see right side of figure 1). Now, messages of the system may be filtered based on certain rules (like window-ID or event-type), dismissed, delayed or modified. Additionally, new messages can be inserted in the message queue as well.

#### IV. PROOF-OF-CONCEPT

When opening a HTTPS-session to a site, whose certificate can not be automatically verified, the Mozilla Firefox pops up some warning dialogue. Now the user has three options: he may either cancel the communication or he may continue by adding a temporary or permanent exception. To choose the first option, the user simply clicks on the button “Get me out of here!” and after closing the connection, Mozilla will return to the previously open webpage. If the user wants to continue, he has to click on “I Understand the Risks”, which displays an additional warning and a button to open the dialogue “Add Security Exception”. By use of this dialogue, the user can accept the certificate in question for the current session or from now on for all sessions. In order to distinguish long-time-and one-term-acceptance, the user has to check or uncheck the according checkbox (see figure 2).

As a proof-of-concept, we implemented a tool, which manipulates the behavior of the “Security Exception” dialogue used in the manual certificate management process of Mozilla Firefox. It has to be mentioned, that this attack doesn't use any

specific weakness of Mozilla Firefox, but uses only features provided by the Windows operating system. So all other web browsers and any programm may be manipulated in this way, too!

The prototype, which is not detected as malware by current security suites like McAfee VirusScan Enterprise & Anti-Spyware Enterprise 8.7.0i or Norton Internet Security 2010, manipulates the behavior of the following elements in the dialogue “Add Security Exception”:

- Checkbox “Permanently store this exception”
- Button “Confirm Security Exception” (called OK-button)
- Button “Cancel” (called Cancel-button)

Manipulation methods:

- **Permanent Storage of the Exception:** Here the tool waits for a left-click on the OK-button. If this click is detected, a message faking a click on the checkbox is inserted into the messaging queue right before the message of the left-click-event. By this the attack “inverts” the user's intention concerning the duration of the exception.
- **Interchanging User Actions:** This attack interchanges the actions of the OK- and the Cancel-button by identifying and dismissing the original event message and inserting the opposite one. This again “inverts” the user's intentions.
- **Denial-of-Service:** The tool disables the OK- and/or the Cancel-button and hence reduces the user's choice of actions. When disabling both buttons, this results in a Denial-of-Service-attack where the user is unable to accept certificates which can not be verified automatically.

Figure 3 shows the log-messages of the tool FireMan (new messages are pushed on the top of the stack) when the OK-button is disabled by dropping the according messages (LEFT\_MOUSE\_DOWN & LEFT\_MOUSE\_UP).

- 1) Firefox activated (370f1e): FireMan detected Firefox (Window-ID 0x370f1e).



Fig. 2. Adding a Security Exception in Mozilla Firefox

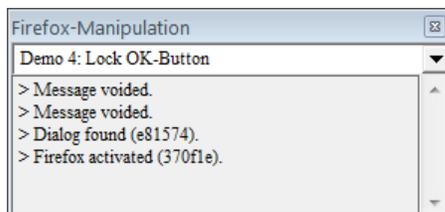


Fig. 3. Log-messages of the Tool

- 2) Dialog found (e81574): FireMan detected the dialogue “Add Security Exception” with Window-ID 0x e81574.
- 3) Message voided: FireMan dropped the message LEFT\_MOUSE\_DOWN.
- 4) Message voided: FireMan dropped the message LEFT\_MOUSE\_DOWN and hence disabled the OK-button.

## V. OTHER ATTACK SCENARIOS

Some other attacks based on manipulating Windows messages include:

- Keystroke-recorders which may be used to sniff passwords or other critical information (like mobile TANs or SMS-TANs). Think of e-banking, where the attacker first sniffs the users password and the TAN which is entered to authorize a specific transaction. He then blocks the users click on the button “Complete transaction” as long as it takes him to enter, authorize and send his transaction.
- A special “key-logger” for software or websites where you have to enter a password by clicking on buttons. Everytime the left mouse button is used, a small screenshot of the surrounding area of the cursor is made and sent to the attacker.
- Since the operating system has no way to distinguish faked events from original ones, the attacker could try to open a command shell or website in order to download and install other malware during the idle-time of the PC.

Unfortunately command shells do not accept messages, but web browsers do as we verified by our proof-of-concept implementation.

- (Unnoticeable) Input simulation on buttons, icons or whatever can be clicked on, can be activated. Some elements can be activated by using e.g. WM\_COMMAND or BM\_CLICK. These messages have the advantage, that they don’t need the placement of the element, but the disadvantage, that they only work with elements, which have an own window-handle. Other elements with no window-handle, which occurs e.g. in the Windows-firewall window, are a bit more difficult to activate, because they need to be clicked on with the mouse-cursor. The positions of the elements depends on the screen-resolution and the scale and position of the window. With functions like `GetSystemMetrics()` the screen-resolution can be found out and with `SetWindowPos()` the window can be sized/positioned; so the position of the elements isn’t a mystery anymore. Now a fake mouse-click can be placed using the WM\_LBUTTONDOWN/UP messages or the `SendInput()` function (which triggers a chain of messages).
- Windows using subclassed window-procedures (that means application-instances use the same window-procedure). For example the dialogue-window ‘file options’ could be opened from the Windows-explorer through the Windows-firewall window (worked on Windows 7 Home Premium by sending WM\_COMMAND with control identifier 28803 to the child-window using the window-class `ShellTabWindowClass`). With that an attacker can manipulate functions of an application, which isn’t even running.

## VI. ABOUT COUNTERMEASURES

First, we would like to explain, why some straight forward countermeasures – especially message-based or countermeasures based on monitoring the user’s behavior – will not work.

- Sequence numbers to detect missing/deleted messages: Actually, there is no need to delete messages. Simply setting it to `WM_NULL` [7] will have the effect, that this message will be ignored by the recipient window.
- Tracing user-behavior (like mouse movement): The idea behind this countermeasure is, that “jumps” to buttons are not longer possible. But unfortunately, the according mouse movement may be faked too. If the malware monitors and records the user’s behavior, it may ‘move’ the mouse in a way which is indistinguishable from the way, the user would do.

In the remainder of this section we will discuss a countermeasure based on the Mandatory Integrity Control (MIC [8]), which Microsoft added as a core component to their security architecture starting with Windows Vista. When using MIC, each object, like an application or other files, gets one of five integrity levels (untrusted, low, medium, high, system), which describes its trustworthiness. Objects have only access rights on other objects on the same or lower integrity level – that also includes sending messages or installing hooks. So MIC protects files on a higher level from being manipulated from files on a lower level, but grants access to files on the same or lower level.

The problem with MIC is (at least) twofold: most commonly, files do not have an integrity label at all and are treated like objects with medium integrity level. Additionally, if a file is duplicated, its integrity label isn’t copied. It doesn’t matter on which level the original file was (e.g. high or low), the copy gets on the medium level.

A user might come up with the idea to raise the integrity level of his internet browser (e.g. Mozilla Firefox) to high, to protect it from malware (which will be on level medium per default). Now its better protected against attacks exploiting the Windows-messaging-system, but with this action he opened a door for every malware which operates through that application using another technique. Same thing if he assigns low integrity level: Now a lot of applications can manipulate it. So maybe it’s not a bad thing, that the useability of setting up the integrity levels isn’t comfortable for common users (the integrity level can be changed via command-prompt using `icacls.exe` or `chml.exe`).

What’s needed is a tool, which protects files from each other. It should be compatible to already existing files, should provide useability for common users and furthermore it shouldn’t be vulnerable against what it should protect from (messaging-system exploits).

If we use a console application, it is not affected from the messaging-system, the compatibility to already existing files could also be achieved somehow, but the useability for common users would be quite bad.

Maybe an anti-malware software could do that job. A kind of ‘personal message-firewall’ could be another (probably better) solution. Every time an unknown application tries to install a hook or sends a message to another application, the user gets a notification and can decide if he blocks it or not.

The problems with these approaches are the following: The mapping between a process and its window-handle (which changes every time a new process is started), the interception and analysis of a message (who is the sender, who is the receiver?) and of course the protection against messaging-system exploits. That’s why such an application must be integrated into the operating system to resolve these problems.

The advantage over an anti-malware software is, that this software is independent from anti-malware definitions and – if it has a special status, that no other process can send messages to it – that it is resistant against messaging-system exploits.

A disadvantage is, that an unexperienced user could block good software or let bad software pass, because he just doesn’t know what the application is good for or what it does. That could be solved via a simulation: the operating system could redirect the messages into an extra message-queue, create a kind of ghost-window of the receiver-process and show the user what will happen, if he blocks or passes the messages of that specific sender-process.

## VII. CONCLUSION

Most operating systems like Windows or Android are message- or event-based. Up to now, everybody can insert additional message processing procedures. Unfortunately, the operating system has no way to verify the trustworthiness of these message handlers, so the door ist wide open for an attacker. In this paper we presented an attack method based on manipulating the message processing and discussed a proof-of-concept implementation. This prototype – which has not been detected by anti-virus- and anti-spyware-software – manipulated the behavior of Mozilla’s Firefox 3.6.10 certificate dialogue. Finally we briefly described some additional attacks and specific countermeasures, we called ‘personal message-firewall’.

## REFERENCES

- [1] Microsoft, “Microsoft Developer Network (MSDN): Overview: Spy++,” [http://msdn.microsoft.com/en-us/library/aa242713\(v=vs.60\).aspx](http://msdn.microsoft.com/en-us/library/aa242713(v=vs.60).aspx), 2011.
- [2] C. Paget (alias Foon), “Exploiting design flaws in the Win32 API for privilege escalation. Or... Shatter Attacks – How to break Windows.” archived version on <http://web.archive.org/web/20060904080018/http://security.tombom.co.uk/shatter.html>, August 2002.
- [3] —, “Shatter attacks - more techniques, more detail, more juicy goodness.” archived version on <http://web.archive.org/web/20060830211709/security.tombom.co.uk/moreshatter.html>, Mai 2003.
- [4] T. Close, A. Karp, and M. Stiegler, “Shatter-proofing Windows,” archived on [http://www.blackhat.com/presentations/bh-usa-05/BH\\_US\\_05-Close/tylerclose\\_whitepaper\\_US05.pdf](http://www.blackhat.com/presentations/bh-usa-05/BH_US_05-Close/tylerclose_whitepaper_US05.pdf), 2005, (Whitepaper at Black Hat USA 2005).
- [5] Microsoft, “Microsoft Developer Network (MSDN): Windows and Messages,” [http://msdn.microsoft.com/en-us/library/ms632586\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms632586(v=VS.85).aspx), 2011.
- [6] —, “Microsoft Developer Network (MSDN): Hooks,” [http://msdn.microsoft.com/en-us/library/ms632589\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms632589(v=VS.85).aspx), 2011.
- [7] “Microsoft Developer Network (MSDN): WM\_NULL Message,” [http://msdn.microsoft.com/en-us/library/ms632637\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms632637(v=vs.85).aspx), 2011.
- [8] Microsoft, “Microsoft Developer Network (MSDN): Windows Vista Integrity Mechanism,” <http://msdn.microsoft.com/en-us/library/bb625964.aspx>, 2011.

## Following the Trail of Image Spam

Shruti Wakade, Robert Bruen, Kathy J. Liszka, and Chien-Chung Chan  
 Department of Computer Science, University of Akron,  
 Akron, Oh 44325-4003, USA  
 {liszka,chan}@uakron.edu

*Abstract- Image spam has evolved from simple images containing spam text to high quality photographic images. This paper explores the current trends in image spam by analyzing the contents of a corpus built over the past three years, courtesy of KnujOn. Statistics over a three year period show that spammers follow different patterns in sending spam which are based on various factors such as time of the year, holidays and politics. Other subjects appear to remain constant. Hate images have surfaced in the past year as well as malware-embedded images. Finally, we offer this corpus to others interested in this research area.*

*Keywords- spam, image spam, malware-embedded images, image scraping*

### 1 INTRODUCTION

Each morning, a daemon running on a server farm in Vermont, activates, zips up a folder of images from the prior day's haul on spam, and forwards it to a server in Akron, Ohio. These gems of information are stripped from emails collected by KnujOn ("no junk" spelled backwards) an anti-spam company [1]. Their mission is to fight against Internet threats, and specifically those delivered by email. They work with Internet governance bodies to help investigate abusive registrars and track cyber criminals. Part of the company's business is to allow users to upload their spam email and then process it, extracting hyperlinks and other information to help track the source of the message. In our case, images are stripped out and forwarded to us in an effort to build a sizeable corpus for the purpose of image spam research.

Spam is certainly not a new phenomenon. Filters work diligently to protect us, but it is a never ending battle. Once a mere annoyance, the motivation of spammers has become increasingly malicious. Originally these emails were primarily digital forms of paper junk mail. Scams quickly

followed to advertise inferior, fake, or non-existent products. Pharmaceutical email accounts for more than 70% of all spam [2]. Filled orders are often not the product advertised, if delivered at all. In any given spam message, a click on the embedded link will likely result in a drive-by download of malware, or result in a phishing attempt [3].



Figure 1. Image spam.

No form of digital communication is immune to these attacks. Social networks are becoming hotbeds of this malignant activity. Anecdotally, in February 2011, a search for "Viagra" on Twitter [4] delivered approximately 1 new posted tweet every 30 seconds, over a three hour period. That was just an observation of one spam keyword!

Email spam can be divided into two basic categories: 1) simple text and 2) an image embedded into the body of a message or sent as an attachment. Image spam, as a filter avoiding tactic, first appeared around 2000, but was noticed in 2005 when it comprised a mere 1% of spam emails. Within 18 months, images such as that in Figure 1, populated over 21% of all email messages [5]. It's a relatively effective mode of content delivery because morphing and other digital manipulations make it difficult for OCR readers to catch, and with minor changes, fingerprinting (ex., via MD5 hashes) is virtually

impossible. The Spammer's Compendium [6] is an excellent resource of information on different techniques spammers use to avoid detection. Use of animated gifs, picture tilting, picture waving, and image slicing are techniques described and demonstrated with real examples collected by volunteers.

In this paper, we focus on trends in image spam collected by KnujOn over the past three years. We have examined content of the images on an almost daily basis from April 2008 through January 2011. Some basic statistics are provided along with several observations. Finally, we delve into new techniques and trends in these types of images, namely scraped images and malware embedding.

## 2 TRENDS

Most e-mail filters check for text based spam but not image spam. Spam filters look for phrases or words related to spam, for example, Viagra, free, money, cash and so forth. When the message is included in an image an OCR needs to read the content to detect these keywords. This is time consuming for anti-spam software, yet easy for spammers who easily find new ways to defeat OCR filters by adding random noise to images, rotating contents, using multipart gif image formats, blurring the text, and adding colorful backgrounds [7]. One exception may be what one may consider more mainstream advertising, such as landscape lighting from Home Depot or Cisco routers.

Spammers have banked on image spam for over a decade. Today, most spam is generated automatically and dispersed with bots, thus the number of images generated is only limited by the computational competency of the bot-infected computers. Figure 2 shows a high quality image spam example where words can easily be picked off by an OCR filter. Figure 3 shows an example that could pass through these filters as legitimate email. Other image spam is simply photographs. Clearly pornography spammers have an agenda, but other photographs seem elusive in their intent, with pictures of sailboats in a harbor or a quiet city street.



Figure 2. Image spam potentially detectable by an OCR filter.



Figure 3. Image spam undetectable by an OCR filter.

Image spam is usually comprised of short text images, URLs and hyperlinks. The content can be broadly classified into following categories:

- Advertisement/Marketing (Rolex watches, outdoor lighting, office furniture)
- Pharmaceuticals (prescription as well as non-prescription)
- Pornography
- Financial
- Freebies, coupons, software
- Politically motivated

According to Computer World [8], image spam hit a peak in 2006-2007 with a dramatic decline at the end of 2008. However, with the shutdown of McColo, all spam declined for a period of weeks until the spammers re-established themselves with another host service. We've

certainly had no lack of data for research in image spam identification since the alliance with KnujOn. Table 1 shows statistics for the number of *unique* images collected per month from August 2008-2011. We used an MD5 checksum script to eliminate duplicate images. What we found is that the computer generated images were highly unique while many of the photo-quality images (mainly adult content) were eliminated because they were actually identical. These statistics are only with respect to the images that we have downloaded. We note that some days of the month, servers were down either in Vermont or Ohio due to maintenance or weather. Since this started out purely for the purpose of collecting enough images for testing an artificial neural network [9], we were not concerned with those missing days. We do feel, however, that the analysis can be extrapolated to the overall picture.

Month	2008	2009	2010	2011
Jan			3171	717
Feb		764	3451	781
Mar		2268	16403	
Apr		1008	18462	
May		1277	7337	
June		10863	18141	
July		7840	6725	
Aug	3660	12883	36003	
Sep	5527	13329	9105	
Oct	8021	8040	2233	
Nov	3525	5883	2601	
Dec	601	4119	943	

Table1. Unique image spam collected in 2008-2011

We manually checked images from January 2010 through February 2011 to see what trends are prevalent during the year. In order to get the trend we noted the most common type of spam in a month and then checked which of these appear across most of the months of a year. Figure 4 shows a graph describing the frequency of occurrence of 47 categorical trends in a year. The category for photo spam covers those photos that were pictures not fitting any category (the sailboat for example). We put pictures of seductive women in this category, when they did not fit clearly in the adult content category. As expected, the most common type of images are pharmaceutical, photo spam, and adult content (pornographic).

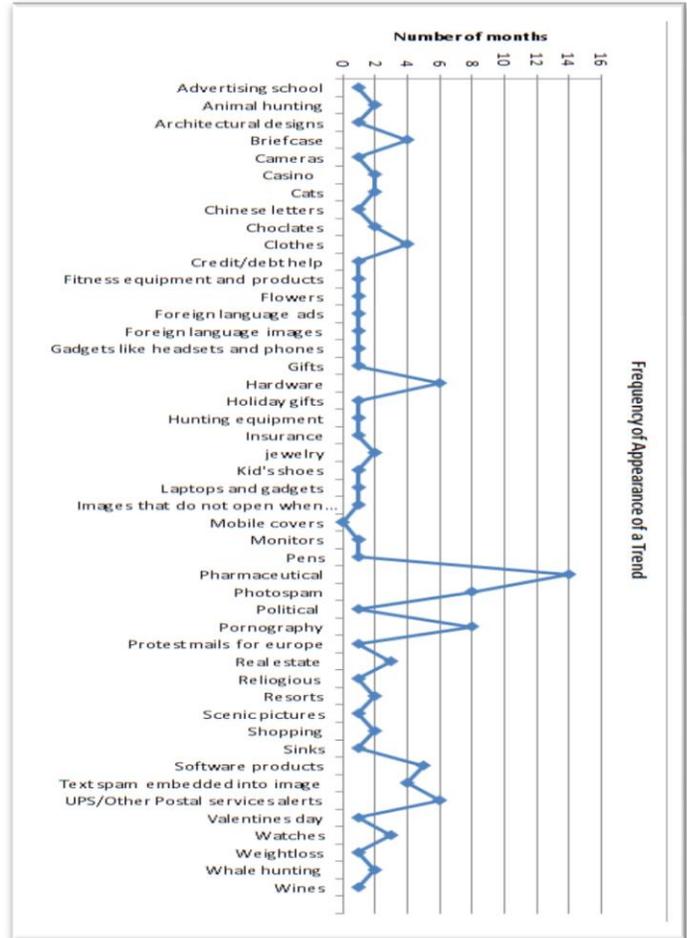


Figure 4. Frequency of trends

We observed that on special days of the year such as New Year's Day, Christmas or Valentine's Day, spam related to only this specific event was noticeably predominant. On 14-Feb-2011 we saw a large number of spam related to Valentine's Day, such as that seen in Figure 5.



Figure 5. Spam on Valentine's Day 14 Feb 2011.

Similarly the months of December and January contain images of chocolates, inexpensive gifts, coupons, clothes and such, all related to the holiday season of Christmas and New Years.

In early 2010, we saw our first animal cruelty pictures, and also our first “hate” images. These were particularly disturbing to look at. The politically motivated hate images first appeared in October 2010 and have continued through January 2011. To date, only a small number have appeared in the corpus, but the timing may not be coincidental to the unrest, protests, and ensuing violence that marked world events in January and February 2011.

### 3 SCRAPING IMAGES

Manual inspection of images shows that spammers have devised a new way to prevent inspection of image content by scraping the header part of the image. This renders the image unreadable by a file reader although it opens using a picture editor. The technique makes it possible to successfully convey the intended message to the user but prevents processing of images. Another technique to tamper with the format of the image is to include improper header information and/or incorrect color maps. File readers expect these be formatted properly and so, fail to read these spam images. Figure 6 is an example of such an image found in our corpus.



Figure 6. Scraped image spam.

### 4 MALWARE EMBEDDING

Embedding malware in files is not a new concept. It is used with MP3 files, video files, text documents and others. We found images during three months (May, September, and November 2010) with malware embedded in them. These were the first occurrences since we began downloading our corpus in April 2008. As yet, there is little literature available on how these images are actually used for malware embedding and how they attack their victims.

In general, when a non-executable file such as a jpeg containing an executable is double clicked, the non-executable file is opened by its associated application. You view the jpeg as a picture and nothing happens with the embedded malware. Another component must be present, such as a loader. This presumes that the host machine is already infected. The second step is for the loader to extract the code from the jpeg (or other image) and run it. Typically, this works because the component on the infected machine downloads the image from the web. In our case, these images came wrapped in spam, so we are certain that the complimentary portion of the malware infection was not present. We did perform some basic reverse engineering, and note that the loader was not present, thus not infecting our machines.

Recently Microsoft's Malware Protection Center discovered a variation of a malicious image which looks like a simple png file [10, 11]. Amazingly, the image displays instructions for the user to open it in MS Paint and then resave it as an hta file, which is an HTML application. Part of the image resembles random noise, but when the file is resaved according to directions, it decompresses into JavaScript. Now when this file is opened, the presumably malicious payload is executed. Without the user's willing participation, this is a lame attempt at spreading malware. The curious user, however, will most likely regret it. Figure 7 shows an example of one of these images logged by Microsoft [11]. Figure 8 shows the binary data of the image before and subsequent hta file [11].

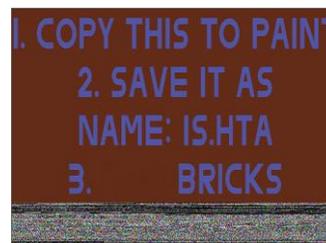


Figure 7. Image in .png form.



[http://www.beskerming.com/commentary/2010/08/12/527/Malware\\_in\\_Images,\\_a\\_Social\\_Engineering\\_Example](http://www.beskerming.com/commentary/2010/08/12/527/Malware_in_Images,_a_Social_Engineering_Example), last accessed February 2011.

11. Painting by Numbers, Microsoft Malware Protection Center, Threat Response and research blog, August 9, 2010, <http://blogs.technet.com/b/mmpc/archive/2010/08/09/painting-by-numbers.aspx>, last accessed February 2011.

# Architecting for Disaster Recovery – A Practitioner View

Octavian Paul ROTARU  
ACMS, Montréal, PQ, Canada

[Octavian.Rotaru@ACM.org](mailto:Octavian.Rotaru@ACM.org)

## Abstract

*Few businesses have the capability to effectively recover after a disaster. For vast majority of organizations, business continuity management activities are compromised by limited budgets and insufficient time and resources. A well-made contingency plan can save an organization from going out-of-business should an incident or disaster occurs.*

*This paper gives a practical perspective on disaster recovery plans and fault tolerant architectures. The intention behind the paper is to be an easy to read practical guide for disaster recovery practitioners. Practical advises, guidelines as well as tips and tricks, are presented, in an attempt to make Disaster Recovery Planning look less murky.*

**Keywords:** Business Continuity (BC), Business Resilience, Data Replication, Disaster Recovery (DR), Fault Tolerant Architectures.

## 1. Introduction

Most organizations today depend heavily on their IT infrastructure and their data in order to be able to provide service to their customers, but how many of them are really ready for a disaster scenario, either natural or man-made?

The Business Continuity Planning (BCP), Business Continuity Management (BCM), Testing and Execution are referred to collectively as Business Resiliency Planning and Business Continuity Management (BCM). This paper address only issues related to the IT infrastructure side of BCM, more specifically related to Disaster Recovery infrastructure and how well is it prepared for a real disaster scenario.

Cyber-infrastructure protection, business continuity and disaster recovery, includes safeguarding and ensuring the reliability and availability of key information assets, including personal information of citizens, consumers and employees. [2]

The existence of business risk observed with service disruptions is an inescapable concern for many organizations. Depending on the criticality of the data handled and service rendered, the business continuity approaches cover a wide range of options.

Disaster Recovery is a critical issue when it comes to information security and business resumption. Disaster recovery concerns a wide range of activities, from backing up data and retrieving it from backups, to repairing networking capabilities and rebuilding primary production sites. Disaster Recovery planning is the preparation for recovery from any disaster and its main aim is to help an organization become resilient after a disaster [4].

Susanto [11] considers IT to be the most important issues of all when discussing BC and DR, not only for being the foundation and backbone of the business but also because IT can play important roles in strategies development and improving efficiency of the whole BCP plan.

## 2. Fault-Tolerant Architectures and Cost

The basic system architecture that is being considered for disaster recovery consists of a primary and a backup site. The primary site is the one the handles the production functions, while the backup site is usually a stand-by location that can be used to run production functions if needed.

The backup location needs to store enough information so that if the primary location is unavailable, the information available at the backup site can be used to recover data lost at the primary and resume production activities.

The backup sites are classified into two main categories:

- Data Recovery Sites – Data is available at an alternate location, but service cannot be resumed until the primary site is back online.
- Service Recovery Sites – Both data and processing capabilities are available at the backup site and service can be resumed from the alternate location.

Both data recovery sites and service recovery sites require a way to synchronize or backup data, either on-line or at pre-defined time intervals.

On-line synchronization of data allows service to be resumed much faster from alternate locations. However, this approach incurs a higher synchronization cost.

Cloud computing also offers a good platform for disaster recovery. Cloud-based applications can be

accessed from any stand-by location, provided that the required communication lines are in place.

A healthy compromise between cost and business objectives is usually hard to achieve in a disaster scenario. However, there are ways to combine functions and save on costs while not compromising your goals:

#### **Distributed environments (Active – Active)**

A simple way to reduce cost is to distribute processing between multiple sites. In case one of the sites is affected by a disaster, the service loss is only related to the capacity of that site, while the service will still be rendered, even if at reduced capacity.

The cost of communication lines, remote clustering and data synchronization are the main drawbacks of distributed processing environments.

Data needs to be synchronized between the sites in order to allow distributed service processing. Also, load distribution mechanisms are required. Some companies distribute the load based on geographical areas while others distribute the load evenly between the centers, irrespective of the geographical origin of the request.

In case of even load distribution, one of the sites will need to provide front-end service and load balancing services. In case the front-end site is lost, a backup front-end site needs to be available to take over load balancing.

Regional processing centers do not require a load balancing service, but each regional site need to have a backup site available to take over at any given moment. Regional processing centers reduce the need for data replication. Each site can have a backup site or even two where to replicate stand-by data. Instead of replicating all data to all locations, you replicate only parts of the data (regions) to other regional centers that are ready to take over the load if needed.

#### **Use the processing power in a backup site for alternate purposes**

Another option is to use the processing power in your backup site in order to serve other business needs. For example, a DR site can be used for testing and development or any other functions that are not mission critical in a disaster scenario.

#### **Share the cost of disaster recovery**

Some organizations choose to share the cost of disaster recovery, by means of sharing resources. A common alternate site is usually setup. Data replication is done by all the partners to the alternate site and the site has enough processing power available to handle the processing needs of any of the organizations that are sharing the cost in a disaster scenario. The main assumption is that only one organization can use the

processing power of the alternate center of any given time.

### **3. Tips and Tricks for efficient DR planning**

#### **Clearly define your recovery goals**

One of the most challenging parts of disaster recovery planning is to define your recovery goals and get them approved by all the stakeholders.

Clearly defined disaster recovery goals are the barebone of a valid business resilient architecture and define its requirements.

Each organization needs to have well defined Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO) for its infrastructure. The recovery time objective defines how long the business can basically go without a specific application, function or service. The RTO is the maximum allowable outage time that the business can tolerate during a disaster scenario. The recovery point objective is the point in time to which you must recover data as defined by your organization. The recovery point objective defines the acceptable loss of data in a disaster situation.

RPO and RTO are independent parameters. RPO is more important than RTO if data availability is more important than service recovery. On the other side, if service recovery is critical, the service availability may overshadow the availability of data.

In any case, the prevalence of RTO over RPO and the other way around are extreme scenarios. For most organizations the requirements are somewhere in the middle, even if one of the recovery parameters has more importance than the other.

The RPO and RTO together define the guidelines of disaster recovery planning for an organization and they need to be in sync with the organization's mission statement and goals.

The RPO and RTO of any organization are translated into an architecture that has a price tag, and ultimately you end up comparing the price tag with the existing budget.

RTO and RPO definition is a very sensitive exercise. Defining goals that are either too ambitious or too low is something to be avoided. The goals need to be realistic in order to be able to translate them into a disaster recovery plan.

Many organizations define unrealistic goals that translate in plans that will fail if ever a disaster occurs because of the many assumptions that are made.

A general RTO and RPO definition for the entire environment is almost impossible to define in most of the organizations. Each business function or service has its own level of criticality.

For example, in case of a bank it is crucial to preserve the customer database and account balances, but the transactions history is not as equally important. Not knowing what the balances of the accounts are will result in revenue loss. A similar example is the telecommunications industry. You need to know the customer details and what each customer is due to pay, while displaying call details on the bill is only a nice to have.

Defining different goals for each set of data and service will help prioritizing their recovery in case of a disaster, and reduce the cost of your DR infrastructure.

#### **Filter mission critical data**

The amount of data that is being stored and processed is growing at a very high pace, and the question that an organization needs to answer is how much of it is really mission critical and needs to be preserved in case of a disaster.

Preserving all the data available is simply a nice to have and not a necessity for many organizations. Careful business impact analysis is required in order to identify the impact of losing information, define priorities and filter what is really critical.

Minimizing the volumes of data and business services that need to be preserved following a disaster is the main solution for minimizing the cost of DR.

Replicating and backing up only critical data will reduce the cost and in the same time simplify planning.

The law of parsimony applies perfectly to disaster recovery plans: the simplest of two or more competing solutions is to be preferred. A complicated disaster recovery plan that has too many variables and needs to much manual intervention is most of the times bound to fail. Key resources may not always be available in the aftermath of a natural disaster, and a complicated disaster recovery plan may be inapplicable because it assumes the availability of those key resources (human or material).

The best option you have is to keep the business continuity plan document to the absolutely bare minimum. Don't overcomplicate procedures and processes. Provide just simple information that the crisis management team can use as the basis of taking action and decisions. [1]

From a Disaster Recovery point of view, data can be classified in a few categories, each category requiring a different approach:

#### **- Temporary Data**

In most organizations there is no need to replicate temporary data to the alternate site. Some examples of temporary data are work files created by long-running batch processes and temporary files created by online transactions. Temporary data is not required in a Disaster Recovery scenario unless the long-running

jobs can resume from a point close to that where the primary site became unavailable.

Such a DR approach is needed for applications of extreme criticality only. Very few organizations have DR goals that are so ambitious and can also cover the cost of such architecture. Apart from data replication and processing power availability at the alternate site, a lock-step mechanism is also required.

Most of the times temporary data becomes unusable the moment the process that created it crashes and it will be re-extracted or re-created once the process reruns at the primary site or at the alternate site in case of DR. Due to its "perishable" characteristic, temporary data is ignored in most organizations while planning DR, and an assumption is made that all the processes interrupted by the disaster event will need to run again from the beginning when the alternate site will be up.

#### **- Raw Data**

Raw data is data that is being processed, and once processed it is not needed anymore. In certain industries the volumes of temporary data are extremely big and replicating them to DR will be very costly. Unprocessed raw data is sometimes needed in DR while in certain situations it can be regenerated. The decision to make raw data available in DR is most of the time influenced by legal requirements, and not by business decisions.

#### **- Replaceable Data**

Most of the organizations collect data that is not critical, but helps employees perform their jobs faster, or IT systems to run faster or more efficient. Such data can most of the times be re-generated or collected following a disaster event.

A good example of such data is database indexes. You need to have the data available, but indexes can be rebuilt. It will take a while to rebuild the indexes and database access will be slow during this time, but the information is redundant.

Most of the organizations can avoid replicating replaceable data and wait for it to be rebuilt after switching to the alternate site.

#### **- Mission-Critical Data**

Mission-Critical data is always replicated to the alternate site in one way or another. The success or failure of a disaster recovery depends on the ability to make mission-critical data available.

#### **Stay away from unrealistic assumptions**

*We just need to preserve the data. We will buy the servers (or any other equipment) required after the disaster occurs. We will install them and resume service very fast.*

Of course your vendors will no doubt provide the hardware or any equipment that you need, but how long will it take? And even if the equipment is

provided immediately, how long will it take to install and configure it?

Making such an assumption is dangerous because the availability of equipment in the aftermath of a natural disaster is usually limited. The natural disaster affecting a certain area may affect equipment vendors alike. The availability will be limited and many organizations may compete to provision any available piece of equipment.

Furthermore, installing equipment requires time and human resources with skills that may be hard to locate. Infrastructure projects take time to implement and assuming that they will be done in a very short time is not realistic.

Think about your last similar infrastructure project and its duration. Take that duration and multiply it with three, and you got yourself a very optimistic estimate of how long the same implementation will take during disaster recovery.

*We need to recover the service as soon as possible. We will reprocess the data while in parallel we will handle new incoming transactions.*

Processing old data in parallel with new transactions requires processing power that is usually not available in a DR scenario. Your backup environment needs to be strong enough to process incoming transactions as well as to catch up and reprocess the data that was lost. The reprocessing of data is usually a lengthy process that assumes resource availability.

*It will never happen to us*

One of the biggest problems of any disaster recovery architecture is cost. Making it cost-effective and proficient enough to be able to restore both data and service in a timely manner is a very complicated problem even for the best system architects.

Ostrich-like upper management sees disaster recovery plans as an expense and not as a necessity, assuming that it will never be needed. The “this will never happen to us” approach is both dangerous and counter-productive when dealing with disaster recovery plans and resilient system architectures.

Upper management support and firm commitment is a must for implementing resilient infrastructures, and managers that only pay lip service to disaster recovery planning are doing more harm than they imagine. Convincing management that the risk of a disaster is real is the biggest hurdle any DRP specialist must overcome.

The price tag of a disaster resilient infrastructure is the main problem for most organizations in today's economic stance and creativity is required in order to drive costs down and make the solution more attractive and easier to present to executives that think mainly in terms of \$.

### **Protect Personal Information**

Organizations that deal with personal information are in many countries subject to a strict set of rules. An Organization is responsible for protection of personal information and the fair handling of it at all times, even during a disaster recovery scenario. Care in collecting, using and disclosing personal information is essential to continued consumer confidence.

Canada is one of the countries that regulate how private sector organizations collect use and disclose personal information in the course of commercial business under the *Personal Information Protection and Electronic Documents Act* (PIPEDA) that became law in April 2000.

Each business is subject to the laws of the country where it operates. The reason to bring PIPEDA into this discussion is the ten principles of fair information practices developed under the auspices of the Canadian Standards Association [3]:

1. Accountability
2. Identifying purposes
3. Consent
4. Limiting collection
5. Limiting use, disclosure, and retention
6. Accuracy
7. Safeguards
8. Openness
9. Individual Access
10. Challenging Compliance

The ten principles of fair information practices listed above can constitute the backbone of a successful DR plan. Limiting collection will reduce the amount of data you need to safeguard and preserve accurate. Clear accountability and well identified purposes for collecting information helps identify the stakeholders and makes it easier to develop an efficient disaster recovery plan.

## **4. Information Assurance Techniques**

There are multiple ways to make sure that data is always available and can be accessed and used in case of a DR. Most of the information assurance techniques fall into two categories:

- Backup
- Data Replication

If restore time is not a problem, data backups to tape or virtual tape libraries (VTL) can be effective methods of data recovery.

Tapes or VTL backups (disk) can be used to restore data once a disaster occurs provided that enough storage is available at the alternate site and tapes (either physical or virtual) can be made available (recalled to site from the vault for physical tapes or available at the alternate site for virtual tape backups) in a timely manner.

Virtual tape libraries can replicate content at distance, allowing backups taken at the primary site to be replicated at remote locations and ready for recovery when needed.

Data replication is the process of sharing information in order to ensure consistency between redundant sources. The purpose of data replication is to improve reliability and fault-tolerance.

Data replication can be done in many ways and results differ. Ranging from live data replication methods to regular data copies, the data replication goals and techniques need to be in harmony with your DR goals.

Choosing between backup and data replication is usually driven by the recovery time objective. If your RTO is very tight, you cannot afford to wait for tape recovery to complete. Also, the quality of the tapes may influence the time to restore. Having multiple copies will mitigate the risk of a restore failing because of a bad tape, but having to run the restore once again is time consuming. Aggressive RTO goals imply data replication.

Once a decision is made between backup and data replication, the way to backup or replicate the data will be driven by the recovery point objective.

Aggressive RPO goals usually require live data replication. Live replication can be done in different ways, depending on the characteristics of the data that needs to be replicated.

Database systems can use transactional replication. All transactions running at the primary site can be replicated at the alternate site, either by using the redo logs (transfer them to the alternate site at pre-defined time intervals), or by running the same transaction simultaneously at different sites. Database replication usually imposes a master-slave relationship between the original and the replicas.

Disk storage replication is done by distributing updates of a block device to several physical disks located at different sites. Disk storage replication can be classified into two categories, depending on the way it is handling write operations: synchronous replication and asynchronous replication. Storage replication covers a wider range of applications and can be used for any kind of data.

Synchronous replication guarantees “zero data loss”. Atomic write operation either complete on both sites or not at all. The biggest disadvantage of synchronous replication is that the primary site will need to wait for the alternate site to confirm the write before proceeding further. As the distance between the sites grows larger, the delay introduced by the communication lines will impact the performance of the writes.

Asynchronous replication doesn't guarantee “zero data loss” but eliminates the performance penalty.

Atomic writes are considered completed as soon as the local storage acknowledges it. Data is replicated at pre-defined time intervals to the alternate site (with a small lag). In case of losing the local storage, the remote storage is not guaranteed to have the most current copy of data and information will be lost.

All remote data replication techniques require considerable bandwidth. Communication lines cost is substantial and becomes an on-going operational cost.

Most storage vendors offer data replication solutions, among which the most notable are EMC SRDF [5], NetApp SanpMirror [6], Hitachi TrueCopy [7], IBM Copy Services [8], HP Continuous Access [9], and FalconStor CDP [10].

Choosing between synchronous and asynchronous replication is usually done based on RPO. If your RPO is zero, the only available choice is synchronous data replication and the performance penalty cannot be avoided. However, if the RPO is greater than zero, an asynchronous data replication technique can be used and the acceptable replication lag will be driven by the defined RPO.

Semi-synchronous replication techniques are also available and provide a good compromise between synchronous and asynchronous methods. Performance penalty is also reduced. Atomic writes are acknowledged by the remote site as soon as received instead of when the write is completed.

## 5. DRP Testing

Testing is an essential part of disaster recovery planning. A plan that was never testing will probably never work in a real disaster scenario.

A new disaster recovery plan requires more frequent testing. After each test, the plan needs to be reviewed in order to make any necessary corrections. The changed procedures need to be retested and incorporated into the disaster recovery plan.

Disaster Recovery plans can be tested in several ways [13, 14]:

- *Structured Walk-Through Testing* – DR team members meet to verbally walk through specific steps of the plan, trying to identify gaps, bottlenecks and other weaknesses or confirm the effectiveness of the plan.
- *Checklist Testing* – ensures that the organization complies with the requirements of the DR plan.
- *Simulation Testing* – disaster scenario is being simulated so that the normal operations will not be impacted.
- *Parallel Testing* – testing is performed at the alternate site while production is not impacted

- Full-interruption Testing – A production systems are shut down and the disaster recovery plan is activated in a situation as real as possible. This is the best way to test your DRP plan, but it is costly and is disrupting the normal operations.

There will always be surprises during DR testing. Unexpected results will occur and alterations to the plans will be needed. The ultimate goal of testing the DR plan is to reduce the sources of error and make your DR plan as best as possible in order to avoid unpleasant surprises when the plan will be employed for real.

## 6. DRP Guidelines

1. *Check the legal requirements applicable for your organization.*

Legal requirements can highly influence the cost of your DR solution, and your DR solution needs to be harmonized with them.

2. *Make sure that all business processes are properly documented.*

You cannot protect what you don't know. All business processes, data inventory, data flows, and data classifications need to be available when DR planning is done.

3. *Classify your data.*

Data classification will help you decide on your DR strategy. Make sure that only what is really important will be available in DR. What you don't collect you don't have to pay to store and provide information assurance for.

4. *Define clear DR goals.*

Make sure that the business understands those goals and is in complete agreement with them. The best way to make the business decision makers understand DR recovery goals is to discuss with them scenarios. Start by taking a set of very specific DR goals and analyze what will be the business impact for it.

5. *Fine-Tune your DR goals*

Try to avoid a general set of DR goals that is meant to cover all types of data and services. Even if fine-tuned DR goals add to complexity, they reduce the cost of the DR solution.

6. *Create DR plans that meet your DR goals and choose the one you want to implement.*

You always have multiple ways to implement a DR solution, and each architecture has its advantages and disadvantages. My advice is to apply the law of

parsimony and chose the simplest one. A disaster scenario is not the time to test exotic technologies. Stick with what you know best.

7. *Include as much information as possible in your DR kit.*

More details than needed will probably do no harm. Missing critical information may make your DR plan fail or increase the time required for recovery. Include as much information as possible in your DR kit. Keep your plan as concise as possible and include additional information in annexes to make sure you have it at hand if needed.

8. *Don't try to achieve too much too soon.*

Try not to overstate your DR capability and readiness. Take time to test every function as soon as it is recovered. Diagnose problems early and do not leave testing for the end.

9. *Avoid making assumptions*

Yes, PBX and digital phone lines may not work in a real DR scenario as well as many other services – and this is only an example. Don't assume that services will be available and always prepare for the worst case.

10. *Avoid the easy way*

Recovering first the functions that you know are easy to recover is a temptation that needs to be avoided. Business functions need to be recovered in the order of importance and not depending on how is it is to recover them. If in a real DR scenario, always follow the plan and the priorities defined (same in case of DR plan testing).

11. *Check for opportunities to combine high availability and disaster recovery.*

High availability is a business requirement for many organizations. High availability architectures protect mission critical applications and services from hardware failure. The usual implementation is using stand-by hardware that is available at the primary site. Combining high availability and disaster recovery architecture can reduce the cost of both, by using the DR hardware available at the remote site for high availability failover in case of hardware failure.

Combining high availability and disaster recovery architecture is not always possible, and it needs to be carefully analyzed.

12. *Automate as much as possible.*

Automation can protect your solution from human errors. Limited human intervention can make your disaster recovery plan succeed even in situations when critical human resources are not available.

13. *Test your DR architecture and plans as often as possible*

Regular testing of your DR architecture and plans gives builds in it. Knowing that the plan was dress-rehearsed many times is the best assurance you can have.

“Quite simply, a plan which has not been tested cannot be assumed to work. Likewise, a plan documented, tested once and then filed away to await the day of need provides no more than a false sense of security.” [12]

14. *Keep your DR plans and architecture up to date.*

Make sure that any application change is analyzed and if needed reflected in DR. Identify as early as possible the impact on your DR of any change, no matter of its scope (new service or changes into existing ones). Reflecting changes in your DR plans and architecture requires budget and cost and impact needs to be well understood and communicated.

15. *Regularly review your DR goals.*

Business needs may change and DR goals review is often required. New business contexts require adjustments of the DR goals, triggering as a result changes in the DR architecture and plans.

## 7. Conclusions

Disaster recovery architecture and plans are driven by many factors. The number of variables involved is very high, budget being one of the most important, and the temptation of making unrealistic assumptions is very high. Proper disaster recovery planning and IT infrastructure ready to support it are crucial for survival of organizations that are facing disasters.

This paper provides recommendations for developing an effective disaster recovery plan and discusses the architectural options available, proposing a set of guidelines that can help practitioners create solid Disaster Recovery plans while avoiding common mistakes.

Finally, the only recommendation that I can make is to use your common sense and keep the solutions you choose as simple as possible. Simplicity never failed me in the design of fault tolerant architectures and it is the biggest lesson I learned.

## 8. References

- [1] David Honour, *Business Continuity on a Limited Budget*, The Business Continuity Institute.
- [2] Constatine Karbaliotis, *Critical Interests: Business Continuity, Disaster Recovery and Privacy*, Symantec, September 2009
- [3] Office of the Privacy Commissioner of Canada – *PIPEDA – A Guide for Businesses and Organizations – Your Privacy Responsibilities – Canada’s Personal Information Protection and Electronic Documents Act* (PIPEDA), Updated September 2009,
- [4] Philip Clark, *Contingency Planning and Strategies*, Proceedings of InfoSecCD 2010, October 2010.
- [5] Symmetrix SRDF Product Page, EMC, <http://www.emc.com/products/detail/software/srdf.htm>
- [6] NetApp SnapMirror Product Page, NetApp, <http://www.netapp.com/us/products/protection-software/snapmirror.html>
- [7] Products: Hitachi TrueCopy (R) Remote Replication, HDS, <http://www.hds.com/products/storage-software/truecopy-remote-replication.html>
- [8] Donald Chesarek, John Hulsey, Mary Lovelace, John Sing, *IBM System Storage FlashCopy Manager and PPRC Manager Overview*, IBM RedBooks paper, <http://www.redbooks.ibm.com/redpapers/pdfs/redp4065.pdf>
- [8] Nick Clayton, *Global Mirror Whitepaper*, IBM TechDocs, 2008, <http://www-03.ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP100642>
- [9] HP StorageWorks Continuous Access EVA, QuickSpecs, [http://h18000.www1.hp.com/products/quickspecs/11617\\_div/11617\\_div.PDF](http://h18000.www1.hp.com/products/quickspecs/11617_div/11617_div.PDF)
- [10] FalconStor *Continuous Data Protector (CDP) - Overview*, <http://falconstor.com/products/continuous-data-protector>
- [11] Lukman Susanto, *Business Continuity/Disaster Recovery Planning*, 2003, <http://www.susanto.id.au/papers/bcdrp10102003.asp>
- [12] U.S. Department of Commerce – National Bureau of Standards, *FIPS PUB 87 – Federal Information Processing Standards Publication, Guidelines for ADP Contingency Planning*, 1981 March 27.
- [13] Geoffery Wold, *Testing Disaster Recovery Plans*, Disaster Recovery Journal, Vol. 3, No. 3, p. 34.
- [14] Guy Witney Krockner, *Disaster Recovery Testing: Cycle the Plan, Plan the Cycle*, SANS Institute – InfoSec Reading Room, 2002.

# Optimized Edge Detection Algorithm for Face Recognition

M Sudarshan\*, P Ganga Mohan and Suryakanth V Gangashetty

Speech and Vision Lab,  
International Institute of Information Technology,  
Hyderabad, Andhrapradesh, India - 50032.

**Abstract** - Face recognition is one of the most challenging tasks in the field of image processing. This paper presents an optimized edge detection algorithm for the task of face recognition. In this method a gradient based filter using a wide convolution kernel is applied on the image to extract the edges. Later a thinning algorithm optimized for the wide convolution kernel is applied on the extracted edges. The advantages of this method over other gradient based methods is its ability to find the missing edges more successfully and boost the significant edges of the overall facial contour.

**Keywords** – Edge detection, face recognition, edge thinning, gradient processing, image processing.

## 1. Introduction

Computer based face recognition systems for security applications is a widely researched topic as facial features provide unique biometric identity for users. Face recognition systems are based on object recognition and tracking technologies. One of the important steps in object recognition is successful edge identification and extraction.

Several well known edge detection algorithms have been proposed in literature [1]. Some edge detection algorithms perform better than others depending on the type of object being recognized [2]. Several classes of edge detection algorithms exist based on the differentiation operator being used [3].

In this paper we propose a gradient based edge extraction algorithm suitable for identifying facial features. Generally gradient based algorithms have the advantage of being simple and able to identify edges along several orientations. Here we explore the effects of using a slightly wider convolution kernel for edge extraction. In the process we identify that the wide kernels are able to detect edges more accurately than smaller kernels. But wider kernels pose significant challenges in terms of edge thickness. Hence we also propose methods to overcome these constraints.

Compared to other gradient based algorithms we are able to detect the missing edges in facial features more accurately. Particularly in our method we boost the main facial features like lower nose, mouth, eye brows and the overall

facial contour and suppress the less significant edges due to falling hair, wrinkles etc..

In section 2 we provide a brief discussion of existing algorithms for edge extraction based on gradient processing. In Section 3 we explain in detail our proposed approach for edge extraction and subsequent thinning. In Section 4 we discuss the experimental results of applying our algorithm over several face images and provide a comparison with other gradient based edge detection algorithms.

## 2. Existing edge detectors

An edge can be defined as a significant change in local intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity. Edge detection algorithms can be broadly classified into following categories [4]:

- Gradient based edge detectors.
- Laplacian edge detectors. (Second derivative)
- Gaussian edge detectors. (Laplacian of Gaussian [5])
- Colored edge detectors.

Gradient based edge detection algorithms use directional first derivative operation. The advantages of gradient based edge detection systems are they are simple and are able to detect edges along several orientations. But the disadvantages are their sensitivity to noise.

Several well known gradient based edge detection algorithms exists like Sobel's, Robert's, Prewitt and Kirsch [6]. The main difference in these edge detection algorithms is the type of convolution kernel being used.

Sobel's edge detection uses a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. This is performed using a combination of horizontal and vertical directional convolution kernels as given in Table 1. It performs a 2-D spatial gradient measurement on an image and emphasizes regions of high spatial frequency corresponding to edges. It is used to find the approximate gradient value at each point in the gray scale image. In order to suppress noise, the weights are assigned such that it is more

concentrated in the centre.

Robert's edge detection filter is similar to the Sobel's edge detection method but the difference is Robert's system uses diagonal convolution operators as given in Table 2. Robert's filter is more effective in detecting diagonal edges in a given image.

Prewitt edge filter is based on sharp intensity transitions in the image and assumption of low Poisson type noise in the image. In this method the image is convolved with different set of convolution kernels which are sensitive to edges in different orientations. At each pixel the gradient magnitude with maximum response of all convolution kernels is chosen. The edge magnitude and orientation of a pixel is then determined by the template that matches the local area of the pixel best. Prewitt's edge detection method exhibits better performance under noisy conditions.

### 3. Proposed design for optimized edge detection

Broadly the proposed edge detection system can be separated into three steps: pre-processing of the image data for noise smoothing, edge extraction based on modified convolution kernel and edge thinning using optimized algorithm.

#### 3.1. Pre-processing

Pre-processing is performed for image smoothing, which is achieved by applying a low pass Gaussian filter. The energy of a typical image is concentrated in its low frequency components. Energy of some form of noises such as wide band random noise is typically more spread out in the frequency domain [7]. Low pass filtering helps in reducing such forms of noise and gives a smoother image. The Gaussian filter used can be represented as,

$$h(x, y) = \exp -(x^2 + y^2) / (2\pi\sigma^2)$$

where, the standard deviation 'σ' determines the cut-off frequency of the filter. A smooth h(x, y) preserves the original shape in the image and is less likely to distort the image.

#### 3.2. Edge extraction using modified convolution kernel

For edge extraction we follow a form of gradient based method. In gradient based models edges are calculated based on the locations where first derivative crosses a certain threshold.

In our model we use a non-directional edge

detector function for calculating edges. This non-directional edge detector function is based on root mean squared of the horizontal and vertical edge detectors, obtained after applying modified wide convolution kernel based on Sobel's operator.

After applying several directional convolution kernels we identified that the combination of a horizontal and vertical convolution kernel is best suitable for facial edge feature extraction.

Let F(X,Y) represent the final image obtained after preprocessing in previous stage. Let us represent the non-directional edge detector function used in this method by G(X,Y). G(X,Y) can be given as,

$$G(X,Y) = [I_x(X,Y)^2 + I_y(X,Y)^2]^{1/2}$$

where,  $I_x(X,Y)$  and  $I_y(X,Y)$  represent the partial derivative of F(X,Y) with respect to X and Y.  $I_x(X,Y)$  is estimated as,

$$I_x(X,Y) = \begin{aligned} &2[f(x+1, y+1) - f(x-1, y+1)] / 4T + \\ &[f(x+2, y+1) - f(x-2, y+1)] / 4T + \\ &3[f(x+1, y) - f(x-1, y)] / 6T + \\ &2[f(x+2, y) - f(x-2, y)] / 8T + \\ &2[f(x+1, y-1) - f(x-1, y-1)] / 4T + \\ &[f(x+2, y-1) - f(x-2, y-1)] / 4T \end{aligned}$$

Similarly  $I_y(X,Y)$  is also estimated with respect to Y. In the above equation, the scaling factor 1/nT is omitted, since the computed derivatives are later compared with a threshold.  $I_y(X,Y)$  is also represented as,

$$I_y(X,Y) = [I_x(X,Y)]^T$$

The convolution kernel for  $I_x(X,Y)$  is given in Table 1 and  $I_y(X,Y)$  is given in Table 2.

Table 1: Modified convolution kernel for vertical edge detection  $I_x(X,Y)$

-1	-2	0	2	1
-2	-3	0	3	2
-1	-2	0	2	1

Table 2: Modified convolution kernel for horizontal edge detection  $I_y(X,Y)$

-1	-2	-1
-2	-3	-2
0	0	0
2	3	2
1	2	1

Once  $G(X,Y)$  is calculated it has to be compared with a suitable threshold value. One of the outcomes of applying gradient based techniques is thick edges. In our case as we are using a wider kernel it further enhances the edges. But the advantage of using a wider kernel here is, it boosts the significant edges relative to the main contour of the overall face, eyes, lower nose. Other minor lines due to noise like hair and wrinkles are suppressed. Another significant advantage of a wider kernel is it finds the missing edges in the face more accurately as illustrated in the experimental results later.

To deal with the issue of thick edges the threshold used for edge detection has to be chosen appropriately. In our case we chose a threshold which is optimized for face recognition based on several trails.

$$\text{Threshold} = [1.10 * \text{Mean}(G(X,Y))]^{1/2}$$

### 3.3. Successive thinning of edges

After extracting edges it is important to apply a thinning algorithm which is suitable for wide kernel. Classical thinning algorithms for binary images consist of applying a 3x3 pixel window throughout the image and extracting the points which meet thinning criteria.

In our case, we use a modified 5x5 pixel window for estimating the edges and while applying the thinning criteria we consider  $f(x \pm 2, y \pm 2)$  pixels. For example to calculate the vertical edge, the thinning criteria applied is,

$$\{f(x, y) > f(x-1, y) \parallel f(x, y) > f(x-2, y)\} \\ \&\& \\ \{f(x, y) < f(x+1, y) \parallel f(x, y) < f(x+2, y)\}$$

This overcomes the effects of applying a 5x5 wider kernel in previous step. Thus we obtain a finer image which shows the significant contours of the face more clearly.

## 4. Experimental results

The optimized edge detection algorithm was applied over several faces from Face Pix database and the results were analyzed. After analyzing the images it is observed that we are able to identify the broken/missing edges in the resultant image more clearly in our method compared to other gradient based methods like Sobel's or Robert's.

The effect of applying different gradient based edge detection algorithm is illustrated below:

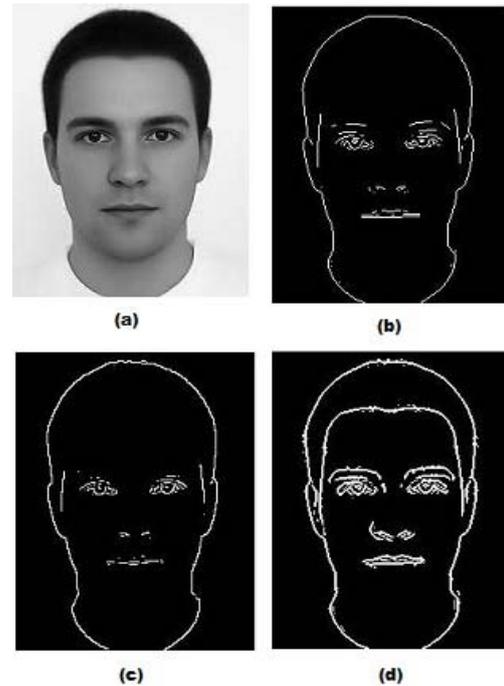


Fig. 1: Effect of applying different gradient based edge detection algorithms on a sample image. (a) Original Image (b) Effect of applying Sobel's filter. (c) Effect of applying Robert's filter. (d) Effect of applying Optimized edge detection algorithm.

In Fig. 1 it can be observed that compared to Robert's filter, the Sobel's filter is able to detect the facial contours more clearly. For e.g. eye brows are extracted more clearly in Sobel's filter. But compared to Robert's and Sobel's methods, our optimized edge detection algorithm identifies and highlights the facial contours more clearly. Our method captures the missing upper edges of the facial contours more clearly. The other prominent features like eyes, eye brows, nose and mouth are also clearly highlighted.

Effect of applying different gradient based algorithms on another sample image is illustrated in Fig. 2 as shown below.

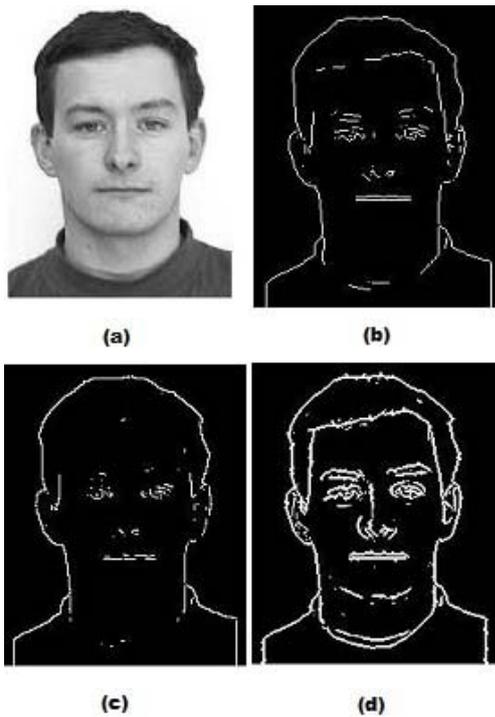


Fig. 2: Effect of applying different gradient based edge detection algorithm on another sample image. (a) Original Image (b) Effect of applying Sobel's filter (c) Effect of applying Robert's Filter (d) Effect of applying modified edge detection algorithm.

By observing the images in Fig. 2 we can find that our method identifies the broken and missing edges more clearly particularly in the chin region. But we can also notice some noise in the image near hair boundary. This is the residual effect of using a wider kernel. This noise sometimes exists even after applying the thinning algorithms. This is one of the drawbacks of using a wider kernel. But with superior noise reduction algorithms in post processing stage we should be able to suppress this noise.

## 5. Conclusion

This paper proposes an optimized edge detection algorithm suitable for the face recognition task. The main idea of the proposed method is to boost the significant edges and then apply successive thinning algorithms. The two advantages of this method over other gradient based systems is its ability to find missing and broken edges more accurately and suppress the less significant edges. Possible future work is to enhance the thinning algorithm such that it is able to suppress the noise in post processing stage more effectively.

## 6. References

- [1] A. Rosenfeld and A. C. Kak, Digital Image Processing, New York: Academic Press, 1982.
- [2] J. Canny: A Computational Approach to Edge Detection, IEEE Tr. PAHI-8, 679-698, November 1986.
- [3] N. Senthilkumaran and R. Rajesh, "Edge Detection Techniques for Image Segmentation - A Survey", Proceedings of the International Conference on Managing Next Generation Software Applications (MNGSA-08), 2008, pp.749-760.
- [4] H.Chidiac, D.Ziou, "Classification of Image Edges", Vision Interface'99, Troise-Rivieres, Canada, 1999. pp. 17-24.
- [5] D. Marr and E. Hildreth, Theory of edge detection", Proc. Royal Society, London, 1980, pp.187-217.
- [6] E. Argyle. "Techniques for edge detection," Proc. IEEE, vol. 59, pp. 285-286, 1971.
- [7] Jae S. Lim, Two-Dimensional Signal and Image Processing, Prentice-Hall, Inc. 1990.

# Risk Management in Healthcare Services

Montri Wiboonrat  
 College of Graduate Study in  
 Management, Khon Kaen University,  
 Bangkok Campus  
 25 Bangkok Insurance Building  
 25<sup>th</sup> floor, South Sathon, Sathon Rd.,  
 Bangkok 10120, Thailand  
 Tel +66 2 677 4140-3

montwi@kku.ac.th,  
 mwiboonrat@gmail.com

## ABSTRACT

The contribution of this research purposed a transition process and model of e-Healthcare services, and created knowledge based from integrated process between public and private partnership for risk transition management in e-Healthcare services. The process, model, and knowledge based are discussed by using several cases of healthcare services transition projects in Thailand to develop and organize transition plan. Risk management and project management are deployed during transition process to reduce project failure as the same time increase possibility of project success. The research objectives of healthcare transition are designed for system efficiency and effectiveness subject to; improved healthcare quality, increased accuracy and traceability of treatment process, reduce operation costs, compliant international standards, and facilitated treatment information for medical and clinical team. The research findings initiate a standardized pattern of risk transition process and model for healthcare services.

## Keywords

e-Healthcare Services, Healthcare Transition, Risk Transition Management.

## 1. INTRODUCTION

In the 21<sup>st</sup> century, it is a generation of technological change. Evaluation in telecommunication technologies drives extremely the growth rate of an internet since year 2000. Social networks, Nano technologies, Wireless Communications, RFIDs, and life expectancies are all apprences (outcomes) of the change prevalent in neo society. The present business consideration is from productivity improvement to quality centric, a change from local competition to global competition, and overall transition from a manufacturing ecosystem to a service or information ecosystem. In the future, digital services are inevitable to increase competitive advantage. It is a huge gap of transition from traditional business to e-business and e-commerce.

Business transition is essential and inevitable, but it is also jeopardized. However, all businesses need to reform and transform for creating a better position and competitive advantage. Transition courses uncertainties and risks which in turn conducted pressure. Therefore, transition is stressful, unpredictable, and risky. It is unforeseen in its outcome and effects on the organization reform.

This research is addressed on system integration beyond the project management, i.e. between the risk transition project and the organization's vision. This interrelationship is instituted a

hierarchy of risk levels that need attention and enterprise risk management. It requires coordination and synchronization of the entire activities. It is not just done in isolated areas. From the project landscape, the natural interface upward is in the program structure that has its own program risk management. Risk management has a special mechanism to transform a transition project because transition project are particularly risky. Risks shall be employed at the outset of the project's objectives and constraints. Normally, there are transformation to project-level risks and overall project risks. Researcher purposed a pragmatic system approach to transform risk management within formal process identified typically in standards and methodologies. There are great definitions of how to perform about preparing a risk manage plan. To make Management transition project be successful, is a problem. Neo and better strategic options are required continually to perform, but former and repeating failures won't disappear. Perchance the legacy focuses on only one method that is inefficiency to improve challenge.

We all have accepted that each project is for some unique purpose, unique problem, unique constraint, and reason to happen. Moreover, they have their own way to success or failure. There are many researchers' articles that demonstrate the projects are often failed to meet their objectives and expectations such as, on budget, on schedule target, and provide intended benefits. Even though, there are several solutions to solve those problems, but the concept of "one solution does not fit all problems" is always applying project by project.

## 2. HEALTHCARE MANAGEMENT

Silo operations is a traditional day life of healthcare operations system which obstructs and constraints a healthcare development in the 21<sup>st</sup> century. Resistance from traditional culture creates misinterpretation objectives of healthcare transition project. We need to believe in a change and transition that will create more benefits to stakeholders in molt dimension payoffs. In a hierarchy of project objectives, strategic implementation objectives will be technically different from the operations ones. Transitional objectives do concern about the entire project lifecycle from requirements and expectations after that interprets to conceptual design, transforms conceptual design, implementation, and compliant international and national standards. Moreover, the process is constructed by a sustained business development. However, during transition process, we cannot avoid risks and uncertainties situation.

All transition processes and methods have their own risks, but how to authentically recognize the root cause of risk. It is

important for organization transform and reform process. In 21<sup>st</sup> century risk analytical skill requires for all organization development to sustainability for creating competitive advantage. During transition in the second millennium, information technology reforms an organization as core business operations which malt down with business strategic management. One can say that business and IT vision shall be the same paradigm not paradox interrelationship.

The objectives of healthcare transition are designed for system efficiency and effectiveness which defines as a probability that a system can successfully meet an overall operational demand within a given time when operated under specified conditions or the ability of a system to do the intention for which in was committed. Many technologies attempt to fall shorter than expectations or do not sufficiently deal with the complexity of the solution required. Furthermore, many of the complementary tools and technologies need transition method that are often lacking of have not been adequately performed. All these factors can drive tendentious risks during implementation and may discourage their use.

Emile [1] defined a system transition as a structural change among technologies, procedures, and ecosystems. Transition management performs as primary object to manage metamorphoses towards sustainability. Transitions can be described as “gradual continuous processes of change where the structural character of a society or complex sub-system of society transforms”. Transition management can be classified into the following characteristics [2]:

- long-term thinking for framing short-term policy;
- multi-domain, multi-actor, multi-tier;
- focusing on learning;
- aligning system innovation and system improvement;
- keeping a large number of options open.

They are two conceptual approaches of how transitions materialize. One, literature on transitions utilizes three analytical and heuristic tiers for system innovations. The micro-tier contains unique technologies, in which neo technologies can come into maturity and be developed. The meso-tier embraces a group-work of procedures in a dynamic equilibrium. The macro-tier grasps technical ecosystem landscapes, with global and natural system development. In this formalization stage, transitions transpire when rejuvenations on the micro-tier evolves and is taken up to modify the group-work of procedures and eventually transforms the landscape on the macro-tier [3].

Other, four transition stages are described in the pathway of transformation, as illustrated in Figure 1.

A stage of predevelopment depicted in (1) that one of dynamic equilibrium I. In the take-off stage (2) changing starts to transpire. During the breakthrough stage (3) obvious structural changes have effect. A transition ends with a stabilization stage (4), where the speed of transforming decreases and a new dynamic equilibrium II is accomplished. There are three system indicators are identified; the time period of a transition; the speed of a transition; and the size of the change [4].

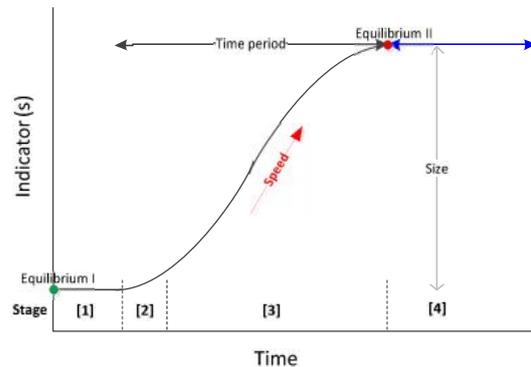


Figure 1. Stages and indicators in a transition procedure.

### 3. METHOD AND SUBJECT OF STUDY

Researchers defined a public healthcare system transition as a structural change in three dimensions; technical systems; innovation processes; and social-impact subsystems.

At policy management level; five organizations have been interviewed: Ministry of Public Health (MOPH); National Health Security Office (NHSO); The Institute of Hospital Quality Improvement & Accreditation (HAI); National Health Commission Office (NHCO); and Health Systems Research Institute (HSRI). Mainly the questions asked for management are concentrated on “how to improve healthcare services by applying IT.” This research was conducted through two interviewees of each organization.

At practitioner level; 4 public and 4 private hospitals have been observed and investigated on existing processes, IT technologies, and patients’ satisfaction levels by direct interviews and questionnaires. Each group of hospital was selected; 5 doctors; 5 nurses; 10 admin staff’s support; and 20 patients at hotspot. Impact level of hospital performs transition to e-Healthcare purposes to perform mathematical model in term of quantitative analysis. Healthcare transition management model (HTMM) was designed and constructed a guideline as a stepping stone for MOPH to consider as standardization for applying to deliver a basic e-Healthcare of public healthcare services in Thailand.

### 4. HEALTHCARE TRANSITION MODEL

Present’s health-care businesses operate in dynamic environments that they need continuous adapts to change as patient’s expected level. The medical and clinical treatment trends are reduced time treatment and operation costs but increased healthcare quality, and changing in improved healthcare standards and legislations. Health organizations are forced from outside rapid environmental changes such as technological innovation, deregulation, competition, patient’s growth rate, and scarcity of medical resources. The consequence impact of not adapting rapidly transition to these changes could create penalty of lost market share, financial difficulty, or entire completed failure in competitive business. The dissipative model (DM) links rapidly moving the system into a highly unstable state or equilibrium II, as seen in Figure 1, and involves huge change over a short period of time [5]. This equilibrium II should be sufficiently different from the old state or equilibrium I. This DM is forced to change to desired future state. Conversely, logical incrementalism (LI)

encourages that change should be accomplished slowly and in small transforms stages. The more conservative approach for healthcare transition model may be appropriated to use LI. Advantage and disadvantage of LI and DM model are compared as illustrated in Table 1. Since healthcare transition involves many health systems: personal health management; healthcare delivery; public health; and researches [6]. The primary choice between DM and LI seems to be based on risks which are in turn usually related to transition period, source of funding, and maturity technologies. As the same time, risk management: ISO 31000 [7] and project management: IEEE std. 1490-2003 [8], are designed and planning for helping and securing healthcare transition success.

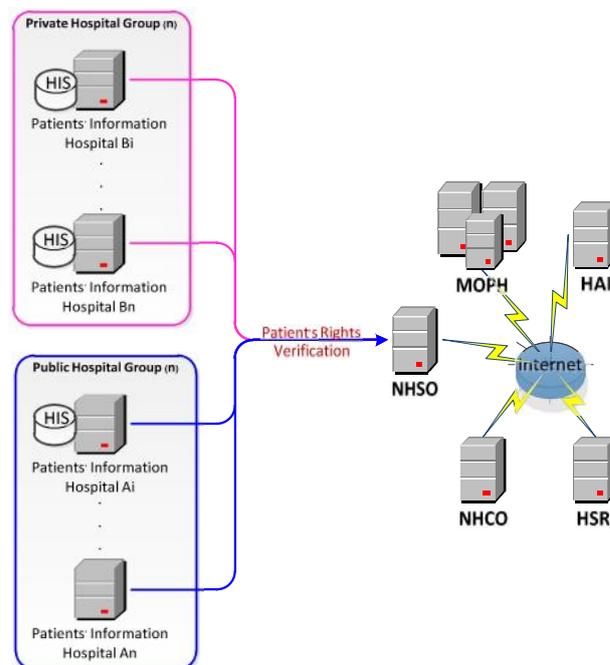
**Table 1. Advantage and Disadvantage of Logical Incrementalism and Dissipation Models [5]**

METHOD	ADVANTAGES	DISADVANTAGES
<b>Logical Incrementalism Model</b>	<ul style="list-style-type: none"> <li>* Useful with a long lifetime project</li> <li>* Promotes cohesion, identify, morale, and consensus</li> <li>* Allows for easy modification as the program matures</li> <li>* Perhaps the safest if time allows</li> </ul>	<ul style="list-style-type: none"> <li>* Slow and may not be responsive enough to rapidly changing environments</li> <li>* Lack of a clear goal may raise anxiety levels</li> <li>* May induce management control problems due to uncertainty</li> <li>* May not be possible because of a catalyst of time constraints</li> <li>* Slow speed of change may increase tension</li> <li>* Lower levels may view the slow change as an indication of management insecurity, hesitancy, or timidity</li> </ul>
<b>Dissipative Model</b>	<ul style="list-style-type: none"> <li>* Allows the organization to change rapidly</li> <li>* Commitment to the new state is increased</li> <li>* A clear signal of corporate policy in given</li> </ul>	<ul style="list-style-type: none"> <li>* The consequence of an unsuccessful transition effort could be a chaotic and disorganized system which could collapse</li> <li>* Bureaucratic, hierarchical organizations may have difficulty using DM</li> <li>* The pace of change is not conducive to monitor or modify of the transition</li> <li>* High amounts of stress are created by rapid change</li> </ul>

### 4.1 Investigation on Existing Healthcare Systems

The investigation by interviewing and observing with 4 public and 4 private hospitals, the result shown that each IT system of each hospital has done as ad-hoc system, they are different from each other subject to: strategic policy, IT facility infrastructure, hospital information system (HIS) applications, type of medical clinical data records, process of data record and storage, system operations, and system maintenance. Since, Ministry of Public Health (MOPH); National Health Security Office (NHSO); The Institute of Hospital Quality Improvement & Accreditation (HAI); National Health Commission Office (NHCO); and Health Systems Research Institute (HSRI), do not have standards and regulations to force them to do. Moreover, some of subjects (selected hospital) still use manual cards for recording their

patient's information. The latest update information demonstrates that only patient's rights verification can do as digital transaction through NHSO from all public and private hospitals for verifying their insurance/social healthcare policy, as depicted in Figure 2.



**Figure 2. Existing health information infrastructure.**

### 4.2 Interview Results

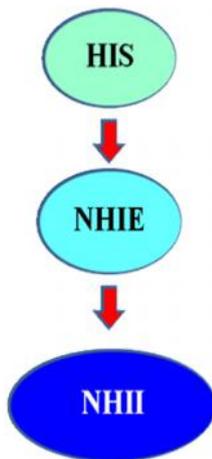
The subject's interview results represented a sample of hospital's population. Research may classify risks into 3 stages: policy risks; implementation risks; and operational risks. Policy risks are involved unpredictable of political climate that direct impacts through policy maker. Healthcare direction always changes when the new government changed. Mostly, head of MOPH is depended on politician. It is no continuous on long term healthcare direction since government team always changed. Moreover, policy also relates to source and amount of funding that supports from national healthcare activities and projects. It has created the phenomena of "struck in the middle of nowhere policy." The report from MOPH (2010) demonstrated that in year 2009, 505 hospitals in Thailand have problems with deficit cash flow management and 175 hospitals have problem with liquidity. The statement declares that this problem came from government insurance policy. Since operations costs in public healthcare increased by 30-40% but central government give a free public healthcare to 47 million citizens with the same operations costs from last year. It showed crisis in public healthcare management.

Implementation risks, 50% of projects fail before/during implementation because they are not related to the new policy maker ideas, 30% of projects fail because of specifications and requirements changed. 20% of projects fail due to cannot deliver as term of requirements (TOR), technical problems, delay by suppliers, and testing and commissioning problems.

Operational risks, it is divided to two groups: soft system (human activities) and hard system (equipment operations). 80% is under human activities such as no plan to fail: misunderstanding processes (interpretation), ignore instructions (attention), tired, feeling and emotion conditions (time); fail to plan: upgrade error, malfunction during testing system, miss version, and time to leave. 20% is failed under systems, equipments, operations condition, and nature disaster for example meantime between failures (MTBF), meantime to repair (MTTR), flooding, earthquakes, etc.

### 4.3 Construction Transition Model

It is time to move forward by leaving old/legacy healthcare operations management and transform to digital operations management. Technology in 21<sup>st</sup> century helps reduce operations costs and time, improve operations efficiency and effectiveness, increase healthcare quality, create healthcare standards, and save more life. Nature of public healthcare management involves with many parties, therefore transformation may take long time from party to party or from collaboration to integration. With this reason LI model is the suitable model for applying during healthcare transition. The healthcare transition needs cooperation from all parties: private and public hospitals, healthcare government agency, and medical schools. The transition shall start from MOPH vision which needs to declare healthcare transition project as national policy and must doing as standards and regulations for all parties that involved. The healthcare transition management model (HTMM) is comprised of 3 stages: hospital information system (HIS), national healthcare information exchange (NHIE), and national healthcare information infrastructure (NHII), as illustrated in Figure 3.



**Figure 3. Healthcare transition management model (HTMM).** The starting transition shall be from the smallest or based of healthcare structure which is hospital. First step: define standard of HIS must take into action. Since HIS is control all hospital information activities and it is the first stage of patient information input and execute as electronic medical record (EMR). Second step: define standard of NHIE must be complying by all parties, as demonstrated in Figure 4. It is the second stage that transforms all EMRs to portable data and information. This NHIE is designed to support all nationwide information requests from any hospital as concept of anytime and anyplace and anywhere. Before this

concept fulfill the last stage must be completed first, that is NHII. NHII describes as backbone networking communication system of national healthcare services. This healthcare transition will change the way of traditional healthcare service to become e-Healthcare services. It links and transports all EMR from hospitals through international information requestors such as when patient travel to aboard and get sick the host hospital in aboard can request and retrieve patient's information for diagnosis and analysis before making decision for treatment. It saves time and reduces more duplicate processes for doctor to making decision more accuracy as a result to save more life.

However, healthcare transition needs more resources to support such as funding, expertise from all healthcare segments, IT specialists, maturity technologies, and public-private collaboration. Therefore, central government must be taken as a host of this project because of huge investment and long term project that relates to high risk of policy, funding, regulations, and technologies.

### 5. CONCLUSION

The result findings demonstrated the system transition from the predevelopment or pre-design, take-off or change starts occur, breakthrough or visible structural changes, of healthcare transition management model (HTMM) till new dynamic equilibrium point the resistance of stakeholders reducing dramatically, from phase by phase. For the design phase of system transition this might be even more problematic, because not only the political system but also the transition process is dependent on objectives and constraints. Integration and synchronization of information, system integration, and requirements and expectations among patients, IT experts, and medical teams is the key to success of transition mechanism from legacy healthcare services to e-Healthcare services. Researcher believes that proper design and planning of e-Healthcare reform is necessity to accomplish a hospital accreditation (HA) requirements. These require new medical and clinical policies, regulations, organization development, maturity technologies, support funds, and neo vision of collaboration and integration strategies. To provide a better support for medical and clinical treatment, decision support system (DSS) shall be integrated and synchronized throughout national healthcare information systems which will underpin transition management for sustainability.

### 6. ACKNOWLEDGMENTS

This research is granted by 40<sup>th</sup> years of research funds of Khon Kaen University of academic year 2010. It is under the New Researcher Development Project. This research is a part of the main topic research on "Business Risk Management in Hospital Information Technology Project: Thailand's Hospitals Case Studies."

### 7. REFERENCES

- [1] Emile, C.J.L. and Gerard, D.P.J. 2008. On the design of system transitions: Is transition management in the energy domain feasible?. In *IEEE International Engineering Management Conference, IEMC-EUROPE*, (June 28 to 30, Estoril, Portugal, 2008), 193-197.
- [2] Rotmans, J., Kemp, R. and Van Asselt, M. 2001. More evolution than revolution: Transition management in public policy. *Foresight*, vol. 3, (2001), 15-31.

- [3] Geels, F. W. 2005. Processes and patterns in transitions and system innovations: Refining the co-evolutionary multi-level perspective. *Technological Forecasting & Social Change*, vol. 72, (2005), 681-696.
- [4] Wiboonrat, M. 2010. An empirical investigation of transition management in public healthcare to e-Medicare: A case study of Thailand. *The Seventh International Conference on eLearning for Knowledge-Based Society*, (Thailand, December, 16-17, 2010).
- [5] Hunsucker, J.L., Law, J.S, and Sitton, R.W. 1988. Transition management- A structure perspective, *IEEE Transactions on Engineering Management*, Vol. 35, No. 3, (August, 1988), 158-166.
- [6] Detmer, D.E. 2010. Building the national health information infrastructure for personal health, health care services, public health, and research. Department of Health Evaluation Science, University of Virginia, USA.
- [7] ISO 31000, 2009. International Standard ISO/FDIS 31000, Risk Management-Principles and Guidelines, (July, 25, 2009).
- [8] IEEE 1490, 2003. IEEE Guide Adoption of PMI Standard- A Guide to the Project Management Body of Knowledge, (December, 10, 2003).

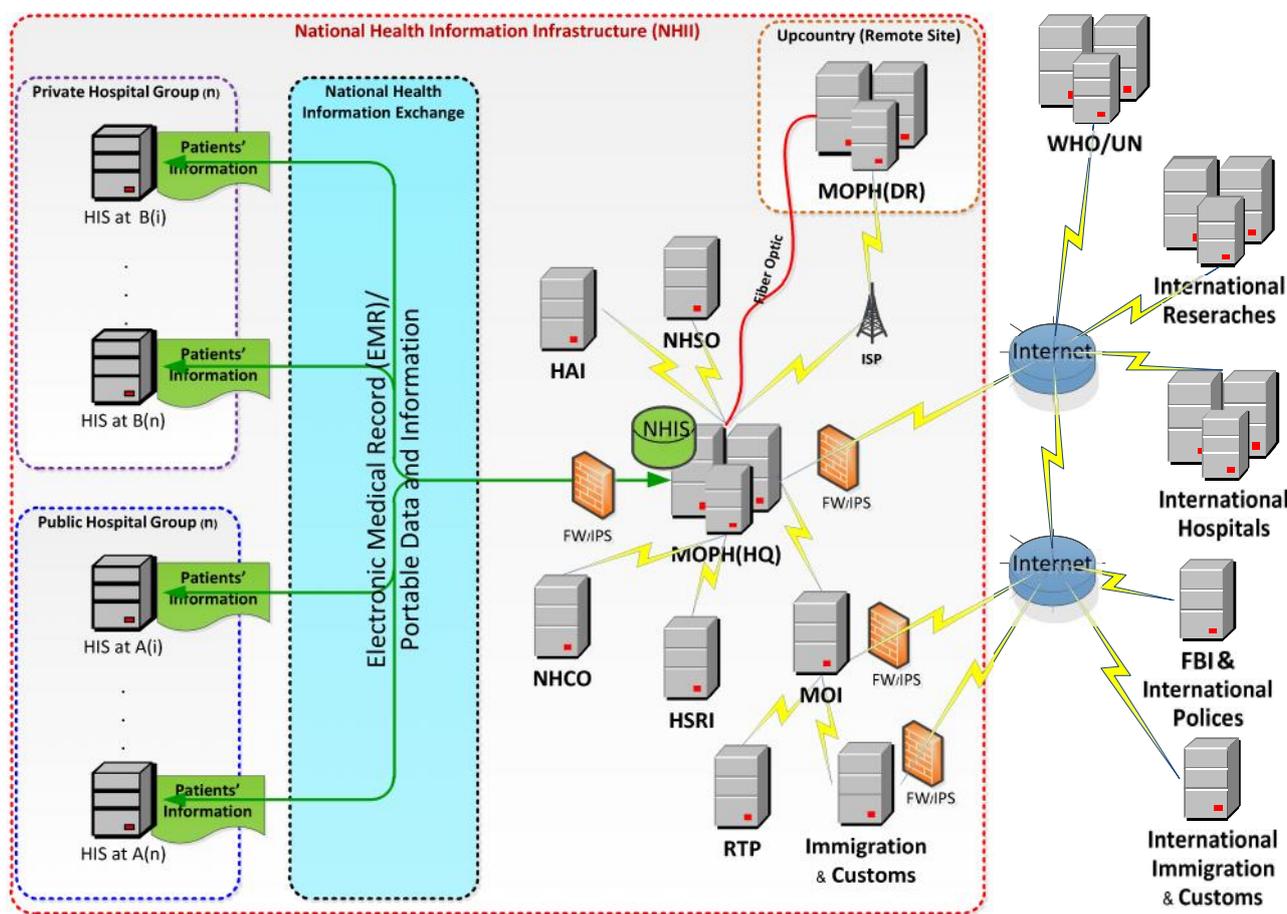


Figure 4. National health information infrastructure (NHII).



**SESSION**  
**PERFORMANCE ANALYSIS AND EVALUATION**

**Chair(s)**

**TBA**



# Analysis of a Man-in-the-Middle Experiment with Wireshark

Ming-Hsing Chiu, Kuo-Pao Yang, Randall Meyer, and Tristan Kidder

Department of Computer Science and Industrial Technology  
Southeastern Louisiana University, Hammond, Louisiana

**Abstract** - With the rapid growth of the Internet user population and the magnitude of the applications depending on the Internet these days, network security measures are becoming extremely important. For the Internet users, one of the best defenses against network attacks is to understand the patterns of the attacks and raise the awareness of abnormality as much as possible. In this paper, an experiment was employed to demonstrate a form of active attacks, called Man-in-the-middle (MITM) attack, in which the entire communication between the victims is controlled by the attacker. A detailed description of setting up the system for MITM is included. The victim initiated a few activities that cause the attacks, which were captured by Wireshark at the attacker site and analyzed. The result clearly reveals the pattern of the MITM attack. Some remarks on the preventive measures were made based on the result.

**Keywords:** Man-in-the-middle attack, Wireshark, ARP

## 1 Introduction

The *man-in-the-middle attack* (often abbreviated *MITM*) is a well-known form of active attack in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker[1,2]. It allows the attacker to eavesdrop as well as to change, delete, reroute, add, forge, or divert data [3]. For the Internet users, one of the best defenses to MITM attacks is to understand the patterns of the attacks and raise the awareness of abnormality during the attacks. As an effort to demonstrate the characteristics of the attack, an experiment was carried out and the network traffic was captured and analyzed by using the packet sniffer, Wireshark. Previous works that used Wireshark in the similar manner can be found in [4,5].

The paper is organized as follows. Section 2 provides the background information on the capturing and display processes of Wireshark. Section 3 gives a detailed description of setting up an experiment for demonstrating the MITM attack under Linux operating system. A few MITM activities were captured in the experiment and analyzed to search for the patterns of the attack in section 4. Preventive measures and warning signs of the MITM attacks were discussed in section 5. Finally, section 6 provides conclusion of this work.

## 2 Wireshark

Wireshark (formerly known as Ethereal)[6] is a free and open-source packet analyzer, based on libpcap. It is widely used in network troubleshooting, analysis, protocol development by network professionals as well as educators. It accepts wide range of protocols, such as TCP, IP, ARP, HTTP, and etc. Note that we use the terms packet and frame interchangeably in this paper.

In display mode, Wireshark presents a colorful window with three different areas when you open a captured file with a set of packets. On the top most area of the window is Area 1(listing area), which is the listing of all the captured frames. Each line is a summary of a frame displaying the information depicted on the top heading. When you click on a packet in Area 1, the detailed packet structure is shown on Area 2(detailed area) directly below Area 1. Clicking on a portion of the packet in Area 2 changes the display in Area 3(raw data area), which is the raw data of the frame shown in Area 2.

## 3 Setting up the system

In general, MITM involves three computers, two victims and one attacker. It is performed by the attacker sending a signal to the first victim telling the victim he is the second victim, and sending a signal to the second victim saying he is the first victim. This creates a Man-in-the-middle effect in which the first victim sends all its packets to the attacker which are then relayed to second victim and vice versa. In the experiment, as depicted in Figure 1, the second victim is a Web server. When the Spoofed connection is made, the victim browses the internet as normal.

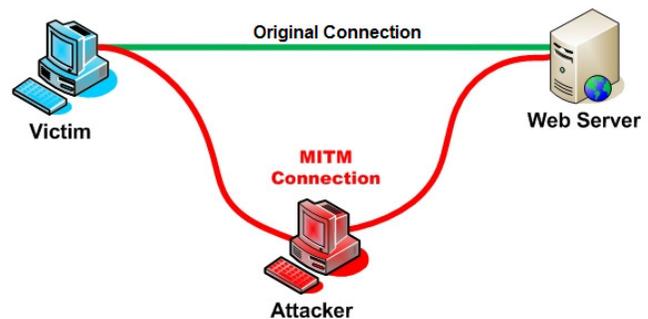


Figure 1: Man-In-The-Middle Attack

The experiment was carried out on a campus lab, under the supervision of the network administrator. Since both first victim and attacker reside in the same LAN (subnet) with a single gateway, we only need to spoof the victim and the gateway. This is because all traffic between the first victim and the Web server must pass through the gateway due to the last hop situation. During the experiment, the attacker was running on a Linux Backtrack3 operating system, since most of the tools needed are included as native applications in the operating system, except *sslstrip* which is to be discussed later.

There are a few steps involved in the setting up of the attack. A script file that carries out the set up, step by step, automatically was implemented and invoked at the onset of the experiment.

### 3.1 Enter interface and the Gateway IP address

Both parameters can be obtained by running *ipconfig* tool and the parameters must be entered to the operating system. In our case, the interface is 'eth0' and the Gateway IP address is 147.174.120.1.

### 3.2 Scan the network to find target IP

Use *nmap* tool to map network for accessible services and use *ipscan* to scan the subnet to find the target IP (victim), then enter the IP. Note that the attacker and the victim reside in the same subnet. The computer with IP = 147.174.120.208 was chosen as the victim.

### 3.3 Enable IP forwarding

In order for ARP spoofing to work, IP tables need to be prerouted and IP forwarding needs to be enabled. This is done by using *iptables* tool.

### 3.4 Complete ARP spoof

This step is also called ARP poisoning, in which the attacker take advantage of the ARP protocol by impersonate as victim to gateway, and as gateway to victim. The effect is the modification of IP forwarding table on both gateway's and victim's sites. After this step, attacker gets all the message exchange between Web server and the victim. Table 1 lists the correct MAC addresses and the modified MAC addresses on the IP tables of the gateway and the victim due to the effect of *arpspoof*. This table is useful for references when doing packet analysis in Wireshark. The MAC addresses shown on the table are the generic names for better readability.

Table 1: IP and MAC addresses table after the arp spoof

	IP address	MAC address	Modified address on Gateway's table	MAC on IP	Modified address on Victim's IP table
Gateway	147.174.120.1	PrimaryA_6b:40:99			DellComp_4e:4f:69
Victim	147.174.120.208	AmbitMic_cc:1b:6c	DellComp_4e:4f:69		
Attacker	147.174.120.235	DellComp_4e:4f:69			

## 4 Activities captured and analyzed

After ARP spoofing was run successfully, the victim initiated a few activities that demonstrate MITM attacks. These activities were captured by Wireshark and a Pcap file was generated at the attacker's site. Below are the analyses of the activities captured during the experiment. To improve the readability and save the space, in figure 2 to figure 6 (showing Wireshark display window), only a single frames will be included on Area 1(listing area). Area 3 (raw data area) will not be displayed. Note that doing a MITM attack produces several ARP frames as well as retransmission frames. Figure 2 shows an ARP frame that appeared several times during the experiment as the result of running *arpspoof*. This frame was sent from the attacker to the victim attempting to impersonate as gateway by inserting its own MAC address. This can be seen in the area showing the detail of Address Resolution Protocol, in which the Sender IP address is the gateway's IP address but the Sender MAC address is that of attacker's.

Figure 2: ARP Frame Showing the Attempt of the Attacker to Impersonate as Gateway

### 4.1 Victim browses www.google.com:

Since the attacker is effectively acting as the relaying station of the messages exchanged between the gateway and the victim, each message coming from gateway/victim will generate a retransmitted message to victim/gateway. Figure 3(frame # 15) and 4(frame # 16) are the original and the retransmitted HTTP frames that request for Web page from www.google.com respectively. Note that the only difference between these two frames is the source and destination MAC addresses at the Network Access Layer, Ethernet II, displayed in detailed area. The pattern of the MAC addresses of the source and destination pair in each frame clearly demonstrates the relaying behavior of MITM attack. The reply from www.google.com shows similar retransmission pattern except the MAC addresses of the source and destination pair are reversed.

No.	Time	Source	Destination	Protocol	Info
15	1.868159	147.174.120.208	74.125.229.18	HTTP	GET / HTTP/1.1
Frame 15: 506 bytes on wire (4048 bits), 506 bytes captured (4048 bits) on interface 0 Ethernet II, Src: AmbitMI_cc:1b:6c (00:d0:59:cc:1b:6c), Dst: DellComp_4e:4f:69 (00:08:74:4e:4f:69) Internet Protocol, Src: 147.174.120.208 (147.174.120.208), Dst: 74.125.229.18 (74.125.229.18) Transmission Control Protocol, Src Port: neoiface (1285), Dst Port: http (80), Seq: 1, Ack: 1, Len: 452 Hypertext Transfer Protocol GET / HTTP/1.1/\r\n Accept: */*\r\n Accept-Language: en-us\r\n Accept-Encoding: gzip, deflate\r\n User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; GTB6.5)\r\n Host: www.google.com\r\n Connection: Keep-Alive\r\n [truncated] Cookie: PREF=ID=b0305f3bbdf5afb2:U=be4f9f74cb21bf4b:TB=5:TM=1268761617:LM=1280796687:S=0Cq:					

Figure 3: Google page request originated from the victim

No.	Time	Source	Destination	Protocol	Info
16	1.879019	147.174.120.208	74.125.229.18	HTTP [TCP Retransmission]	GET / HTTP/1.1
Frame 16: 506 bytes on wire (4048 bits), 506 bytes captured (4048 bits) on interface 0 Ethernet II, Src: DellComp_4e:4f:69 (00:08:74:4e:4f:69), Dst: PrimaryA_6b:40:99 (00:20:9c:6b:40:99) Internet Protocol, Src: 147.174.120.208 (147.174.120.208), Dst: 74.125.229.18 (74.125.229.18) Transmission Control Protocol, Src Port: neoiface (1285), Dst Port: http (80), Seq: 1, Ack: 1, Len: 452 Hypertext Transfer Protocol GET / HTTP/1.1/\r\n Accept: */*\r\n Accept-Language: en-us\r\n Accept-Encoding: gzip, deflate\r\n User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; GTB6.5)\r\n Host: www.google.com\r\n Connection: Keep-Alive\r\n [truncated] Cookie: PREF=ID=b0305f3bbdf5afb2:U=be4f9f74cb21bf4b:TB=5:TM=1268761617:LM=1280796687:S=0Cq:R1EP					

Figure 4. Google page request relayed by the attacker

## 4.2 Victim logs in to an insecure login site: projecteuler.net

This example shows that the password in an insecure login can be easily snatched by way of MITM attack. Again, we show the two frames that were originated from the victim and the relay from the attacker to the Web site.

As depicted in Figure 5 (frame 113) and Figure 6 (frame 114), the password can be seen on the last line of the detail area, in which the username is iTruth and the password is CMPS309. The Attacker may alter the login information in the retransmission, as a result, the victim cannot login.

No.	Time	Source	Destination	Protocol	Info
113	5.020459	147.174.120.208	95.154.244.24	HTTP	POST /index.php HTTP/1.1 application/x-www-form-urlencoded
Frame 113: 556 bytes on wire (4448 bits), 556 bytes captured (4448 bits) on interface 0 Ethernet II, Src: AmbitMI_cc:1b:6c (00:d0:59:cc:1b:6c), Dst: DellComp_4e:4f:69 (00:08:74:4e:4f:69) Internet Protocol, Src: 147.174.120.208 (147.174.120.208), Dst: 95.154.244.24 (95.154.244.24) Transmission Control Protocol, Src Port: hp-sci (1299), Dst Port: http (80), Seq: 1, Ack: 1, Len: 502 Hypertext Transfer Protocol Line-based text data: application/x-www-form-urlencoded username=iTruth&password=CMPS309&login=Login					

Figure 5: Login to Projecteuler originated from the Victim

No.	Time	Source	Destination	Protocol	Info
114	5.020857	147.174.120.208	95.154.244.24	HTTP [TCP Retransmission]	POST /index.php HTTP/1.1
Frame 114: 556 bytes on wire (4448 bits), 556 bytes captured (4448 bits) on interface 0 Ethernet II, Src: DellComp_4e:4f:69 (00:08:74:4e:4f:69), Dst: PrimaryA_6b:40:99 (00:20:9c:6b:40:99) Internet Protocol, Src: 147.174.120.208 (147.174.120.208), Dst: 95.154.244.24 (95.154.244.24) Transmission Control Protocol, Src Port: hp-sci (1299), Dst Port: http (80), Seq: 1, Ack: 1, Len: 502 Hypertext Transfer Protocol Line-based text data: application/x-www-form-urlencoded username=iTruth&password=CMPS309&login=Login					

Figure 6: Login to Projecteuler relayed by the attacker

## 4.3 Other example

In a separate experiment, the attacker used Wireshark to capture eavesdrop of Instant Messenger. Figure 7 depicts a composite picture, in which Messenger window and the corresponding Wireshark displays are linked by the red marker. Note that the filter “MSNMS” was applied to obtain the displays.

Figure 7: A snapshot of the eavesdrop of Instant Messenger

## 5 Preventative measures

To the average victim, the MITM attack is relatively hard to detect. This is particularly true when the victim is engaged in non-secure transactions as shown in the previous examples. In fact, there are some free tools available for detecting the anomaly when MITM is being performed. Thus help preventing the user from becoming a victim of the attack. A good example is the tool, called *DecaffeinatID*, which monitors user's gateway MAC address. If that changes, as in the case during a MITM attack, it notifies the user with a popup box as shown in Figure 8.



Figure 8: A warning Message of Potential MITM Attack

Some variants of MITM attack, such as those equipped with the *sslstrip* tool, are capable of compromising SSL/TSL type of security measures, making the interception of the secured data possible. To some extent, though, when dealing with secure login sites, the attacker is relying on the victim's ignorance to achieve success. For instance, when logging into sites like Chase.com or live.login.com, it verifies the certificate. When a MITM attack is being performed, the victim will receive a certificate warning. If the victim accepts the false certificate, then the attacker will intercept the login information; however, if the victim does not, they will not be able to login to that site, which is a wiser choice. In summary, the awareness of abnormality is important in the preventive measures.

## 6 Conclusions

For the Internet users, one of the best defenses against network attacks is to understand the patterns of the attacks and raise the awareness of abnormality as much as possible. We use an experiment to demonstrate a form of active attacks, Man-in-the-middle (MITM). Wireshark was used to capture and analyze the MITM activities in the experiment. From the result, we identified the characteristics of the MITM attack. We also make some remarks on the preventive measures and emphasize the importance of awareness of the abnormality. We found that Wireshark is an indispensable tool in carrying out the experiment which is suitable in disseminating the knowledge of the MITM attack in the classroom environment.

## 7 References

- [1] [http://en.wikipedia.org/wiki/Man-in-the-middle\\_attack](http://en.wikipedia.org/wiki/Man-in-the-middle_attack)
- [2] M. E. Whitman, H. J. Mattord, *Principles of Information Security*, Thomson Course Technology, 2005.
- [3] D. Radcliff, "What Are They Thinking?" *Network World*, March 1, 2004
- [4] M. A. Qadeer, M. Zahid, A. Iqbal, M. R. Siddiqui, "Network Traffic Analysis and Intrusion Detection Using Packet Sniffer," *Proc. Of Seond. Int. Conf. on Comm. Software and Networks*, 2010, pp 313-317.
- [5] S. Wang, D. Xu, S. Yan, "Analysis and Application of Wireshark in TCP/IP Protocol Teaching," *Proc. Of Int. Conf. on E-Health Networking, Digital Ecosystem and Technologies*, 2010, pp 269-272.
- [6] <http://www.wireshark.org/>

# Evaluation of Network Port Scanning Tools

Nazar El-Nazeer and Kevin Daimi  
 Department of Mathematics, Computer Science and Software Engineering  
 University of Detroit Mercy,  
 4001 McNichols Road, Detroit, MI 48221  
 {elnazen, daimikj}@udmercy.edu

## ABSTRACT

Neglecting network port scans could result in unavoidable consequences. Network attackers continuously monitor and check communication ports looking for any open port. To protect computers and networks, computers need to be safeguarded against applications that aren't required by any function currently in use. To accomplish this, the available ports and the applications utilizing them should be determined. This paper attempts to evaluate eight port scanning tools based on fifteen criterions. The criteria were reached after fully testing each tool. The outcomes of the evaluation process are discussed.

## Keywords

Network Security, Evaluation Criteria, Network Security Tools, Network Port Scanning

## I. INTRODUCTION

A computer network is any group of independent computers and devices that communicate with one another over a shared network channel. With networking, people can share files, printers, and storage devices. Furthermore, they can exchange e-mail, disclose internet links of common interest, or conduct video conferences. Computer Networks are used for business, home, mobile, and social applications. There are different categories of networks including Local Area Networks, Wide Area Networks, Wireless Network, and Internetworks. Within a network, computers and devices communicate with each other via protocols [3], [11], and [27].

It is currently almost impossible to end or weaken the ties between humans and computer networks. People rely on computer networks to accomplish many essential and critical tasks. Therefore, it is very demanding to secure our networks. Network security

implies protecting data and information from attacks during their transmission from the source to destination. Attackers can detect the vulnerabilities in networks and possibly pose enormous threats in these situations. To prevent problems, cryptology provides the most promising measures to deter, prevent, detect, and correct security violations.

To protect computer networks, a number of protection tasks need to be implemented. These tasks are needed to enforce the security for wireless network, electronic mail, IP, and at the transport level. Furthermore, these tasks should efficiently deal with intruders and malicious software [23].

Internet and web are tremendously vulnerable to various attacks. Therefore securing web services is a critical requirement. In particular, security at the transport layer must never be overlooked. The subdivision of the Internet by the transport layer presents ample outcomes both in the way in which business is performed on the network and with regard to the vulnerability caused by the openness of the network [6]. Patel et al [20] presented a system capable of granting a high level of security and performance. It permits each host to shield itself from untrusted transport code and to guarantee that this code will not impair other network users. For wireless networks, the Wireless Transport Layer Security (WTLS) should efficiently provide the highest level of protection. To achieve this, an efficient architecture for the hardware implementation of WTLS is demanding. Such architecture must support bulk encryption, authentication and data integrity, and operate alternatively for a set of ciphers, such as IDEA, DES, RSA, D.H., SHA-1 and MD5 [22].

Wireless Local Area Networks (WLANs) are subject to some vital security vulnerabilities and the preference of security protocol is a critical concern for IT administrators. Users need to be aware of the threats of the wireless security protocols; WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access) and RSN (Robust Security Network) [9]. Cryptology is

undoubtedly suitable for Wireless Sensor Networks (WSNs). The application of a simple non-interactive key exchange scheme at the system-level has been investigated with regards to its suitability. It was concluded that it is particularly suitable for many Wireless Sensor Network (WSN) scenarios. [25]. Attacks are possible on wireless LANs if suitable precautions are not exercised. Tews et al [26] introduced two possible attacks: an improved key recovery attack on WEP and an attack on WPA secured wireless networks. These attacks are effective if network traffic is encrypted using Temporal Key Integrity Protocol (TKIP).

Electronic mail (email) systems have demonstrated an increase in complexity to the point where their reliability and usability are becoming questionable [14]. A number of electronic mail security protocols exist, such as the Pretty Good Protocol (PGP), Secure/Multipurpose Internet Mail Extension (S/MIME), and DomainKeys Identified Mail (DKIM). Roth et al [21] indicated that support for robust electronic mail security is broadly available yet only few users appear to take advantage of these features. It seems that the operational cost of security outweighs its recognized advantages.

Internet Protocol (IP) security should be recognized by current and future users and applications [7]. IP security takes care of authentication, confidentiality, and key management. Any overlay network on top of IP, such as The IP Multimedia Subsystem (IMS), must be fully protected. IMS, which employs the Session Initiation Protocol (SIP) as the primary signaling mechanism, introduces a number of new security challenges for both network providers and users [15]. A survey of common security threats which mobile IP networks are exposed to as well as some proposed solutions to deal with such threats are presented in [18].

Unauthorized intrusion into computer networks poses a great threat, especially if it is not detected. Intrusion detection systems identify unusual activities or pattern of activities that are known to trigger attacks. Once such activities are detected, measures could be followed to prevent or minimize the consequences of such attacks. A number of approaches for intrusion detection have been suggested. A solution to the problem of capturing an intruder in a product network, based on the assumption of existing algorithms for basic member graphs of a graph product, was proposed in [16]. A process for the algebraic intruder model for verifying a brand of liveness properties of security protocols was presented in [10]. With regards to this model, formal verification of fair exchange protocols was discussed.

Malicious software aims at harming computing systems when deliberately brought in or incorporated on a system. This is another critical threat that should be detected and deterred. The number of malware variants has increased dramatically. Automatic malware classification is becoming a central research area. A behavior-based automated classification method based on distance measure and machine learning was proposed in [17]. Confidential information protection is a key concern for organizations and individuals. One of the main threats to confidentiality is malicious software. Present security controls are insufficient for preventing malware infection [8]. To detect unknown malicious software, it is vital to analyze the software for its influence on the system when the software is executed. To implement that, the software code must be statically analyzed for any malicious activity [12].

Many network security tools exist. Some of these are open source tools. The goal of these tools is to scan various parts of the network looking for possible threats. This will enhance the security of what was mentioned above. Examples of these tools include Vulnerability Scanners, Packet Sniffers, Vulnerability Exploitation tools, and Port Scanners. A port is an application identifiable software construct acting as an endpoint in various communications. Ports are mainly used by the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) of the Transport Layer. Ports are identified by numbers. For example, Port 25 is reserved for Simple Mail Transfer, and port 80 is used by HTTP. A port scan is an attack that tries to identify known vulnerabilities of a service on active ports. Both network administrators and attackers use port scanner tools to probe servers/hosts for open ports, but with different purposes. The administrator's goal is to verify and ensure that security policies are enforced. Attackers intend to compromise the running services.

The purpose of this paper is to evaluate network port scanning tools. For this purpose 17 tools have been initially selected for this study. For the time being, only eight tools are fully tested and selected for the evaluation purposes. The rest are by no means rejected, but will be included in the final evaluation process in the future. For the evaluation procedure, fifteen criteria have been selected. Evaluation tables will be presented and the findings will be discussed.

## II. PORT SCANNING TOOLS OVERVIEW

The port scanning tools, which are included in the evaluation process, are briefly explained below.

### A. Nmap

Nmap [19] is an open source program (GNU). It is an important tool for network administrators. Nmap can be used for discovering, monitoring, and troubleshooting TCP and UDP based systems.

Nmap is a general purpose network scanner. It supports most of the known operating systems including Windows, Linux, UNIX, and Mac OS X. However, for Windows the Windows Packet Capture Driver (WinPcap) is needed.

Command line arguments could be used but are case sensitive. Many scanning options require administrator privileges. On Linux and Unix, Nmap is run using the "sudo" command. If a user scans remote hops that are not in their LAN, incorrect information might be received due to the fact that firewalls, routers, proxy servers and other devices are capable of skewing the scanning results of Nmap. Aggressive scanning may crash some systems leading to system downtime and data loss.

### B. SuperScan 4.0

The SuperScan [24] tool was created by Foundstone's security experts. They established the first network security consulting practices at two Big 6 accounting firms. Foundstone made their reputation as an enterprise network security company. They contributed to improving network security knowledge through numerous articles and white papers.

Foundstone was obtained by McAfee in September 2004. They will continue to provide their services as a division of McAfee.

SuperScan provides three main tools: TCP port scanner, Ping tool, and Resolver tool. To run the software, administrator privileges are needed.

### C. Advanced Port Scanner

Advanced Port Scanner [2] is a GUI-based free and small tool. It is a fast and simple port scanner for Win32 and Win64 platforms. It contains descriptions for common ports database, and can perform scans on predefined port ranges.

Advanced Port Scanner is a multithreading tool. Therefore, it is capable of performing faster scans by increasing the maximum number of threads. It only allows the observation of alive/dead computers. Users can define the maximum time (in milliseconds) that the LAN scanner needs to take on each port scan.

### D. Advanced Administrative Tools

Advanced Administrative Tools (AATools) [1] is mainly a security diagnostic and testing utility. It is used to verify the integrity of the security and firewall functions to protect the computer and the data it stores. AATools network monitor maps the operational ports to their proper applications. This implies that it provides a tracking facility to track applications with port maps.

This tool can perform the following tasks: Port Scanner, Proxy Analyzer, RBL Locator, Trace Route, Email Verifier, Links Analyzer, Network Monitor, Process Monitor, System Information, Resource Viewer, and Registry Cleaner.

The Port Scanner is used to conclude the active ports/services using TCP/UDP ports. It also allows multiple addresses and a list of ports scan, resolves or replaces host names into IP addresses, searches on the DNS for a host name before scanning, supports editing ports from a list, and scans active ports that Trojan or Backdoor programs may use.

### E. Angry IP Scanner

Angry IP Scanner [4] is an open source GUI-based cross-platform software. It is free to use and can be redistributed, and modified. For this tool, Java presents a solid platform for cross-platform development, rendering more than 95% of the code to be platform independent.

It was selected to use the Standard Widget Toolkit (SWT), provided by the Eclipse project. Its advantages comprise the usage of native GUI controls and widgets on every supported platform. These will make Java programs indistinguishable from the native ones. This is important to users because they desire their system-wide settings, themes, and operating system standards to be admired.

### F. Atelier Web Security Port Scanner

Atelier Web Security Port Scanner [5] can carry out TCP Port and UDP Port Scanning. It has the ability to map open ports to applications, provide complete details of local host network information as well as accurate and ample LAN details. It has a prevailing NetBIOS scanner, and ports database.

The tool also provides a complete statement of network errors during the TCP scanning. The statement includes standard service keyword, remote port number, error

description, and error number. Atelier Web Security Port Scanner has TCP Sync Scanning engine. The adjustable maximum number of all ports opened together is 60.

### G. Unicornscan

Unicornscan [28] is a TCP and UDP port scanner. It was designed to produce an engine that would be accurate, scalable, effective, and adjustable. It runs under the rules of the GPL license. Unicornscan supports UNIX operating system and it has now an available version for Fedora Linux operating system.

Unicornscan is capable of providing asynchronous stateless TCP scanning with all alternatives of TCP Flags, asynchronous stateless TCP banner grabbing, asynchronous protocol specific UDP Scanning, packet capture (PCAP) file logging and filtering, and relational database output.

### H. GFILANguard

GFILANguard [13] is employed for Patch Management, Vulnerability Checking and Network Auditing. This tool can scan networks and ports to detect, identify and correct security vulnerabilities. It manually or on scheduled basis scans and then analyzes the services running in the open ports. It deploys fingerprint technology to check whether the service is safe or there is a hijack operation. This helps to maintain the network. GFILANguard needs 102 MB to run.

GFILANguard supports Patch Management, Vulnerability Management, Network and Software Auditing, Assets Inventory, Change Management, and Risk Analysis and Compliance.

## III. NETWORK SECURITY TOOLS EVALUATION

The eight tools were assessed using fifteen criterions. In section A, the criteria will be stated. Section B will provide the actual assessment using tables.

### A. Evaluation Criteria

To evaluate the various tools, we have based our assessment on fifteen criterions. These criteria were concluded after examining the tools specifications and fully testing each tool. We only relied on the tool documentation for criterions 1 and 15. The rest are technical criterions, and thus, were extensively tested. We are not claiming, however, that the set of criterions is complete. The fifteen criterions are stated below:

- *Last Update*: Date when the current version was released.
- *IP Ranges*: Maximum number of IPs which the tool can scan in one entry.
- *Test Method*: Method used before initiating port scanning to check if the computer is live or not.
- *TCP SYN Scanning*: Capability of the tool to scan TCP.
- *UDP Scanning*: Capability of the tool to scan UDP.
- *Banner Grabbing*: Whether the tool can gather information about computer systems on a network and the services running on its open ports.
- *Port List DB*: Whether the tool contains a database of descriptions of services associated with the port number.
- *Useful Tools*: Other features or services besides the basic port scanning.
- *Interface*: Type of user interface.
- *Platform*: Supported operating systems.
- *Active Port Mapping*: Whether the tool allows a mapping of the open port with the application using that port.
- *MAC Address Detection*: Ability to detect MAC address.
- *Query Application Protocols*: Whether the tool is capable of looking for all types of application protocols, such as web servers, databases, DNS servers, FTP, and Gopher servers.
- *UN/PW Recovery*: Ability to recover user name (UN) and password (PW) using brute force.
- *Free*: Whether the tool is free or not.

### B. Evaluation Procedure

The above criteria are used to compare the eight tools in question. The same approach will be used when new tools are added. The criteria were distributed among three tables, with five criterions per table. Depending

on the criteria used, some cells will contain yes/no, and others will contain various values. Tables I – III illustrate the outcomes of the evaluation.

Table I  
TOOLS COMPARISON – PART I

	L/Update	IP Ranges	Test Methods	TCP SYN Scanning	UDP Scanning
Nmap	1, 2011	Unlimited	ICMP	Yes	Yes
SuperScan 4.0	8, 2003	Unlimited	ICMP	Yes	Yes
Advanced Port Scanner	7, 2006	Unlimited	ICMP	Yes	Yes
AATools	1, 2006	Unlimited	ICMP	Yes	Yes
AngryIP	3, 2009	Unlimited	ICMP	Yes	Yes
AWSP	2, 2002	Unlimited	ICMP	Yes	Yes
Unicornscan	2, 2010	Unlimited	ICMP	Yes	Yes
GFILANguard	11, 2010	3999	ICMP	Yes	Yes

Table III  
TOOLS COMPARISON – PART III

	Active Port Mapping	MAC Address Detection	Query Application Protocols,	UN/PW Recovery	Free
Nmap	Yes	Yes	Yes	Yes	Yes
SuperScan 4.0	No	No	No	No	Yes
Advanced Port Scanner	No	No	No	No	Yes
AATools	Yes	No	No	No	No
AngryIP	No	No	No	No	yes
AWSP	Yes	yes	No	No	No
Unicornscan	No	No	No	No	Yes
GFILANguard	Yes	Yes	No	No	No

Table II  
TOOLS COMPARISON – PART II

	Banner grabbing	Port List DB	Useful Tools	Interface	Platform
Nmap	yes	Yes	179 Scripts, 60 Libraries	GUI and command line	Linux, Mac OS X, Windows, and many UNIX platforms (Solaris, Free/Net OpenBSD, etc.), and some smart cell phone
SuperScan 4.0	yes	Yes	Ping, Traceroute, Whois	GUI	Windows 2000 and XP
Advanced Port Scanner	No	No		GUI	Windows 95/98/ME/NT4.0/2000/XP/2003/Vista/2008 and Windows 7 (32 bit, 64 bit)
AATools	No	Yes	Proxy Analyzer, Real Time Blacklist Locator, Trace Route, Email Verifier, Links Analyzer	GUI	9x/Me/NT4/2000/XP
AngryIP	No	No	Fetcher, Openers, Exporters,	GUI	Mac OS X, Linux systems, and Windows 98/ME/2000/XP/Vista
AWSP	No	Yes	Ping, Traceroute, NSLOOKUP	GUI	Windows NT/2000/XP
Unicornscan	Yes	Yes	Relational database output, Custom module support	Only command line	Unix, fedora
GFILANguard	No	Yes	Patch Management, Vulnerability Checking, Network Auditing, DNS Lookup, Traceroute, Whois	GUI	(x86 or x64) - Windows Server 2008, 2003, 2000, Windows 7, Vista, XP, Windows SBS 2008, 2003

IV. OUTCOMES DISCUSSION

A number of interesting observations can be spotted in the above tables. Table I reveals that all the tools are capable of TCP SYN and UDP Scanning. Also, all the tools in question use the ICMP method to check whether the computer is live or not. With regards to IP ranges, all of them allow unlimited range except *GFILANguard*, which is limited to 3999. *Nmap*, *Unicornscan* and *GFILANguard* received the most recent update.

Table II indicates that *Nmap*, *SuperScan 4.0*, and *Unicornscan* are capable of gathering information about computer systems on a network. All tools except *Advanced Port Scanner* and *AngryIP* support a database of service descriptions. In addition, all tools except

*Advanced Port Scanner*, grant other features or services with varying amounts of services in addition to the basic port scanning. The tools, with the exception of *Unicornscan*, accommodate GUI interface. *Nmap* adds a command line interface. The eight tools run on various operating systems. However, *Nmap*, followed by *AngryIP*, *GFILANguard*, and *Advanced Port Scanner* support more operating systems than the rest.

From Table III, we can detect that *Nmap*, *AATools*, *AWSP*, and *GFILANguard* allow for active port mapping. Only *Nmap*, *AWSP*, and *GFILANguard* can detect MAC addresses. Finally only *NMAP* grants querying application protocols, and recovering user name and password via brute force search.

The assessment exhibits that *Nmap* is the superior tool given these criteria. *AWSP* and *GFILANguard* follow. *The Advanced Port Scanner* and *AngryIP* satisfy fewer criterions than the rest.

## V. CONCLUSIONS

Network administrators implement conditions and policies needed to inhibit and monitor unauthorized access, exploitation, modification, or denial of the network and its resources. To do this, there are many network security tools available for various security functions. This paper concentrated on network port scanning tools. To this extent, eight tools have been compared based on fifteen criterions. As this is a continuous process, more tools will be added in the future to complete the study. Based on the comparison tables above, it is concluded that *Nmap* provides more features than other tools involved in the study. The set of criterions is by no means a closed set. Further criterions will be added in the future.

## REFERENCES

- [1] Advanced Administrative Tools, G-Lock software, Available: <http://www.glocksoft.com/aatools.htm>.
- [2] Advanced Port Scanner, Radmin, Available: <http://www.radmin.com/products/previousversions/portscanner.php>.
- [3] M. Agrawal, *Business Data Communications*, Wiley, 2011.
- [4] Angry IP Scanner, Available: <http://www.angryip.org/w/Home>.
- [5] Atelier Web Security Port Scanner, AW Atelier Web, Available: <http://www.atelierweb.com/pscan>.
- [6] J. A. Audestad, "Internet as a multiple graph structure: The role of the transport layer," *Information Security Technology Report*, Vol. 12, No. 1, pp. 16-23, March 2007.
- [7] Z. Bojkovic, "Some IP security issues," in *Proc. the 10th WSEAS International Conference on Mathematical Methods and Computational Techniques in Electrical Engineering*, Wisconsin, 2008, pp. 138-144.
- [8] K. Borders, "Protecting confidential information from malicious software," Ph.D. Dissertation, Dept. University of Michigan, Ann Arbor, MI, USA, 2009.
- [9] H. I. Bulbul, I. Batmaz, and M. Ozel, "Wireless network security: comparison of WEP (Wired Equivalent Privacy) mechanism, WPA (Wi-Fi Protected Access) and RSN (Robust Security Network) security protocols," in *Proc. 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia*, Adelaide, Australia, January, 2008.
- [10] J. Cederquist, and M. Dashti, "An Intruder Model for Verifying Liveness in Security Protocols," in *Proc. the fourth ACM workshop on Formal methods in security (FMSE'06)*, Alexandria, VA, 2006, pp. 23-32.
- [11] D. E. Comer, *Computer Networks and Internets*, Prentice Hall, 2009.
- [12] J. Dai, "Detecting Malicious Software by Dynamic Execution," Ph.D. Dissertation, University of Central Florida, Orlando, FL, USA, 2009.
- [13] GFILANguard, GFI, Available: <http://www.gfi.com/lannetscan>.
- [14] R. Hall, "Fundamental Non-modularity in Electronic Mail," *Automated Software Engineering*, Vol. 12, No. 1, pp. 41-79, 2005.
- [15] M. T. Hunter, R. J. Clark, and F. S. Park, "Security issues with the IP multimedia subsystem (IMS)," in *Proc. the 2007 Workshop on Middleware for next-generation converged networks and applications (MNCNA'07)*, Newport Beach, 2007.
- [16] N. Imani, H. Sarbazi-Azad, and A.Y. Zomaya, "Capturing an Intruder in Product Networks," *Journal of Parallel and Distributed Computing*, Vol. 67, No. 9, pp. 1018-1028, 2007.
- [17] J. Lin, "On Malicious Software Classification," in *Proc. the 2008 International Symposium on Intelligent Information Technology Application Workshops (IITAW '08)*, Shanghai, China, 2008, pp. 368-371.
- [18] M. C. Niculescu, E. Niculescu, and I. Resceanu, "Mobile IP Security and Scalable Support for

- Transparent Host Mobility on the Internet, in *Proc. 7th WSEAS International Conference on Applied Computer Science*, Wisconsin, 2007, pp. 214-221.
- [19] Nmap, Nmap.org, Available: <http://nmap.org/>.
- [20] P. Patel, A. Whitaker, D. Wetherall, J. Lepreau, and T. Stack, "Upgrading transport protocols using untrusted mobile code," in *Proc. the Nineteenth ACM Symposium on Operating Systems Principles (SOSP '03)*, New York, 2003, pp. 1-14.
- [21] V. Roth, T. Straub, and K. Richter, "Security and usability engineering with particular attention to electronic mail," *Int. Journal of Human-Computer Studies*, Vol. 63, No. 1-2, pp. 51-73, 2005.
- [22] N. Sklavos, P. Kitsos, K. Papadopoulos, and O. Koufopavlou, "Design, Architecture and Performance Evaluation of the Wireless Transport Layer Security," *The Journal of Supercomputing*, Vol. 36, No. 1, pp. 33-50, April 2006.
- [23] W. Stallings, *Network Security Essentials – Applications and Standards*, Prentice Hall, 2011.
- [24] SuperScan, McAfee Foundstone Practices, Available: <http://www.foundstone.com/>
- [25] P. Szczechowiak, A. Kargl, M. Scott, and M. Collier, "On the application of pairing based cryptography to wireless sensor networks," in *Proc. the second ACM conference on Wireless network security (WiSec '09)*, New York, 2009, pp. 1-12.
- [26] E. Tews, and M. Beck, "Practical attacks against WEP and WPA," in *Proc. The second ACM conference on Wireless network security (WiSec '09)*, New York, 2009, pp. 79-86.
- [27] A.S. Tanenbaum, and D.J. Wetherall, *Computer Networks*, Prentice Hall, 2011.
- [28] Unicornscan, Available: [www.unicornscan.org](http://www.unicornscan.org).

# Engineering Aspects of Hash Functions

Saif Al-Kuwari

Department of Computer Science  
University of Bath, Bath, BA2 7AY, UK

**Abstract**—Hash functions have numerous applications in cryptography, from public key to cryptographic protocols and cryptosystems. Evidently, substantial effort was invested on designing "secure" hash functions, unintentionally overlooking other engineering aspects that may affect their use in practice. However, we argue that in some applications, the efficiency of hash functions is as important as their security. Unlike most of the existing related works in the literature (which merely report on efficiency figures of some popular hash functions without discussing how and why these results were obtained), we not only discuss how to carry out efficiency evaluations, we also provide a set of optimization guidelines to assist implementers in optimizing their implementations. We demonstrate this by adopting an existing SHA-1/SHA-2 implementation and show how minor optimization can lead to significant efficiency gain.

**Keywords:** Hash Function, Efficiency, Optimization, Evaluation.

## 1. Introduction

Today, cryptographic hash functions play a major role in most cryptographic applications. Abstractly, hash functions are transformation procedures that given data, they return (small, fixed) fingerprints. A typical hash function consists of two components: a compression function and a construction. The compression function is a function mapping a larger fixed-size input to a smaller fixed-size output, and the construction is the way the compression function is being repeatedly called to process a variable-length message. Most of the literature is exclusively concerned with the design and cryptanalysis of hash functions. However, while the security of hash functions is certainly a highly important aspect, for some applications, especially the ones processing large amount of data, the efficiency (how fast the hash function is) is also important. Although there have been some efforts in evaluating the performance of hash functions, e.g. [1], it is clear that this is a largely overlooked evaluation criterion. Even the contributions that provide such efficiency evaluations, they generally only make the efficiency reports, without elaborating on how to improve<sup>1</sup> them. In this paper, we try to do this by considering implementations targeted for Intel platforms.

<sup>1</sup>We note that some SHA-3 submissions include optimization discussions, e.g. [2], but these by no means are comprehensive.

The paper is organized as follows, in section 2, we discuss the main factors affecting the efficiency of hash functions (and any code in general). Section 3 provides a concise overview of contemporary Intel platforms and some of their advanced architectural features. Our main discussion is in section 4 where we investigate how to optimize code on Intel platforms; though most of these optimization techniques are generic and applicable to other platforms. In section 5 we show how to carry out performance evaluations of hash functions and present a sample SHA-1/SHA-2 optimization case study in which we demonstrate how minor optimizations can greatly improve the overall efficiency of hash function. Finally, we conclude in section 6.

## 2. Efficiency Evaluation

The efficiency of any cryptographic primitive can significantly influence its popularity. For example, Serpent [3] was one of the AES (Advanced Encryption Standard) competition finalists and it was described by NIST (the competition organizer) as having a *high* security margin. However, in the last round of the competition, Serpent failed in favor of Rijndael [4], which was described as having just an *adequate* security, because Serpent was very slow in software compared to Rijndael. In this paper, we will be mainly concerned with the software efficiency of hash functions (but see section 2.2 for a brief discussion about hardware efficiency).

### 2.1 Software Optimization

Generally, there are two types of software optimizations, *high-level* and *low-level* optimizations. In high-level optimization, a cross-platform implementation written in a high-level language, such as C, is optimized. However, different compilers may treat high-level code slightly differently such that a code might be considered optimized only if it was compiled by a particular compiler. On the other hand, low-level optimization involves optimizing a machine (or assembly) code, and is rarely cross-platforms since different platforms often use different instruction sets. While optimizing a low-level code is tedious and error-prone, it gives the highest degree of control over the code. In general, efficiency requirements highly depend on the application, and thus the targeted application should also be taken into account when implementing a hash function. Software efficiency of hash functions can be influenced by several factors including,

the platform in which the hash function is executed, the compiler used to compile the hash function code, and the executing operating system (hash functions optimized for 64-bit operating systems are slower in 32-bit operating systems, e.g., Skein [2]).

### 2.1.1 Platforms

Both high-level and low-level optimizations are usually *tuned* for a specific platform. For example, in the SHA-3 competition<sup>2</sup> the reference platform in which the candidates were instructed to evaluate their submissions on was Intel Core 2 Duo, thus most of the candidate submissions were especially tuned to be optimal in Intel platforms (which mean that they may not be optimal in other platforms!). Platforms can be roughly classified as follows:

- High-end. These are platforms with high computational and memory resources, and usually based on 32-bit or 64-bit architectures, often with multiple processing cores. Examples include Intel and AMD.
- Intermediate. These are most 16-bit and 32-bit microcontrollers<sup>3</sup>. Examples include ARM and AVR.
- Low-end. These are 8-bit platforms with limited computational and memory (usually kilobytes) resources. Examples include Smart Cards and FRID.

### 2.1.2 Compilers

Another very important factor to consider when investigating hash functions efficiency is the sophistication of the compiler. Most of the available compilers (commercial and open source) like Microsoft Visual Studio and GCC are sophisticated enough to automatically optimize the code. However, these compilers sometimes apply some optimization techniques that may not be optimal for all platforms. Thus, it is advisable to compile the code with several different compilers and use different optimization switches, then only choose the optimum one for a target platform, although this process may be tedious. One would think that a commercialized compiler developed by the vendor of a particular platform, would outperform other open source or third-party commercial compilers. However, Wenzel-Benner and Graf [5] showed that this is not always the case when they implemented several SHA-3 hash function candidates on an ARM Cortex platform and then compiled them twice, once by ARM-CC compiler (ARM's own compiler) and another with GCC compiler (open source compiler). Surprisingly, they found that in some cases, GCC compiled code is more efficient than that compiled by ARM-CC.

<sup>2</sup>For comprehensive resource about SHA-3 competition and all its candidates, see [http://ehash.iaik.tugraz.at/wiki/The\\_SHA-3\\_Zoo](http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo)

<sup>3</sup>Note that some microcontrollers are low-end platforms.

### 2.1.3 Instruction Sets

High-level code will eventually be converted into a low-level (machine) code that consists of instructions. A platform with only several tens of instructions will most likely not perform as well as another with hundreds of instructions, basically because no matter what optimization techniques are applied, if no efficient instructions exist for a particular operation, the code will need to be converted to a series of instructions implementing that operation. Most recent platforms adopt the so-called SIMD technology, which provide instructions allowing for parallel data execution.

## 2.2 Hardware Optimization

Although hardware implementation and optimization is not the main focus of the current paper, in this section we discuss a few interesting results of recent hardware evaluation of SHA-3 candidates. In [6], Tillich *et al.* presented optimized hardware implementations of the 14 SHA-3 round 2 candidates. Their results show that Keccak and Luffa significantly outperform all other candidates. The authors didn't make any conclusions, but we point out that Keccak and Luffa are the only round 2 candidates adopting permutation-based (sponge and sponge-like) constructions. Although not strictly a hardware implementation aspect, Intel has recently released a new instruction set named AES-NI [7]. Hash functions based on AES, such as LANE, ECHO and Lesamnta, will benefit from these instructions significantly. However, in order for a hash function to make the most of these new AES-NI instructions, it should be based on an unmodified AES construction.

## 3. Intel Platform

The scope of this paper is restricted to software optimization on Intel platforms because these appear to be very widely spread nowadays. Currently, the most popular Intel processors are those of families descending from the Core architecture which introduced many revolutionary features to improve the processing performance, these features include:

- Wide Dynamic Execution. With Core Microarchitecture, each core can execute up to 4 instructions simultaneously. Intel also introduced Macro-fusion and Micro-fusion, which fuse micro-ops.
- Advanced Smart Cache. This feature allows each core to dynamically utilize up to 100% of the (fast) L2 cache if available, which was previously not possible resulting in an inefficient use of the cache.
- Advanced Digital Media Boost. This feature improved the execution of SIMD instructions by allowing the whole 128-bit instruction to be executed in one clock cycle, which wasn't possible previously.

### 3.1 SIMD Instruction Set

In SIMD (Single Instruction, Multiple Data), a single instruction operates on multiple data simultaneously achieving a data level parallelism. SIMD instructions are especially useful when the same operation needs to be executed repeatedly on different data. In 1997, Intel introduced MMX instruction set based on SIMD, which was later improved by introducing the SSE (Streaming SIMD Extensions). Later versions of SSE include: SSE2, SSE3, SSSE3, SSE4, SSE5 and recently AVX.

### 3.2 Registers in Intel Platforms

Registers are very fast storage mediums located near the processors. Below we provide brief descriptions of some register types found in most Intel platforms:

- General-purpose Registers: these are eight 32-bit registers (EAX, EBX, ECX, EDX, EBP, ESI, EDI, ESP) hold operands and memory pointers. In 64-bit mode, there are sixteen 64-bit registers.
- Segment Registers: these are six 16-bit registers (CS, DS, SS, ES, FS, GS) used to hold pointers.
- MMX Registers: these are eight 64-bit registers (MM0 – MM7) to perform operation on 64-bit data.
- XMM Registers: these are either eight (in 32-bit mode) or sixteen (in 64-bit mode) 128-bit registers (XMM0 – XMM15), introduced to handle the SSE 128-bit data types.
- EFLAG Register: this is a single 32-bit (in 32-bit mode) or 64-bit (in 64-bit mode) register used to reflect the results of the comparison instructions by setting its of flags appropriately.

The EFLAG register contains 1 control flag, 6 status flags, 11 system flags and the rest are reserved bits. Below we elaborate on some of EFLAG flags.

- Carry Flag (CF): set if a carry or borrow of an arithmetic operation is generated.
- Parity Flag (PF): set if the number of 1's in the least significant byte of the result from the previous operation is even (indicating even parity), otherwise it is unset (indicating odd parity).
- Zero Flag (ZF): set if the result of the previous operation is zero.
- Sign Flag (SF): used with signed values and is set to the same value of the sign bit (most significant bit), which is 0 if positive or 1 if negative.
- Overflow Flag (OF): set if the result of an operation doesn't fit the specified destination operand (e.g. storing a 64-bit value in a 32-bit register).

These flags are modified based on results of arithmetic operations and can be tested by other instructions by suffixing a "condition code" to the instruction. Some condition codes are described in table 1 [8]; note that some condition code mnemonics are synonym to others, such as B (Below) and NAE (Not Above or Equal).

Code	Description	Flag
O	Overflow	OF=1
NO	No Overflow	OF=1
B/NAE	Below/Not Above or Equal	CF = 1
NB/AE	Not Below/Above or Equal	CF = 0
E/Z	Equal/Zero	ZF = 1
NE/NZ	Not Equal/Not Zero	ZF = 0
BE/NA	Below or Equal/Not Above	CF $\vee$ ZF = 1
NBE/A	Not Below or Equal Above	CF $\vee$ ZF = 0
S	Sign	SF = 1
NS	No Sign	SF = 0
P/PE	Parity/Parity Even	PF = 1
NP/PO	Not Parity/Parity Odd	PF = 0
L/NGE	Less/Not Greater than or Equal	SP $\oplus$ OF = 1
NL/GE	Not Less than/Greater than or Equal	SF $\oplus$ OF = 0

Table 1

INTEL CONDITION CODES

## 4. Intel Optimization

Compilers, like the Intel C++ compiler, will certainly try to optimize the code, but they might sometimes make wrong decisions. To take advantage of the available powerful instructions, it is desirable to code directly in assembly where we have full control over the flow of the program. Programming in assembly is, however, tedious, time-consuming and error-prone. Therefore, it will only be worth coding the most critical and frequently called parts of the program in assembly. These critical parts may be a particular function that is being called very frequently, or a loop iterating many times. For example, the performance of hash functions may significantly be improved if its compression function (which is repeatedly called) is coded in assembly. Such critical parts can be spotted by running a performance analyzer/profiler software, like Intel's VTune.

When optimizing an implementation for a specific platform, and beside that it may not be optimal for other platforms, it also may not be optimal for different families from the same platform. That is, a particular set of optimization techniques targeted for Pentium processors, for example, may not be optimal for Core or Core 2 processors. Nevertheless, we personally believe that developing a generic code optimized for the Intel Core Microarchitecture is, in general, likely to be an optimal solution for most current and future processors but may not be so for older processor generations. Another solution would be to write multiple versions of a particular code where a programmer can explicitly write different versions of the same code, each optimized for a different processor, then, at the run time, the compiler uses the CPUID instruction to identify the platform it is running on and only compiles the appropriate version of the code. However, this approach is not efficient if there is a restriction on the code size.

In the follow sections we discuss a few optimization techniques [9], [8], [10], [11], [12], [13] developed especially to optimize code targeting Intel platforms and applicable

to most Intel Microarchitecture including Intel Core Microarchitecture, Enhanced Intel Core Microarchitecture and Intel Microarchitecture (Nehalem); though some of these techniques are generic and may also be applicable to other platforms.

## 4.1 Instruction Selection

Most of the unexpected results/errors are caused by poor selection of instructions. Choosing which instructions to use in a program may not only affect the execution of the program, but also its efficiency. In this section we briefly discuss a few considerations when selecting instructions in an Intel's platform.

### 4.1.1 Micro-operations

Each instruction is decoded into micro-operations (micro-ops) before it is executed. Intuitively, an instruction that decodes to less micro-ops runs faster. Thus, its recommended to use instructions that consist of fewer micro-ops if possible. Its also recommended to use a series of simple instructions that decode to fewer micro-ops, rather than using a single complex instruction decoding to more micro-ops.

### 4.1.2 INC/DEC instructions

In Intel, it is better to use the `ADD` (addition) and `SUB` (subtraction) instructions instead of `INC` (increment) and `DEC` (decrement) instructions because `INC/DEC` only update a subset of the flags in the `EFLAG` register. This is especially problematic when dealing with condition codes, which are dependant on the `EFLAG` register's flags. On the other hand, `ADD/SUB` instructions first clear all the flags in the `EFLGA` register before updating the appropriate ones based on the addition/subtraction result, so if any of the flags was previously set by an earlier instruction, it will be reset. This is not the case with `INC/DEC` since they only update the flags affected by the increment/decrement result and ignore the rest of the flag, which can create false dependency one previous instructions.

### 4.1.3 Shift/Rotate Instructions

Shift instructions are less expensive than rotate instructions. However, rotate by 1 has the same overhead as the shift instructions. Hence, it might be more efficient to use a series of rotate by 1 for small number of rotations.

### 4.1.4 CMP/TEST Instructions

`TEST` instruction ANDs its operands and updates the `EFLAG` register. If the result of the AND operation is not needed, then using `TEST` is better than using the `AND` instruction because `AND` wastes extra cycles to produce the result. `TEST` incurs less overhead than the `CMP` instruction and is preferred where possible. For example, using the `TEST` instruction on a register with itself is the same as

comparing the register to zero. It is also possible to compare values with zero by condition codes if the appropriate flags in `EFLGA` register were set by earlier instructions, in which case neither `TEST` nor `CMP` is needed.

## 4.2 Optimizing Branches

Branches (also called jumps) can greatly influence the performance of the program. Branches are points in the program where the flow of the program is interrupted and diverted. This jump from a point in the program to another certainly incurs more processing overhead. Branches present in assembly code and are comparable to the condition statements ('if' statements) in high-level languages. Branches are usually conditional (based on conditions), but they can sometimes be unconditional where the flow of the program always jumps to where the branch points to as soon as it reaches the branch; if the branch is executed, we say the branch is taken, otherwise, it is not taken, a technique called *branch prediction* predicts whether a branch is more likely to be taken or not taken. It is important to predict a branch especially in pipelined processors (these are almost all the modern processors) because when pipelining instructions, the address of the following instruction has to be known before the execution of the current one; that is, if a branch is to be taken, the following address is going to be the address of the first instruction at the portion of the code where the branch jumps to, otherwise, the following address is the address of the next sequential instruction after the branch instruction. However, even with efficient branch prediction mechanism, it is still advisable to minimize branches as much as possible because even if a branch has been correctly predicted, there is still an overhead for actually taking the branch. Below we discuss some guidelines for optimizing branches (though, we dont explicitly discuss any branch prediction algorithm).

It is always recommended that conditional codes are used instead of branches where possible. Conditional codes are dependent on the `EFLAG` register and are usually proceeded by `CMP` or `TEST` instructions which set the flags in `EFLAG` appropriately. In particular, the instructions `CMOVcc` or `SETcc` (where `cc` is the condition code) are preferred over branches (note that `SETcc` can only set operands to 1 or 0; if different values are required, `CMVcc` is used). For example, consider the following C code:

```
if (x >= y) {x = 1;} else {x = 0;}
```

In assembly, this can be written as:

```
CMP eax, ebx    ;compare values of eax and ebx
Jge Gtr        ;if eax >= ebx, jump to 'Gtr'
MOV eax 0      ;otherwise, set eax = 0
JMP Less      ;jump to 'Less' where the
               ;rest of the program is

Gtr:
Mov eax 1      ;set eax = 1
Less: . . .
```

Compare this with the following optimized code:

```
CMP    eax, ebx    ;compare  eax and ebx
CMOVge eax, #1    ;if  eax >= ebx,  eax = 1
CMOVL  eax, #0     ;otherwise,  eax = 0
```

### 4.3 Optimizing Loops

With the advent of Core Microarchitecture, Intel introduced the Loop Stream Detector (LSD) which expedites the execution of loops containing up to 18 instructions. When a loop is detected, the loop instructions are buffered in a special LSD buffer and the fetch and branch prediction stages are powered off until the loop is completed. Intel Microarchitecture (Nehalem) further improved LSD by moving it beyond the decode stage to hold up to 28 micro-ops for immediate execution, powering off all the pipeline stages except the execute stage; here, the LSD buffer is similar to the trace cache<sup>4</sup>. Even with the presence of LSD, the implementer can still further optimize loops. In the following sections we introduce a few generic loop optimization techniques that can be tuned for other platforms (not just Intel).

#### 4.3.1 Loop unrolling

Unrolling a loop entails reducing the number of loop iterations by increasing the size of the loop body. Since each loop incurs extra overhead for checking the end-of-loop condition at the end of every iteration, minimizing the number of iterations does greatly optimize its execution. Unrolling a loop, however, increases the size of the code and may congest the trace cache. Thus, it is recommended to only unroll the frequently called loops. Intel recommends that a loop should not be iterated more than 16 times, and if it does, it should be unrolled to keep this maximum number of iterations [9]. For example, suppose that the operations `Operation1` and `Operation2` need to be executed 8 times each, this can be written as:

```
for (i = 0; i < 8; i++)
    {Operation1; Operation2;}
```

with unrolling, the above code can be re-written as:

```
for (i = 0; i < 4; i++){
    Operation1; Operation2;
    Operation1; Operation2;}
```

This potentially saves four iterations and consequently four end-of-the-loop condition checks. For loops that iterate many times, saving the end-of-loop condition does significantly improve the speed of the execution.

#### 4.3.2 Loop-blocking

A very effective technique for optimizing loops is loop-blocking. This technique is useful when dealing with large amount of data. If the data on which the loop is operating is large, the cache might not be sufficient to hold the data during the whole execution time; then the slow memory

<sup>4</sup>Trace cache is part of the first-level cache and used to hold the decoded instructions before execution.

access is required. In this case, using loop blocking allows for partitioning the loop into smaller chunks to operate on data with size small enough to fit in the cache. These chunks are then executed in turn such that the cache is reused every time a new chunk is executed. Consider the following example with a cache of size 8 bytes (the cache can store 8 bytes at any given time, any extra values are stored in memory):

```
for (i = 0; i <= 16; i++){Function1(x[i]);}
for (i = 0; i <= 16; i++){Function2(x[i]);}
```

Since the cache can only store 8 bytes, when executing `Function1` only 8 bytes of array `x` can be stored in the cache, the other 8 bytes will be stored in memory and will need to be loaded for `Function2`. Compare this with the following code after applying loop-blocking:

```
for (i = 0; i < 2; i+=8){
    for (j = i; j < min(16,8+i); j++){
        Function1(x[j]);}
    for (j = i; j < min(16,8+i); j++){
        Function(x[j]);}}
```

Here, instead of operating on the 16 bytes of the array `x` at once, the loop is divided into two portions of size 8 each. `Function1` operates on the first 8 bytes of array `x` which fit in the cache, then the same 8 bytes are transferred to `Function2` to be operated on. Once both `Function1` and `Function2` finish executing the first 8 bytes of array `x`, the cache is purged for the other 8 bytes of array `x` and the same process is repeated.

#### 4.3.3 Decrementing Loop

When writing a loop, it is more efficient to use a decrementing loop rather than an incrementing one. An incrementing loop requires 3 instructions: an addition instruction `ADD` to increment the loop counter, a compare instruction `CMP` to compare the loop counter to the maximum value at which the loop should terminate and, a conditional branch `JLE` to iterate through the loop. On the other hand, a decrementing loop will only need two instructions: a subtraction instruction `SUB` to decrement the loop counter, and a conditional branch instruction `JNZ`; a comparison instruction is not needed in this case since `SUB` will set the condition flag `ZF` in the `EFLAG` register when it reaches zero (end of loop) which is then tested by the conditional branch instruction `JNZ` before iterating through the loop. For example, the program below implements a loop that calculates 5! (5 factorial).

```
for (i = 1; i <= 5; i++) { factorial *= i; }
```

which will be translated into assembly as follows:

```
MOV  eax, #1        ;eax=1
MOV  ebx, #1        ;ebx=1
Loop:
MUL  eax, eax, ebx  ;eax = eax * ebx
ADD  eax, eax, #1   ;eax++
CMP  eax, #0x05     ;eax = 5 ?
JLE  Loop          ;branch if eax <= 5
```

but, if we rewrite using a decrementing loop:

```
for (i = 5; i >=1; i--){ factorial *=i; }
```

the assembly will be translated as follows:

```

MOV eax, #1           ;eax=1
MOV ebx, #1           ;ebx=1
Loop:
MUL eax, eax, ebx     ;eax = eax * ebx
SUB ebx, ebx, #1      ;eax--
JNZ Loop              ;branch if ZF != 0

```

saving one instruction (one cycle) per iteration.

#### 4.4 Optimizing Functions

If a particular function is frequently called, it is recommended to inline that function. Inlining a function involves creating a local copy of the function inside the calling program and thereby eliminating the overhead of jumping outside the program and back again repeatedly while calling it. Like loop unrolling, inlining a function increases the code size; hence, it is advisable that only small functions are inlined for an implementation targeting a memory-constrained platforms. For example, to optimize a hash function, it would worth inlining only its corresponding compression function or small parts of the compression function if they are being heavily used during the hashing process.

Moreover, the decision of whether *calling* a function or *jumping* to it can affect the efficiency. When calling a function more overhead is incurred because it requires a return and the return address should be saved in the stack. On the other hand, when jumping to a function, neither is required. Therefore, if a return from a function is not necessary, it is recommended to jump to that function rather than calling it.

#### 4.5 Optimization for SIMD

The following techniques apply to processors supporting MMX and SSE instruction sets; these are (mostly) processors based on Intel Core Microarchitecture, Enhanced Intel Core Microarchitecture and Intel Microarchitecture (Nehalem) as well as a few Pentium processors. For the code to benefit from the SSE instructions, it has to be *vectorized*. Vectorization is the process of parallelizing the code to take advantage of the inherent parallelism of SSE instructions. For example, four 32-bit double words<sup>5</sup> can be stored in a single 128-bit XMM registers for a single SSE instruction to operate on simultaneously.

There are several ways in which a code is optimized for MMX/SSE technologies. The most straightforward way is to code directly in assembly. This can be either a standalone assembly implementation or an assembly embedded in a C/C++ code using inlined assembly extension to C/C++. Another way is to use intrinsic functions, which can directly access the SSE instructions using sufficiently large data types (e.g. `_m128` which is 128-bit long). These functions

<sup>5</sup>In XMM registers a byte is 8-bits, a word is 16-bits, a double word is 32-bit, a quadword is 64-bit and a double quadword is 128-bit, that is, a single XMM register consists of either 16 bytes, 8 words, 4 double words, 2 quadwords or 1 double quadword.

are built-in to the compiler and will be inlined at compilation time. However, intrinsic functions are not portable and compiler-dependant. Alternatively, special classes [9] implemented for this purpose can also be used, which are, again, compiler-dependant.

Another important consideration when dealing with MMX/SSE instructions (or any instruction set in general) is data alignments. It is important to align the MMX operands to 64-bit boundaries to fit on the 64-bit MMX registers because it is very expensive to operate on unaligned data. Similarly, SSE operands should be aligned to 128-bit boundaries to fit on the 128-bit XMM registers. One way to align unaligned data is to pad the operands appropriately. Also, in some cases, rearranging the data may help in aligning it. Carefully rearranging data of different sizes (and types) assigned to structures is an example of such practice.

### 5. Performance Evaluation

Instructions are executed in clock cycles, which are the fundamental units of the CPU clock rate. CPU's that operate on 2.0 GHz clock rate, for example, execute  $2 \times 10^9$  instructions per second. Therefore, the performance of a particular code is usually evaluated by counting the clock cycles the CPU wasted in executing it; the fewer these cycles, the more efficient the code. In this section, we briefly describe the most common ways of counting CPU cycles in Intel platforms. When evaluating the efficiency of hash functions, we usually count cycles/byte. In order to calculate how many cycles each byte of the message takes to be hashed, the overall number of clock cycles counted for hashing the whole message is divided by the message size in bytes.

To count the clock cycle in Intel platforms, the RDTSC (Read Time Stamp Counter) instruction is used. RDTSC indicates how many clock cycles the CPU has executed since it was powered up or reset. There are a number of header files that can be used to inline the RDTSC instruction, we adopt `cycle.h`<sup>6</sup>, which uses the function `getticks()` to record cycle readings and then the function `elapsed(.,.)` to subtract readings and get the actual number of cycles wasted during execution.

```

t1 = getticks();      //take first reading
    run hash function ...
t2 = getticks();      //take second reading
t3 = elapsed(t2,t1); //get the difference

```

When executing the code, the CPU is not exclusively reserved for this executing process; instead, in reality, it usually executes other processes, such as OS transactions etc. in parallel. Consequently, if we count the clock cycles as above only once, it is very likely that the obtained result will also include other delays not caused by the executing code. One way to improve the accuracy of the clock counting process is to execute the code (i.e. hash function) several times, accumulate the clock cycle readings, and then take the average. The pseudocode below illustrates this procedure:

<sup>6</sup>available from: [www.fftw.org/cycle.h](http://www.fftw.org/cycle.h) (accessed March 2011).

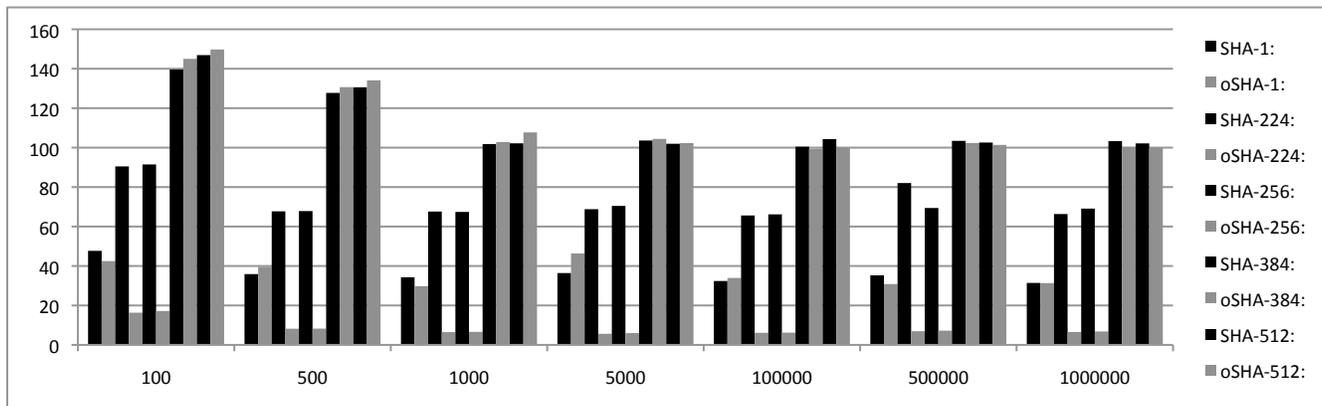


Fig. 1

PERFORMANCE IMPROVEMENT WITH AES-NI (ESTIMATED)

```

for (x=0; x < MAX; x++) {
t1 = getticks();
    run hash function ...
t2 = getticks();
t3 += elapsed(t2 - t1)    //accumulate
} t4 = t3 / MAX; //average

```

## 5.1 Demonstration

As a demo, we adopted Brian Gladman's implementations<sup>7</sup> of SHA1 and SHA2 and very slightly optimized them, then compared our optimized implementations to Gladman's original ones. The experiments were carried out on Intel Core 2 Duo 3.00 GHz, 4.00 GB of RAM, running 32-bit Windows Vista Home Premium, and using GNU GCC compiler, while hashing messages with sizes ranging from 100 to 1000,000 Bytes. Figure 1 plots the results. As shown in the figure, apart from some discrepancies in SHA-384 and SHA-512 implementations for short messages, we observe a general efficiency improvement, especially in SHA-224 and SHA-256 since these handle data in 32-bit words, which suits our testbed OS. We also observe that the hashing rate is higher in large messages since the processing overhead of the message initialization and hashing finalization fades out. In this experiment, we deliberately only made minor optimization on the loops (without unrolling them and so maintaining the original code size) to observe what effect even minor optimization can have on the efficiency of the code. We expect even higher efficiency gain should more optimization techniques are applied, but that may trade off code size, coding effort etc.

## 6. Conclusion

While security is certainly of high importance, for a hash function to be practical in some applications, it may also need to be efficient. This efficiency requirement has largely

been overlooked in the literature. In this paper, we not only show how to evaluate the efficiency of hash functions, we also discuss how to improve it. Although we present a set of optimizations techniques considering Intel platforms, most of these are generic and apply to a wide range of platforms. Future and current work will investigate similar optimization techniques on other widely used platforms, such as ARM and AMD.

## References

- [1] S. Mathew and K. P. Jacob, "Performance Evaluation of Popular Hash Functions," *World Academy of Science, Engineering and Technology*, vol. 61, pp. 449–452, 2010.
- [2] N. Ferguson, S. Lucks, B. Schneier, D. Whiting, M. Bellare, T. Kohno, J. Callas, and J. Walker, *The Skein Hash Function*, 2008, <http://www.skein-hash.info>.
- [3] R. Anderson, E. Biham, and L. Knudsen, *Serpent: A Proposal for the Advanced Encryption Standard*, 1997, <http://www.cl.cam.ac.uk/~rja14/serpent.html>.
- [4] J. Daemen and V. Rijmen, *AES Proposal: Rijndael*, 1997, <http://www.daimi.au.dk/~ivan/rijndael.pdf>.
- [5] C. Wenzel-Benner and J. Graf, *eXternal Benchmarking eXtension*, 2009, (manuscript under preparation).
- [6] S. Tillich, M. Feldhofer, M. Kirschbaum, T. Plos, J.-M. Schmidt, and A. Szekely, *High-Speed Hardware Implementations of BLAKE, Blue Midnight Wish, CubeHash, ECHO, Fugue, Grstl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD, and Skein*, Graz University of Tehnology, 2009, ePrint: <http://eprint.iacr.org/2009/510.pdf>.
- [7] R. Benadjila, O. Billet, S. Gueron, and M. Robshaw, "The Intel AES Instructions Set and the SHA-3 Candidates," in *Asiacrypt '09*, ser. LNCS, vol. 5912. Springer, 2009, pp. 162–178.
- [8] *Intel 64 and IA-32 Architectures Software Developers Manual - Vol. 1 Basic Architecture*, Intel Corp., 2008.
- [9] *Intel 64 and IA-32 Architectures Optimization Reference Manual*, Intel Corp., 2008.
- [10] *Intel 64 and IA-32 Architectures Software Developers Manual - Vol. 2A Instruction Set Reference A-M*, Intel Corp., 2008.
- [11] *Intel 64 and IA-32 Architectures Software Developers Manual - Vol. 2B Instruction Set Reference N-Z*, Intel Corp., 2009.
- [12] *Intel 64 and IA-32 Architectures Software Developers Manual - Vol. 3A System Programming Guide, Part 1*, Intel Corp., 2009.
- [13] *Intel 64 and IA-32 Architectures Software Developers Manual - Vol. 3B System Programming Guide, Part 2*, Intel Corp., 2008.

<sup>7</sup>Available from: [www.gladman.me.uk](http://www.gladman.me.uk) (accessed Mar. 2011).

# Modern Hash Function Construction

B. Denton<sup>1</sup> and R. Adhami<sup>1</sup>

<sup>1</sup>Dept. of Electrical & Computer Engineering, *The University of Alabama in Huntsville*, Huntsville, AL, USA

**Abstract-** This paper discusses modern hash function construction using the NIST SHA-3 competition as a survey of modern hash function construction properties. Three primary hash function designs are identified based on the designs of SHA-3 candidates submitted as part of the NIST SHA-3 competition. These designs are Wide-pipe, Sponge, and Hash Iterated FrAmework (HAIFA).

**Keywords-** cryptography; hashing; hash function

## 1 Introduction

Modern secure hashing algorithms are critically important to the integrity and non-repudiation of information and data in many different computer systems. The most widely used cryptographic hash functions, MD5 and SHA-1, have considerable weaknesses [1,2]. The National Institute of Standards and Technology (NIST) is currently holding an international competition to select the next generation secure hashing algorithm, called SHA-3. This paper covers the construction properties of modern cryptographic hash function as well as the security requirements that motivate these construction properties. After an overview of cryptographic hash function security properties and attacks, we will discuss three primary classifications of modern hash function construction: Wide-pipe, Sponge function, and the Hash Iterated FrAmework (HAIFA).

## 2 Hash function security properties and attacks

Cryptographic hash functions play a fundamental role in modern cryptography, specifically in the areas of message authentication, data integrity, digital signatures, and password schemes. In general, a hash function

$$h : \{0,1\}^* \rightarrow \{0,1\}^n \quad (1)$$

maps an arbitrary finite sized binary input message  $m$  to a fixed sized,  $n$ -bits, binary output called the *hash value*, *message digest*, or simply *hash*. (Figure 1) For a cryptographic hash functions, the hash value serves as a unique and fixed-sized representation of the unique message input. Unfortunately we have one big problem. Given a domain  $D$  and range  $R$  with  $h : D \rightarrow R$  and  $|D| > |R|$  implies that *collisions* are inevitable, where multiple inputs map to the same output. [3]

An acceptable solution is to design a cryptographic hash function (simply hash function from here forth) in such a way that a collision is difficult to find. In other words, finding a collision should be computationally infeasible. In the requirements for NIST's SHA-3 competition, NIST notes that  $2^{80}$  work is considered too small of a security lower boundary and requires all SHA-3 candidates to have a higher security boundary. [4] This measure is simply the number of calls to the hash function an attacker would have to make in order to find a collision. Although not an ideal definition, computational infeasible today means that more than  $2^{80}$  work is required to find a collision in a hash function.

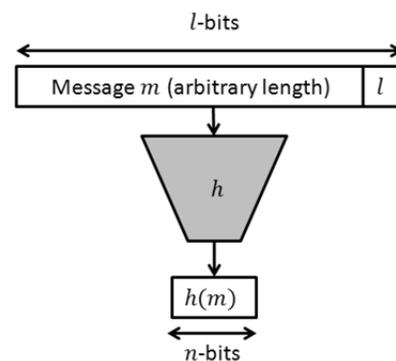


Figure 1 Hash Function

Three distinct attack methods exist for finding a collision in a hash function based on the goals of the attacker. These methods form the fundamental security properties of a hash function. The basic properties of a hash function are:

1. *Preimage resistance* - given an output  $h(m)$ , it is difficult to find  $m$ . See Figure 2.
2. *Second-preimage resistance* - given a specific input  $m_1$  with an output  $h(m_1)$ , it is difficult to find another input  $m_2$  such that  $h(m_1) = h(m_2)$ . See Figure 3.
3. *Collision resistance* - it is difficult to find two messages  $m_1$  and  $m_2$  such that  $h(m_1) = h(m_2)$ . (Note: There is free choice for both messages.) See Figure 4.

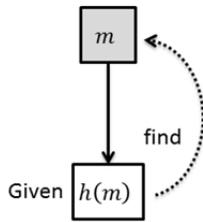


Figure 2 Preimage resistance

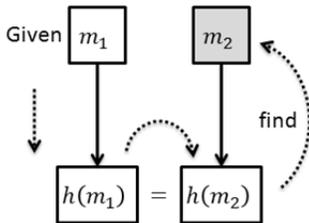


Figure 3 Second-preimage resistance

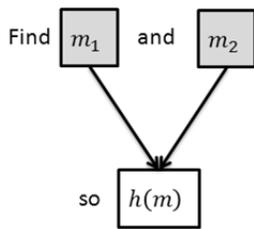


Figure 4 Collision resistance

Preimage resistance has a fixed output value, Second-preimage resistance has a fixed input value, and the attacker seeks to find another colliding input message for either of these values. For collision resistance, the attacker does not care what the two input message are, only that they “hash” to the same output.

A common security model used to describe the ideal hash function is the random oracle. Visualize the random oracle as an elf sitting in a box with a source of physical randomness and some means of storage. A common explanation uses dice and a scroll. The elf will accept queries from anyone and will look in the scroll to see if an entry exists for that query. The elf will answer queries from anyone, both friendly and foe. If the query exists, the elf will respond with the recorded result. If the query does not exist, the elf will throw the dice, record the randomized result, and respond. The elf can only work so fast and thus has a limited amount of queries that can be answered every second. The end result is a “perfect” one-way function. [5]

In order for a hash function to have an acceptable security level it should behave like a random oracle. The minimum amount of work required by an attacker to violate the preimage or second-preimage resistance property for an  $n$ -bit output hash function should be  $2^n$ . The minimum amount of work to violate the collision resistance property (due to the birthday paradox [6]) should be  $2^{n/2}$ . [7] For

example, SHA-1 has a 160-bit output, so any attack that finds a preimage or second preimage in less than  $2^{160}$  or a collision in less than  $2^{80}$  demonstrates that SHA-1 provides less security than a random oracle. In fact, a collision attack exists against SHA -1 that only requires  $2^{69}$ work [1] which was one of the largest motivating factors for NIST to form the SHA-3 competition to select a new standard hash function.

Table 1 Minimum Security Requirements for a Hash Function

Attack	Security Boundary
Preimage	$2^n$
Second-Preimage	$2^n$
Collision	$2^{n/2}$

The hash functions, MD5 and SHA-1, use a classic Merkle-Damgård construction [8,9] which, in general, splits the input message into  $r$  equal sized  $m$ -bit message blocks ( $m_0 \dots m_{r-1}$ ) padding the last block as necessary and appending the message length, then iterates through each message block applying a compression function

$$c : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^n. \quad (2)$$

The input to the compression function is the previous compression function output  $CV_{i-1}$  and the current message block  $m_i$ . The compression function output  $CV_i$  is called the Chaining Value and the initial chaining value  $CV_0$  is called the Initialization Vector or IV. This process is show in Figure 5 below.

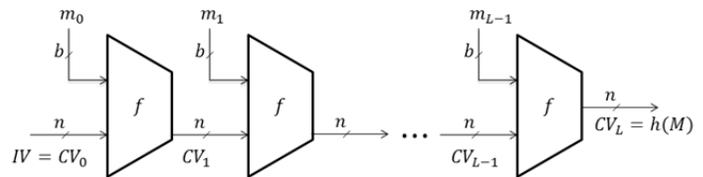


Figure 5 Iterated Hash Function

Unfortunately, due to the iterative nature of classic Merkle-Damgård (MD) construction certain generic attacks exist that differentiate an MD hash function from a random oracle [5] such as the multicollision attack [10] and length extension attacks.

Imagine a collision finding function  $C$  that takes an initialization vector ( $CV_0 = h_0$ ) and outputs two message blocks  $m_1 \neq m'_1$  that both hash to the same value

$$h_1 = f(m_0, h_0) = f(m'_0, h_0). \quad (3)$$

The amount of work required to find this collision is  $2^{n/2}$ . Set  $CV_1 = h_1$  and the collision finding function will find two more messages,  $m_1$  and  $m'_1$ , that also collide which requires another  $2^{n/2}$ . Thus for finding  $k$ -colliding pairs (Figure 6) of message blocks that all hash to the same final value  $h_k$ , the attacker only has to expend  $k \times 2^{n/2}$  effort instead of the expected  $2^{n(2^k-1)}/2^k$  effort if the hash function was truly random. This attack also leads to other serious attacks such as

the long message second-preimage attack [11] and the herding attack [12].

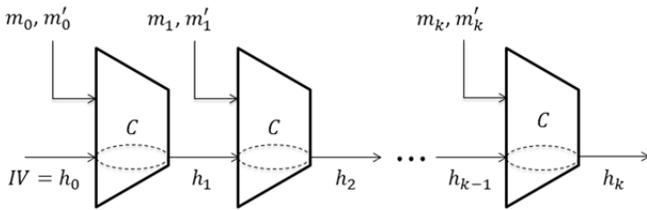


Figure 6 Multicollisions in Merkle-Damgård construction

Another problem with MD based hash functions is length extension. A message  $m$  is divided into  $m_0, m_1, \dots, m_k$  blocks and hashed to a value  $H$ . Now choose another message  $m'$  that divides into  $m_0, m_1, \dots, m_k, m_{k+1}$  blocks. Since  $m$  and  $m'$  share the same first  $k$  blocks, the hash value  $h(m)$  is simply the intermediate hash value after  $k$  blocks when computing  $h(m')$ . This is shown in Figure 7 below. This certainly is not a property of the ideal hash function and creates serious problems in practice. [13]

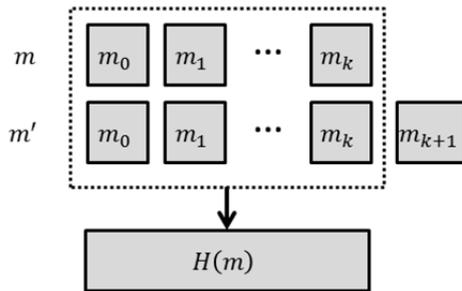


Figure 7 Length Extension

Merkle and Damgård [8,9] proved that a collision resistant compression function implies a collision resistant hash function but due to internal collisions caused by the iterative nature of MD construction, these additional attacks are possible. A random oracle does not have these weaknesses. These generic attacks have motivated cryptographic researchers to find new methods of designing hash functions.

### 3 State-of-the-art hash function design

The NIST SHA-3 competition began in November 2008 with the first round consisting of 51 candidate hash functions. In August 2009, the second round cut the competition down to 12 candidates. The final round began in December 2010 with 5 finalists. Nearly all these candidate hash functions can be broken down into one of three general categories: wide-pipe, sponge functions, and HAIFA.

#### 3.1 Wide-pipe

Wide-pipe hash function construction is designed to overcome the multicollision attack and length extension attack

by trying to prevent internal collisions. [14] In simple terms, this is accomplished by increasing the size of internal state of the hash function. For an  $n$ -bit output hash function with a  $w$ -bit  $CV$ ,  $w > n$ . (Figure 8) In order to obtain the  $n$ -bit output, an output transformation is performed on the final  $CV_{r-1}$ . This could be just discarding bits or some other more complicated function.

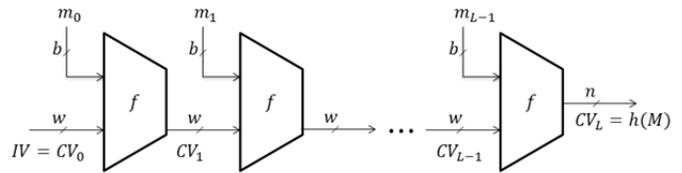


Figure 8 Wide-pipe Construction

#### 3.2 Sponge Functions

The next class of hash function construction, sponge functions [15], was designed in such a way to mimic a random oracle. In general terms, a random oracle

$$R : \{0,1\}^* \rightarrow \{0,1\}^\infty \tag{4}$$

maps a variable-length input message to an infinite output string. It is also completely random; the output bits are uniformly and independently distributed. The requirement of a random oracle that makes it suitable for a proper hash function security model is that identical input messages generate identical output strings. A secure hash function with an  $n$ -bit hash value should behave exactly like a random oracle whose output is truncated to  $n$ -bits.

Again, due to the iterative nature of classic Merkle-Damgård construction, internal state collisions exist in the chaining values,  $CV_i$ . State collisions introduce properties that do not exist for a random oracle such as the length extension attack. For example, consider that  $M_1$  and  $M_2$  are two messages that form a state collision. For any suffix  $N$ , the messages  $M_1|N$  and  $M_2|N$  will have identical hash values. State collisions alone are not a problem but they do lead to a differentiator from a random oracle. The random oracle security model is an unreachable goal for an iterated hash function. Instead of abandoning iterated hash functions, Bertoni et al. [15] designed the sponge function construction as a new security model that is only distinguishable from a random oracle by the presence of internal state collisions.

Sponge construction consists of a fixed-length permutation on a fixed number of bits,  $r + c$ , shown in Figure 9 below [15]. It can be used to create a function with variable-sized input and arbitrary length output.

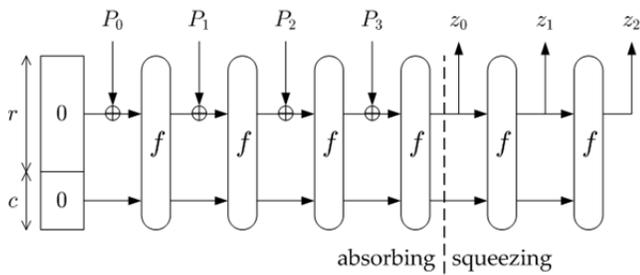


Figure 9 Sponge Construction

The value  $r$  is called the bitrate and the value  $c$  is the capacity. Both are initially set to zero. The input message is first padded and split into  $r$ -bit length message blocks. The sponge construction consists of two stages, absorbing and squeezing. In the absorbing stage,  $r$ -bit sized blocks of the input message ( $P_i$  in Figure 9) are XORed with the first  $r$ -bits of the state, followed by an application of the permutation function  $f$ . This is repeated until all message blocks are processed. In the squeezing stage, the first  $r$ -bits are returned as output of the sponge construction, again followed by applications of the permutation function  $f$ . The length of the squeezing phase is user-specified depending on the desired output length.

### 3.3 HAIFA

The Hash Iterative FrAmework (HAIFA) [16] design solves many of the internal collision problems associated with the classic MD construction design by adding a fixed (optional) salt of  $s$ -bits along with a (mandatory) counter  $C_i$  of  $t$ -bits to every message block in the iteration  $i$  of the hash function. Wide-pipe and HAIFA are very similar designs. The counter  $C_i$  keeps track of the number of message bits hashed so far. The HAIFA design is both prefix- and suffix-free and as a consequence is collision resistant and indifferentiable from a random oracle. [17] This design is also proven secure against  $2^{\text{nd}}$ -preimage attacks if the underlying compression function behaves like an ideal primitive. [17] Figure 10 below shows the construction of the compression function. This can also be expressed as

$$CV_i = C(CV_{i-1}, m_i, C_i, S). \tag{5}$$

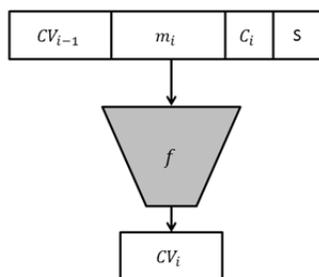


Figure 10 HAIFA Construction

## 4 Discussion

This paper covered three general secure hash function construction methods represented by various hash functions specifications in the NIST SHA-3 competition. All of these are general descriptions and do not represent an actual hash function. Instead, these construction methods serve the purpose of describing general solutions to some serious attacks against classic Merkle-Damgård hash function construction.

## 5 References

- [1] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu, "Finding Collisions in the Full SHA-1," in *Proceeding of CRYPTO'05*, vol. 3621, 2005, pp. 17-36.
- [2] Marc Bevand, "MD5 Chosen-Prefix Collisions on GPUs," in *Black Hat USA*, Las Vegas, 2009.
- [3] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*.: CRC Press, 1996.
- [4] National Institute of Standards and Technology, "Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA-3) Family," *Federal Register*, vol. 72, no. 212, pp. 62212-62220, November 2007.
- [5] Mihir Bellare and Phillip Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proceedings of the 1st ACM conference on Computer and Communications Security*, 1993, pp. 62-73.
- [6] Gideon Yuval, "How to Swindle Rabin," *Cryptologia*, vol. 3, no. 3, pp. 187 - 191, 1979.
- [7] William Stallings, *Cryptography and Network Security: Principles and Practices 5th Ed.*, 4th ed.: Prentice Hall, 2010.
- [8] Ivan Damgård, "A Design Principal for Hash Functions," in *Proceedings of CRYPTO '89*, vol. 435, 1989, pp. 416-427.
- [9] Ralph Merkle, "One Way Hash Functions and DES," in *Proceedings of CRYPTO'89*, vol. 435, 1989, pp. 428-446.
- [10] Antoine Joux, "Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions," in *Proceedings of CRYPTO'04*, vol. 3152, 2004, pp. 306-316.

- [11] John Kelsey and Bruce Schneier, "Second Preimages on  $n$ -Bit Hash Functions for Much Less than  $2^n$  Work," in *Proceedings of EUROCRYPT'05*, vol. 3494, 2005, pp. 474-490.
- [12] John Kelsey and Tadayoshi Kohno, "Herding Hash Functions and the Nostradamus Attack," in *Proceedings of EUROCRYPT'06*, vol. 4404, 2006, pp. 183-200.
- [13] Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno, *Cryptographic Engineering: Design Principles and Practical Applications.*: Wiley Publishing, Inc., 2010.
- [14] Steven Lucks, "Design Principles for Iterated Hash Functions," Cryptology ePrint Archive, 2004. [Online]. <http://eprint.iacr.org/2004/253>
- [15] Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche, "Sponge Functions," in *ECRYPT Hash Function Workshop*, 2007.
- [16] Eli Biham and Orr Dunkelman, "A Framework for Iterative Hash Functions - HAIFA," Cryptology ePrint Archive, 2007. [Online]. <http://eprint.iacr.org/2007/278>
- [17] Andreeva, Elena; Mennink, Bart; Preneel, Bart, "Security Reductions of the Second Round SHA-3 Candidates," Cryptology ePrint Archive, 2010. [Online]. <http://eprint.iacr.org/2010/381>



**SESSION**  
**CYBERSECURITY EDUCATION**

**Chair(s)**

**Prof. George Markowsky**



# DefEX: Hands-On Cyber Defense Exercises for Undergraduate Students

Sonja M. Glumich and Brian A. Kropa

Cyber Sciences Branch, Air Force Research Laboratory, Rome, NY, USA

**Abstract** - *DefEX incorporates a set of hands-on cyber security exercises aimed at developing problem-solving proficiency, teamwork, and cyber defense skills in undergraduate students. The exercises include Code-Level and System-Level Hardening, Static and Dynamic Reverse Engineering, Detect and Defeat, Digital Forensics, and the Wireless Access Point Treasure Hunt. Providing a diverse group of students with a common set of foundational knowledge and finding the balance between enabling participation of novice students and generating problems complex enough to challenge experienced students posed the major curriculum design risks. Instructors reduced the risks by administering a technical survey, requiring students to complete a set of fundamental exercises, and assigning balanced student teams. As a result, student teams successfully completed all of the exercises.*

**Keywords:** Cyber Defense Exercise, Cyber Security Curriculum, Cyber Security Education

## 1 Introduction

DefEX consists of hands-on cyber security exercises designed to promote problem-solving proficiency, teamwork, and cyber defense skills in undergraduate students. Students worked in teams of three to complete the following exercises: Code-Level and System-Level Hardening, Dynamic and Static Reverse Engineering, Detect and Defeat, Digital Forensics, and the Wireless Access Point Treasure Hunt.

DefEX falls under the full spectrum of cyber defense activities including preventing attacks, detecting attacks, surviving attacks, and recovering from attacks. The Code and System-Level Hardening exercises address proactively preventing attacks before they occur, the Detect and Defeat exercise involves discovering and countering attacks, the Forensics exercise involves enabling recovery efforts by analyzing the aftereffects of attacks, and the Reverse Engineering exercises address analyzing malicious code to aid in surviving and recovering from attacks.

Student participants were rising seniors studying Computer Science, Computer Engineering, Electrical Engineering, Mathematics, and Physics at academic institutions across the country. The major curriculum design challenge involved

enabling participation of novice students and generating problems complex enough to challenge experienced students. Students all possessed one semester of computer programming in a high level language, a semester of discrete math, and a year of calculus. While enforcing the set of minimum academic requirements helped, instructors took additional steps to account for the varied backgrounds of students.

Accommodations included creating a technical survey and a set of fundamental exercises and assigning balanced student teams. Issuing the technical survey to students before the exercises to assess prior experience in areas such as networking, computer security, and digital logic helped instructors tailor background materials for the exercises. Requiring students to complete a set of fundamental exercises before the DefEX exercises ensured a common familiarity with VMware server, the Linux operating system, networking protocols, and the client-server model. Assigning students into teams to maximize the diversity of college majors created balanced teams performing on near-equal footing. For example, a team might consist of a mathematics student, a computer science student, and a computer engineering student.

## 2 Code-Level and System-Level Hardening

### 2.1 Background

Code-level hardening involves activities undertaken by software developers or testers to produce secure source code. System-level hardening includes actions carried out by users or operators to securely configure an existing system. The software development lifecycle (SDLC) serves as a beneficial contextual framework for teaching secure application design, implementation, testing, and hardening. The instructor presented the SDLC in seven phases: 1) Specify (Requirements), 2) Architect, 3) Design, 4) Implement, 5) Test, 6) Deploy, and 7) Maintain. Software developers and testers engage in code-level hardening mainly during the Implement and Test phases, and operators conduct system-level hardening during the Deploy and Maintain phases.

One point reinforced by examining security activities in the context of the SDLC was that code-level and system-level hardening activities only eradicate a limited number of vulnerabilities introduced in hardware or in the early SDLC phases. Code-level hardening has a limited ability to remove vulnerabilities introduced during the Specify, Architect, or

Design phases. System-level hardening is additionally restricted in fixing Implement and Test phase vulnerabilities. Conversely, developers and testers have a limited amount of influence in helping users to securely configure a system in the Deploy and Maintain phases. Designing a user friendly system defaulting to strict security settings helps, but doesn't guarantee secure configuration.

This examination contradicts the validity of the practice of blaming the user for current cyber security woes. Most users lack the ability to remediate architectural, design, or code-level vulnerabilities such as those existing in proprietary operating system binaries. It also belies the notion that if cyber operators are diligent enough while monitoring packets and intrusion detection system (IDS) alerts they can successfully thwart all attacks during the Deploy and Maintenance phases. Although there exist many knowledgeable, committed cyber operators, by the time they deploy a majority of systems, the cyber security battle was already lost earlier in the SDLC.

## 2.2 Exercise Goals

The aim of the Code-Level and System-Level Hardening exercises was for students to grasp the complexities and manpower involved in developing a secure system. In the context of the SDLC, students should have comprehended that requirements engineers, architects, designers, developers, testers, operators, and users all play essential roles in producing and maintaining secure systems.

## 2.3 Code-Level Hardening Description

After a short lecture introducing foundational concepts including the SDLC, system-level hardening, and code-level hardening, students examined the vulnerable application "Madam Zora" (Zora). The instructor designed Zora, a custom web application, specifically for the Code-Level and System-Level Hardening exercises.

Students tested four vulnerabilities exhibited by the web application: 1) Cross-Site Scripting (XSS), 2) Structured Query Language (SQL) Injection, 3) Command Injection, and 4) File Upload. Next, the students patched the associated flawed Perl and PHP Hypertext Preprocessor (PHP) code. Finally, students retested the vulnerabilities to ensure the coding changes fixed the vulnerabilities. The secure programming practice take away from this exercise was to filter all user-influenced input and output for web applications. To accommodate students of varying programming expertise, the instructor led the testing, patching, and retesting of the four vulnerabilities, conducting frequent checks for understanding and inviting questions.

### 2.3.1 Cross-Site Scripting

XSS occurs when users manipulate web application input to execute client-side commands on a system. A well-known test for XSS entails inputting the JavaScript *alert*

command into a web application text field. Students tested Zora text fields by entering the following command:

```
<script language='javascript'>alert('Zora!');</script>
```

If an alert box containing *Zora!* appeared, students established the XSS vulnerability of the underlying script. The Zora XSS vulnerability existed in a PHP file that echoed unfiltered user input back to the screen. To eliminate the vulnerability, students filtered the input using the PHP *htmlentities* function and retested the code. The *htmlentities* function translates certain ambiguous characters into their corresponding character entity references. For example the '<' character becomes '&lt;'. This prevents inputted JavaScript commands from being evaluated. The vulnerable Zora code outputting unfiltered user input is shown below:

```
$unfiltered['input'] = ($_GET['user_input']);
<?php echo $unfiltered['input']?>
```

The fixed Zora code outputting input filtered with *htmlentities* is as follows:

```
$unfiltered['input'] = ($_GET['user_input']);
$filterd['input'] = htmlentities($unfiltered['input']);
<?php echo $filterd['input']?>
```

### 2.3.2 Command Injection

Command injection occurs when users manipulate input to execute terminal commands. The Zora command injection vulnerability stemmed from a line in a Perl file using the *system* function in conjunction with user input saved into the *\$username* variable:

```
system "cat ./${username}";
```

Students executed desired commands with the privileges of the Apache2 web server process by inputting a semicolon terminator followed by the command of choice into the username field on a login web page. For example, if students inputted a semicolon followed by the *print working directory* command (*pwd*), the subsequent screen listed the command output. The output in this case was the current working directory (*/home/zora*).

Students implemented multiple strategies to fix the vulnerability. Some students filtered the input saved into the *\$username* variable to remove characters typically not found in usernames such as semicolons and slashes. Other students eliminated the *\$username* variable from the *system* command while adding code to preserve the original functionality of the web server.

### 2.3.3 SQL Injection

SQL injection vulnerabilities occur when attackers craft input data to cause SQL statements to be executed in ways unanticipated by the original programmer. A common method

of testing for the vulnerability involves inputting a value that causes *WHERE* statements to evaluate to true.

In the Madam Zora web application, a vulnerable PHP script included the following statement:

```
$sql = "SELECT fortune FROM fortune_table
        WHERE spirit_name='{$unfiltered['name']}';"
```

If a user inputted a value causing the overall statement to always evaluate to true, such as:

```
' or 'x'='x'
```

the executed command would be:

```
$sql = "SELECT fortune FROM fortune_table
        WHERE spirit_name='' or 'x'='x';"
```

Since 'x'='x' is always true, the SQL server displays all fortune records in *fortune\_table*. Students filtered the input with the built-in PHP *mysql\_real\_escape\_string* function, which strips out special characters such as quotes. The fixed Zora PHP script included the following code:

```
$filtered['name']=mysql_real_escape_string
($unfiltered['name']);
```

Students eliminated the vulnerability by inserting the filtered value *\$filtered['name']* in place of the unfiltered value *\$unfiltered['name']* in the *SELECT* statement.

### 2.3.4 File Upload

The file upload vulnerability occurs when user input influences the creation, naming, and content of files. In the exercise, students used a web form containing multiple fields to upload malicious code and save it to a file name of their choice. A Zora Perl file contained the vulnerable code:

```
open (FILE, "> ./{$zora}") or $er = 1;
if ($er == 0) { print(FILE "{$fortune}"); }
```

where *\$zora* and *\$fortune* are set by user input fields. The Perl file saved the content of the *\$fortune* variable as a file with the name of *\$zora*. Due to the large amount of text users could save into the *\$fortune* variable, it was possible to upload the source code for a small backdoor to the Zora server. Although the web server saved the file without execute permission, students leveraged the command injection flaw (discussed in section 2.3.2) to issue commands to change permissions on, compile, and run the uploaded backdoor.

## 2.4 System-Level Hardening Description

During the System-Level Hardening exercise, students identified and patched 22 system-level vulnerabilities exhibited by the Madam Zora VMware image. The Zora

image distributed to students ran standard File Transfer Protocol (FTP), Telnet, Hypertext Transfer Protocol (HTTP), Internet Printing Protocol (IPP), and MySQL servers and non-standard persistent Netcat backdoors on seven ports in the 4000-6000 range. Students possessed a list of operational requirements to provide while patching the system: 1) Apache2 server running on standard port 80, 2) MySQL database server running on standard port 3306, 3) Netcat and Cron installed and operational, 4) Working sudo account named *zora*, 5) Secure mechanism to command and control the system, and 6) Secure mechanism to copy files to and from the system.

Students received a blank checklist for the 22 vulnerabilities. Vulnerabilities included persistent backdoors, file permission and account misconfigurations, and the use of outmoded, unencrypted services such as Telnet. As students identified and patched vulnerabilities, the instructor checked the vulnerabilities off of each student list. At the conclusion of the exercise, the instructor discussed all 22 vulnerabilities with the students.

## 3 Reverse Engineering

### 3.1 Background

Reverse engineering involves deciphering the construction and function of a system by studying its observable structure, characteristics, and behavior. Performing reverse engineering requires familiarity with computer architecture, operating systems, programming in assembly and high-level languages, and tools such as disassemblers, virtual environments, network monitors, and system monitors.

There exist two major categories of reverse engineering activities, dynamic analysis and static analysis. Dynamic analysis entails performing a behavior-based study of an executing binary. Dynamic analysis activities include executing the analyzed binary in a controlled environment, viewing file, process, and registry key modifications, and detecting opened ports and established network connections. Static analysis involves studying a binary code listing or artifact without executing the binary. Static analysis activities consist of dissecting a written code representation of a binary by identifying functions, parameters, and arguments and tracing its control flow.

Studying reverse engineering in nine hours by students of varying backgrounds posed a significant risk of incurring student alienation or despair. Although all students had experience programming in a high level language, many had limited exposure to assembly language and computer architecture. Conducting an introductory lecture and smaller supporting exercises before the main exercises reduced this risk. Managing student expectations regarding mastery of the material and providing numerous hints, code comments, and intensive instructor assistance also helped.

### 3.2 Exercise Goals

The instructor stressed to students the impossibility of becoming expert reverse engineers in nine hours. Instead, the instructor presented the main goals as: 1) Reinforcing foundational computer science and engineering topics such as computer architecture, programming, networking, and operating systems, 2) Exposing students to the range and complexity of reverse engineering activities, and 3) Inspiring further study in the field of reverse engineering.

### 3.3 Reverse Engineering Preparatory Lecture

The three-hour preparatory lecture covered reverse engineering concepts such as static and dynamic analysis of code, the computer stack, registers, core x86 assembly language instructions, malicious code, packers, and a set of reverse engineering tools. The instructor provided students with a Windows VMware image containing tools and sample code for analysis. After introduction to each reverse engineering tool, students accomplished a small, but meaningful task using the tool. For instance, after learning about hex editors, students used the XVI32 hex editor program to view the strings embedded in a binary.

After the lecture, students completed dynamic and static analysis activities. The activities consisted of thirty-minute warm-ups intended to prepare students for the three-hour dynamic and static analysis exercises. For the dynamic analysis activity, students applied tools such as hex editors, packers, file system monitors, process monitors, and network monitors to analyze the behavior of a packed binary that output a string to the terminal when executed. The static analysis activity involved determining the purpose, parameters, and return value of an assembly language function. At the conclusion of each activity, the instructor reviewed the solution with the students.

### 3.4 Dynamic Exercise Description

During the three-hour Dynamic Reverse Engineering Exercise, students analyzed three malicious code binaries. Establishing selection criteria for the malware to be analyzed proved crucial for exercise success. The instructor downloaded malicious code from *offensivecomputing.net*, which provided a name, classification, hash values, and a brief description of available malware samples. The instructor obtained and tested over thirty samples before making the final selections.

Selection criteria included the following: 1) Malware must consist of three samples clearly representing different malware categories (e.g. worm, bot, rootkit), 2) Malware must be robust and reliable, 3) Due to time constraints, malware must be unpackable without applying complex techniques such as dumping the original code from memory with a debugger, 4) Malware must exhibit overt, complex, and varied behavior, such as file system changes, persistence

or defensive actions, data collection, and propagation, and 5) Malware must immediately demonstrate effects.

After applying the selection criteria, the instructor chose three samples including a keylogger, a trojan, and a bot. All samples exhibited the desired levels of robustness and reliability, used no packing technology or proved easy to unpack, and demonstrated immediate and varied effects upon installation.

Because the exercise required executing malware, students performed the following tasks: 1) Divided laptops into research machines and analysis machines, 2) Copied the VMware image containing the malware and reverse engineering tools to the analysis machines, 3) Connected the analysis machines to the provided isolated switches (not connected to the Internet), and 4) Connected the research machines to the network providing Internet connectivity. As all student laptops were identically configured and patched and the malware either didn't spread over networks or took advantage of well-known, patched vulnerabilities, spreading from the VMware images to student analysis machines did not present a concern.

After students setup their analysis and research machines and started the VMware image containing the malware samples and analysis tools, they completed an analysis worksheet for each of the three malware samples. The worksheet included fields such as functions and libraries referenced in the binary, files and registry key changes, command and control method, malware defenses, and remediation recommendation.

### 3.5 Static Exercise Description

During the three-hour static analysis exercise, students studied a four-page assembly language listing and packet capture of the Structured Query Language (SQL) Slammer worm. Slammer, also known as Sapphire, is a memory-resident Internet worm that uses a buffer overflow exploit to take control of hosts running a vulnerable version of SQL Server. Slammer propagated by using its victim hosts to generate and send single User Datagram Protocol (UDP) packets containing attack code to random Internet Protocol (IP) addresses. The instructor selected Slammer for the static analysis exercise due to its short assembly code listing and relative simplicity.

To account for the varying backgrounds and technical skill-sets of the students, the instructor split the code into logical, manageable segments, and provided students with research questions for each segment. After students analyzed each code segment and answered the associated questions, the class reconvened to discuss the questions and the instructor checked for understanding before proceeding to the next segment.

Due to time constraints, the instructor provided code comments for instructions requiring tracking values on the

stack or deciphering with a debugger. In addition, the instructor provided the identity and value of the stack entries referenced by the code. For example, the instructor provided the comment *Push address of sock\_addr structure* for the instruction *push eax*. The instructor held a final discussion for lessons learned at the end of the exercise.

## 4 Detect and Defeat

### 4.1 Background

The Detect and Defeat exercise introduced students to technologies that detect cyber threats within a network or on a host computer, how to defeat those threats, and how to architect defense-in-depth systems. Covered technologies included firewalls, network intrusion detection systems (NIDS), and host intrusion detection systems (HIDS).

A firewall is a device that filters network traffic at one or more of the seven layers of the Open Systems Interconnection (OSI) networking stack. There are many different types of firewalls including network packet filters, application proxies, and host system firewalls. At each operating level a firewall will permit or deny traffic based on a set of rules or policy.

Intrusion detection systems come in two different varieties, NIDS and HIDS. The common NIDS is a passive device that examines the ingress and egress traffic of a network and flags suspicious or malicious traffic based on a set of signature rules. A HIDS monitors the integrity of important files, binaries and executables on a host machine and alerts on suspicious or malicious actions. Regarding detection techniques, this exercise focused on signature-based IDSs and did not address anomaly-based IDSs.

### 4.2 Exercise Goals

The goals of the Detect and Defeat exercise were to teach students common network security practices, introduce them to popular network defense tools, and to use these tools to identify and defeat threats within a network. The exercise used tools for the Linux operating system to provide students with additional Linux experience.

### 4.3 Description

The Detect and Defeat exercise introduced students to firewalls, NIDS, and HIDS for the Linux operating system. For practice in configuring firewalls and network intrusion detection systems, the Linux operating system provides an optimal environment for students to learn how operating systems execute firewall rules and process intrusion detection signatures. The two-part exercise consisted of the following: 1) Introductory tutorials for the Linux application iptables, the Snort IDS, and the host Advanced Intrusion Detection Environment (AIDE), 2) Hands-on practice using the tools in a live environment. In part one of the exercise, the instructor asked students to implement the requirements outlined in each tutorial on a practice Linux virtual machine (VM). For

part two, the instructor gave students two VMs to use in a live scenario. The two VMs consisted of an aggressor VM and a defender VM. The defender VM had iptables, Snort, and AIDE installed but not configured. The aggressor VM periodically executed a set of scripts that sent data to open ports on the defender VM, triggered Snort IDS signatures, and created alerts within AIDE. The objective for students was to correctly configure the defender VM to detect and defeat the aggressor VM actions. The students demonstrated to the instructor the following items: 1) Used iptables to filter incoming traffic and allowed traffic on port 80 (HTTP), 443 Secure Sockets Layer (SSL), and 22 Secure Shell (SSH), 2) Configured Snort to recognize and alert against the traffic coming from the aggressor VM, and 3) Established an AIDE hash database for the files in the /etc directory and demonstrated an alert on a file in the /etc directory.

## 5 Digital Forensics

### 5.1 Background

Computer forensics is the discipline that combines elements of law and computer science to collect and analyze data from computer systems, networks, wireless communications, and storage devices in a way that is admissible as evidence in a court of law [1]. The traditional forensics methodology consists of acquiring evidence without altering the original media, analyzing the data to produce the necessary evidence, and proving the authenticity of the evidence.

Ten years ago computer forensic practices mainly entailed examining computer hard drives after a crime took place. Recently, live response forensics, network forensics, and rapid evidence gathering of data have been included under the computer forensics field of study. In digital forensics there are two basic data types, persistent data and volatile data. Volatile data is the transient data found on a digital system that exists while the computer is powered on. Persistent data is the information stored on a hard drive. This exercise concentrated primarily on the technical aspects of performing digital forensics and did not include in-depth coverage of the legal aspects of forensics.

### 5.2 Exercise Goals

The goal of the Digital Forensics exercise was to enable students to analyze digital media using established digital forensic techniques. The goal included having students analyze a piece of digital media and gather digital evidence using established methods that will stand up in a court of law.

### 5.3 Description

The digital forensics exercise focused on the following scenario:

Agent Johnson of the Office of Special Investigations (OSI) is in the middle of an investigation against several groups

connected to the mafia. Agent Johnson just arrived to a murder scene of someone connected to the mafia group. Agent Johnson is in charge of digital evidence collection and has found a running desktop at the scene. He must: 1) Perform a live forensic investigation, 2) Conduct a post-mortem investigation, and 3) Build a report and present the findings to the instructor. Recovered evidence must include an IM conversation between mafia members, photos of a weapons exchange, and the passwords to a user account with important information.

The two-part exercise included a live response investigation and a traditional forensic analysis of the physical hard drive. The exercise setup consisted of a Windows XP SP1 virtual machine representing the “running desktop” and a USB external hard drive representing the physical hard drive of the desktop. Students used a combination of the following forensics tools to accomplish exercise objectives: 1) Helix Forensics Live Linux CD, 2) WinHex, 3) md5sum, 4) Linux ‘DD’ command, and 5) FTK Imager.

Throughout the exercise the instructor checked to ensure students performed their forensic analysis using sound techniques. In a forensic investigation the integrity of the original evidence is crucial. The instructor required students to preserve the chain of command while interacting with the digital evidence. This included performing the following steps to accomplish the exercise: 1) Used statically linked tools from the Helix Linux CD to record all the running processes, applications, logged in users and all pertinent transient data of the running desktop, 2) Calculated an md5sum hash of the physical hard drive in a read-only manner, 3) Created a forensic image of the physical hard drive, 4) Took an md5sum hash of the forensic image and compared it to the original media, 5) Used a hex editor like WinHex or forensic analysis tool to examine the forensic image, and 6) Thoroughly documented the process and all the evidence findings. At the conclusion of the exercise, each team presented their findings to the instructor.

## 6 Wireless Access Point Treasure Hunt

### 6.1 Background

The capstone DefEX was the Wireless Access Point (WAP) Treasure Hunt. Students engaged in wardriving to locate a series of time-constrained challenges distributed across a small city. Wardriving entails utilizing a vehicle and a portable computing device to search a geographic area for a wireless network. Students used network detector software such as NetStumbler and Kismet to locate WAPs of interest. The Treasure Hunt challenges located at the WAPs included completing a cryptography problem and circuit worksheet, discovering a password vulnerability, conducting a forensics analysis on a thumb drive, bypassing authentication on a website, and identifying file, database, and mail server misconfigurations.

### 6.2 Exercise Goals

The overarching objective of the WAP Treasure Hunt was to test the leadership and decision making skills of the students in a time-critical environment. The Treasure Hunt provided a cumulative team-based challenge for students that reinforced select topics from other exercises.

### 6.3 Description

In the WAP Treasure Hunt exercise, teams searched for a series of WAPs temporarily dispersed by instructors around a city. At each WAP site, students completed a challenge to receive the map leading to the next WAP site. Students earned points for completing challenges within a given time deadline. If the deadline passed, instructors gave students the next map but did not award any points. The team accumulating the most points won the exercise. The exercise culminated with a final challenge at a bowling alley where students found the “treasure”, a package of silver and gold chocolate candies and a surprise pizza and bowling party.

The key organizational activities included constructing the order of site visits, creating the map scrolls, and setting rules of engagement for the exercise.

1) Construct Challenge Site Visit Order: The instructor generated the order of the challenge sites each team visited. Although a unique path for each team was desirable to minimize challenge site congestion, due to the number of teams and sites some redundancy was unavoidable.

2) Create Map Scrolls: For each site, the instructor created a satellite map of the pertinent city area, delineated a reasonably sized search area incorporating the challenge site with a red box, and added red text indicating the SSID for the WAP at the site. The instructor rolled the maps and tied them with different colored ribbons to create scrolls. Each team had a unique ribbon color, which helped site attendants provide teams with the correct scrolls.

3) Set Rules of Engagement: For safety reasons, the instructor assigned each team a driver, who was not allowed to assist students with locating the WAPs or completing the site challenges. Instructors prohibited students from connecting to any WAPs other than the ones involved in the exercise and required them to gain permission from their driver before connecting to any WAP. Instructors also disallowed using any equipment other than student laptops, such as external antennae.

The challenges included completing a cryptography problem and circuit worksheet, examining WAP password authentication, conducting a forensics analysis on a thumb drive, investigating website authentication, and identifying file, database, and mail server misconfigurations.

1) Initial Challenge: Student teams gathered at an initial site for an exercise briefing, to review the rules of engagement, and to complete the first challenge, a cryptography problem and a circuit diagram worksheet. After reviewing the exercise purpose and rules, each team received six map scrolls tied with six different colored ribbons. The cryptography and circuit diagram challenge answer resulted in one of the six colors. If students solved the problem correctly and chose the right scroll, they proceeded to the next challenge. If the team calculated an incorrect color and chose a wrong scroll, the map led to a time penalty site where an instructor required students to answer a set of cyber operations questions before giving them the map to the next challenge.

2) WAP Configuration: When conducting vulnerability assessments, students often overlook supporting networking infrastructure such as switches, routers, and wireless equipment. The WAP configuration challenge tested this common oversight while distracting students with a decoy Nepenthes honeypot. In this challenge, the password-protected WAP web interface had an easily guessed username and password. The main WAP configuration screen listed a passphrase that students could exchange for the next map.

3) Forensics: Each student team analyzed a USB thumb drive using sound digital forensic techniques. Instructors set up the challenge site as an “investigation scene”, placing a thumb drive containing the evidence and a decoy laptop workstation at the scene. Instructors formatted the thumb drive with a persistent version of Ubuntu Linux that used an ext3 file system and embedded the passphrase for the next map in the slack space of the file system. To find the passphrase, each team created a forensic image of the thumb drive and examined the image in a hex editor or a piece of forensic analysis software.

4) Website Authentication Challenge: The Website Authentication Challenge required students to complete a six-level password challenge. Inputting any values for the username and password on the first level advanced students to the second level. The passwords for levels two through four resided in a comment, an image tag, and a hidden form field in the html source code respectively. Inputting any values into the username and password fields on the fifth level provided a list of nine file names from file1.txt through file9.txt. Inputting each file name into a field titled *Magic Phrase* gave students Morse code representations for letters of the password and a Morse code key. The sixth level gave the hint *l6pass.txt*. Students could obtain the password by inputting `http://<ip address>/l6pass.txt` into the browser window. Completing the sixth level gave students a passphrase to exchange for the next map.

5) FTP, MySQL, and Simple Mail Transfer Protocol (SMTP) Configuration: Students explored multiple machines running FTP, MySQL, and SMTP servers to derive a passphrase and earned the next map. Students retrieved a MySQL username and password from an FTP server with anonymous login

enabled. Students used the username and password to log into a MySQL server and retrieve the first part of the map passphrase from a table. The instructor hid the second part of the map passphrase in the banner of a SMTP server. Students created a Transport Control Protocol (TCP) connection with the SMTP server to retrieve the banner.

## 7 Conclusion

Successful completion of DefEX required Computer Science, Computer Engineering, Electrical Engineering, Physics, and Math undergraduates to exercise a broad range of skills. Technical skills included interpreting Assembly language, analyzing and patching Perl and PHP scripts, finding and eliminating persistent backdoors, configuring intrusion detection systems and firewalls to repel known attacks, and conducting live and post-mortem digital forensics analyses after an attack. Exercises also required leadership, teamwork, executing under pressure, and problem-solving skills.

Providing a diverse group of students with a common set of foundational knowledge and finding the balance between enabling participation of novice students and generating problems complex enough to challenge experienced students posed the major curriculum design risks. Instructors reduced the risks by administering a technical survey, requiring students to complete a set of fundamental exercises, and assigning balanced student teams. As a result, student teams successfully completed all of the exercises.

## 8 Acknowledgements

The authors wish to thank Dr. Kamal Jabbour, ST, Air Force Senior Scientist for Information Assurance, Regina Recco, Dr. Sarah Muccio, Lt Col (ret) Ken Chaisson, and Thomas Vestal for their diligent support during the development and execution of these exercises.

## 9 References

- [1] US-CERT, “Computer Forensics,” [Online]. Available: [http://www.us-cert.gov/reading\\_room/forensics.pdf](http://www.us-cert.gov/reading_room/forensics.pdf).

## A Plan for Training Global Leaders in Cybersecurity

A. Bobkowska<sup>1</sup>, L. Kuźniarz<sup>2</sup>, G. Markowsky<sup>3</sup>, A. Ruciński<sup>4</sup> and B. Wiszniewski<sup>5</sup>

<sup>1</sup>Department of Software Engineering Gdańsk University of Technology, Gdańsk, Poland

<sup>2</sup>School of Computing, Blekinge Institute of Technology, Karlskrona, Sweden

<sup>3</sup>Department of Computer Science, University of Maine, Orono, ME, USA

<sup>4</sup>Department of Electrical and Computer Engineering, University of New Hampshire, Durham, NH, USA

<sup>5</sup>Department of Knowledge Engineering Gdańsk University of Technology, Gdańsk, Poland

*Abstract - Cybersecurity is a challenge that requires global cooperation and global education. This paper will present a vision for a global university that will educate potential leaders in global cybersecurity. It will also provide a roadmap for realizing this vision using existing assets. Global leaders in cybersecurity should have not only excellent technical skills, but also a global mindset that allows for conducting efficient and innovative activities globally. Such leaders should understand cultural differences, should have high ethical and professional standard, have excellent interpersonal and management skills, understand administrative and legal constraints, and be able to work with both academia and industry. We will describe in detail how to train such leaders.*

**Keywords:** global, cybersecurity, education, curriculum, graduate, masters

## 1 Introduction

We believe that the best way to train global leaders is globally. By this we mean that these leaders need to be educated in more than one place and that they need to be educated in multiple places so they better understand the environments in which they expect to work. In particular, we believe that these leaders need to earn degrees that reflect the values of all of the environment in which they will work.

We will focus on the case of two environments because it is the simplest and can provide a basis for more complicated environments. Assuming that we are talking about functioning in two different environments, we believe the optimal approach is to think in terms of a dual degree program. A dual degree program would give students a real taste of the different perspectives of the environments in which they will work

At a minimum, we would expect the degree to be at least a two year degree with at least a year spent in each environment. This is enough time for a student to absorb elements of a new culture and begin to understand the alternative perspectives that a new culture presents. We strongly feel that cultural learning is an important part of the program because dealing with humans is the most challenging part of cybersecurity.

## 2 Objectives

There are a number of important objectives for our proposed program. One objective is to promote mutual understanding between the people of the two environments. This includes providing broader knowledge of the languages, cultures and institutions of the two environments. Another objective is to improve the skill of practitioners seeking to work globally and to enhance their satisfaction and to stimulate their dedication to their profession.

Another objective is to enhance collaboration between higher education and vocational training institutions in the two environments with a view of promoting joint study programs and mobility. Yet another objective is to support the development of global standards and expectations in the area of cybersecurity. We also expect that the graduates of such a program will maintain a high level of interest in the political and social developments of their environments in addition to staying technically current.

Additional objectives include graduating students with excellent employment opportunities. We also expect these students to have new perspectives on cybersecurity that would lead them to spot new business opportunities that would be less likely to occur to people more narrowly trained. We also want the program to eventually be self-sustaining.

Another objective for the program is to get as much integration of admission standards and courses as possible. We expect that the exercise of integrating such things would be valuable to all concerned parties.

## 3 Outcomes

Our goal would be to assess the program using an ABET style assessment mechanism. In addition to the academic and technical outcomes that one would expect in such a program, we also plan to include outcomes addressing the language competence of all graduates. The plan is for all instruction to be in English, so English competence would need to be demonstrated by all graduates. For students whose native language is English, there would be a requirement to demonstrate competence in a different language appropriate to the program. Additional language competence requirements would be considered based on specific requirements of particular programs.

The outcomes for the proposed program would need to be compatible with the outcomes of the degree programs that would be incorporated into the dual degree program.

More generally, we would have assessment carried out by independent entities. We intend to work closely with international organizations such as IFEEES, SEFI, ACM and the IEEE to develop the outcomes and assessment plan.

## 4 Program Structure

An important factor in the success of our program is its administrative structure. We are in the process of working these details out. In particular, we expect the final structure to be based on the structures in place at the cooperating institutions influenced by structures used by such programs as the ERASMUS (European Community Action Scheme for the Mobility of University Students) [1].

## 5 Challenges

We fully expect there to be many challenges. In particular, different degree requirements at different institutions need to be satisfied. Admission standards need to be harmonized. There are issues of graduate assistantships that need to be resolved. Also various tuition rates will need to be harmonized and other financial details will need to be worked out. There could also be visa and other political issues that need to be addressed depending on the environments chosen. We also expect that there will be issues that will need to be addressed that we do not realize are issues at this time. As noted in the previous section, we expect that determining the final governing structure for the program will also be a challenge. For this reason, we feel that the next step is a pilot project that we will describe in the next section.

## 6 Implementation

Although we intend our discussion to describe education that will work in a variety of environments, we will plan to run a pilot project using Europe and the US because those are the environments with which we are most familiar. We expect other environments to be more challenging for us and we can better deal with them once we have a successful example to work from.

We propose piloting this program with an initial group of 24 students. The proposed program would be a two year degree program. All students would spend one year at two universities in Europe and one year at one or more US universities. Students will be awarded one European degree, either a Master's in a Specialisation of Informatics from Gdańsk University of Technology, Poland, or a Master's in Software Engineering or Master's in Computer Science from Blekinge Institute of Technology, Sweden) and one American

degree (Master's in Computer Engineering from the University of New Hampshire or a Master's in Computer Science from the University of Maine). Details will depend on the particular course plan. The initial class should contain 24 students, 12 from Europe and 12 from the US. We also expect to have a faculty exchange to build a common ground for instruction. We plan to develop this program in more detail and apply for funding from the Atlantis program [2]. We feel that the goals of the Atlantis program fit very well with the goals of our proposed program.

## 7 References

- [1] Website for the ERASMUS program  
[http://ec.europa.eu/education/lifelong-learning-programme/doc80\\_en.htm](http://ec.europa.eu/education/lifelong-learning-programme/doc80_en.htm)
- [2] Website for the Atlantis program  
[http://ec.europa.eu/education/eu-usa/doc1156\\_en.htm](http://ec.europa.eu/education/eu-usa/doc1156_en.htm)

# Goals, Models, and Progress towards Establishing a Virtual Information Security Laboratory in Maine

C. Cavanagh<sup>1</sup> and R. Albert<sup>2</sup>

<sup>1</sup>University of Maine at Fort Kent, Fort Kent, ME, USA

<sup>2</sup>Professional Management Division, University of Maine at Fort Kent, Fort Kent, ME, USA

**Abstract** - *Information security education remains a critical topic in today's information driven societies. Educational institutions have been called to action to help raise information security awareness, knowledge and skills in those they serve. Cyber defense competitions are an attractive option to help raise awareness and interest in information security. Effective information security educational activities and cyber defense competitions most often require significant technical resource requirements including an information security laboratory infrastructure. Universities and other organizations have made progress to date in defining and demonstrating practical information security laboratory infrastructures. The purpose of this paper is to identify the goals associated with establishing an information security laboratory that will support information security education and outreach efforts within Maine, identify models that have been successfully demonstrated to date, and report on progress made in the design and implementation of this laboratory in Maine.*

**Keywords:** Cybersecurity competition, information security education, security lab, virtual machine

## 1 Introduction

Information security remains a critical topic in today's information driven societies. Driving much of this increased attention is the increased severity of damage that has resulted from failed efforts to secure information systems, the prolonged dearth of information security practitioners, and the low level of information security awareness in our general population.

Educational institutions have been called to action to help raise information security awareness [1]. In 2009, President Obama ordered a 60-day, comprehensive, *clean-slate* review to assess U.S. policies and structures for cybersecurity. Specific recommendations contained in the resulting report included a call to initiate a K-12 cybersecurity education program for digital safety, ethics, and security as well as expanded university curricula [12].

Universities are ideally positioned to orchestrate such competitions for the betterment of current and future students and to contribute to the best preparation of future information workers and leaders.

Cyber defense competitions are an attractive and effective option for raising awareness and interest in information security while simultaneously educating for prevention and addressing the private and government sector needs, but they require significant technical resource requirements [2, 5]. Chief among these is an information security laboratory that can be used to help prepare participants and avail them an environment in which to compete.

The intended audience and instructional delivery modalities that must be supported are two considerations that must be made when establishing an information security lab in support of higher education initiatives, including outreach efforts. Our context requires provisioning for secure access to lab facilities by students enrolled in our distance education programs and potentially by those participating in the Maine Cyber Defense Competition (MECDC).

The educational potential of virtual computer labs, including those in support of information security instruction in different contexts has been reported by many researchers [4, 13, 14, 18, 19]. Such experiences have contributed greatly to the evolution of goals and models used to inform subsequent designs.

A virtual information security laboratory does not yet exist in Maine and this is believed to be the first effort of its kind in the state.

## 2 Common Information Security Lab Goals

The goals for the Maine virtual information security lab were established following a review of the goals, outcomes, benefits, and recommendations stemming from similar efforts to date. For example, goals were defined to ensure realization of the instructional advantages identified as being associated with delivering a rich learning experience made possible through virtualization.

Desirable characteristics of a virtual lab include accessibility, observability of host and network events, ability to simulate realistic scenarios and devices, separability of virtual networks, remote configurability, and the ability to share resources efficiently [14]. In this context, a virtual lab is defined as a facility that provides a remotely accessible

environment to conduct hands-on experimental work and research in information security. An additional primary characteristic should be the ability to isolate the virtual lab systems from the campus network [11].

Virtualization has been reported to provide benefits including giving each student control over configuration, providing unique IP addresses to each server, and the ability to demonstrate centralized logging [16]. Additional benefits include the provisioning of an appropriate platform in support of different areas of IA instruction, support for rapid prototyping of computer and network configuration, increased availability of lab environment, providing a uniform experience across students, and simplified and cost effective course administration [13]. All of the benefits can be viewed as supporting constructivist approaches to instruction.

The instructional advantages of a constructivist approach to instruction are well established [3, 9, 15]. Problem solving in authentic environments is an example of an effective constructivist learning technique [19]. Using virtualization to support lab activities that reinforce problem solving skills in authentic environments should be considered an essential component of a well designed virtual information security lab.

Virtualization, when coupled with online synchronous instruction software (e.g., Elluminate, GoToMeeting), has been reported to provide opportunities for dealing with heterogeneous groups, facilitating instructional design possibilities for a diverse audience, enhancing student-teacher interactions, and assisting students in acquiring complicated technical skills with greater ease and more interest [13]. Inclusion of online synchronous instruction software should therefore be considered an essential component of a well designed virtual information security lab that supports distance education.

For students to fully comprehend and benefit from lab activities, it is also essential to ensure they first have the necessary background knowledge [17]. Well crafted lab activities that are remotely accessible can well serve this need. "Web labs" in particular, have been constructed at a high level of abstraction and used to introduce, demonstrate, reinforce, and encourage experimentation with complex security concepts. Lab activities such as these should also be considered an essential component of a well designed virtual information security lab.

Efforts such as the University of Maine Cybersecurity Education Guide [10] to collect, categorize, and improve the searchability and accessibility of all forms of information security educational materials are greatly valued for their potential to aid a much larger audience of educators. The addition of appropriate search metatags and corresponding data to form true digital libraries of such resources represents a significant contribution. Access to the fruits of these efforts

should be considered an essential component of a well designed virtual information security lab.

### 3 Maine Virtual Information Security Lab Design Goals

The goals established for the virtual Maine Information Security Lab (MEISLab) are based predominantly on the need to establish an information security laboratory in support of information security outreach and educational efforts within Maine. They are also based on the aforementioned goals, outcomes, benefits, and recommendations. Finally, the goals are based in part on the design considerations for constructing a virtual computer lab environment appropriate to small campus environments. These design considerations include particular sensitivity to the often very limited financial, space, machine, time, and staffing resources available at many schools [7].

The aim is to design an instructional laboratory environment that:

- Harnesses the instructional benefits of virtualization;
- Provides remote access in support of online education modalities including the option for synchronous instruction; and
- Supports learning activities that:
  - Ensure students have sufficient background knowledge to maximize comprehension
  - Promote students' appreciation of the ethical dimensions associated with engaging in information security activities
  - Promote students' compliance with all information security and technology use policies
  - Ensure students meet the student learning outcomes and related curricular requirements defined for the information security degree programs.

### 4 Virtual Information Security Lab Models

There are several predominant virtualization solution models including, Apache Virtual Computing Lab (AVCL), Microsoft HyperV (MHV), VMLogix LabManager, and Xen. Each model exhibits its own benefits and challenges.

For example, benefits reported being associated with an information security lab model combining VMware Lab Manager, Virtual Center, and ESX include:

- increased accessibility for students to laboratory resources;
- fewer hardware components resulting in decreased administrative overhead; and
- support for complex laboratory exercises [4].

Challenges, as reported for this same example, include:

- Training requirements to prepare instructors and students to correctly navigate and complete the process of selecting and deploying a virtual machine;
- Limited cataloging and organizing features for virtual machine images, thereby making searching and selection more challenging;
- Limited browser support; and
- Significant demands on underlying server resources (especially disk space) and their management [4].

Similar reports are available that provide an account of the benefits and challenges of each of the other models. In the end, a model should be selected based on the particular needs and resources of the organization.

## 5 Progress

This report is as much an outline of the process used to identify the design goals of the MEISLab as it is an account of progress toward implementation and experience gained to date. Nevertheless, several final product comparisons have been made and this has led to the selection and installation of several key hardware and software components.

As this is a small scale pilot operation, the server that has been selected is a Dell PowerEdge R610 configured with Dual Xeon E5620 2.4 GHz processors each with 12MB cache, 24GB RAM, 146GB RAID configured storage spanning 6 physical disks and a quad port Gigabit Ethernet NIC. The virtualization solution that was selected is a “bare metal” VMware vSphere Hypervisor (ESXi) and VMware vCenter Lab Manager 4.0. As of this writing, the Lab Manager product is being deprecated and replaced by VMware vCloud Director that will require evaluation in the year ahead.

The department has also registered with the VMware Academic Program that provides free or reduced cost access to many VMware products in non-production, instructional settings. The selection of VMware was made in part due to it being the closest to being an “out-of-box” solution [4]. The combination of low/no cost and feature maturity of the commercial product line made it the best candidate for this particular application. Having access to VMware's products

will avail students the opportunity to quickly access and deploy “stock” course-oriented virtual machine images enabling them to take the assignments further with additional exploration and experimentation. The Lab Manager product will be used to control access by both local and distance situated students.

The Citrix GoToMeeting service was purchased for use in synchronous instruction as needed to support distance education students. The service provides for impromptu or scheduled presentations and live demos to individual students or groups that can be used to fill instructional gaps. It can also be used to provide the instructor the ability to remotely observe and control a student's computer. This can be very useful when a simple demonstration or technique correction within the student's actual computer is called for. Training of key personnel has also been completed.

Progress has begun on the establishing a collection of useful VM images in support of laboratory exercises. This effort is expected to continue and include the preparation of images in support of the cyber defense competition.

## 6 Conclusion

The need for better securing the information that today's societies increasingly dependent upon is expected to continue well into the foreseeable future and so to the need for more and better information security education. Educational institutions have been called to action to help raise information security awareness, knowledge and skills in those they serve and cyber defense competitions are but one attractive option.

Significant technical resource requirements including an information security laboratory infrastructure must be met to fully support information security education and outreach initiatives. Providing remote availability of these resources to a distant population is considered essential in situations in which distance education is involved.

There are many design considerations that should be taken into account and it is essential to identify the specific goals of a virtual information security Lab and select the most appropriate model that will support attainment of these. In our context, the infrastructure model that has been selected consists of a combination of VMware products.

Our progress to date in implementing this infrastructure includes the purchase of the necessary services, software and prerequisite hardware components. Efforts have begun to build a library of “stock” course-oriented virtual machine images in support of specific instructional lab activities. Additional effort has been made to ensure the proper training necessary for the synchronous instruction support system and to ensure the availability to students of “Web labs” for the purpose of ensuring students will be better able to fully

comprehend and benefit from lab activities by having the prerequisite conceptual knowledge.

## 7 References

- [1] Albert, R. (2009). "The 'U' in Information Security". *Proceedings of the 2009 ASCUE Summer Conference*, pp. 23-31. Retrieved May 1, 2010 from [http://www.eric.ed.gov/ERICDocs/data/ericdocs2sql/content\\_storage\\_01/0000019b/80/45/2f/85.pdf](http://www.eric.ed.gov/ERICDocs/data/ericdocs2sql/content_storage_01/0000019b/80/45/2f/85.pdf)
- [2] Albert, R. & Wallingford, J. (2010). "Cyber Defense Competitions - Educating for Prevention", *Proceedings of the 2010 ASCUE Summer Conference*, pp. 22-30.
- [3] Boghossian, P. (2006). "Behaviorism, Constructivism, and Socratic pedagogy", *Educational Philosophy and Theory*, 38(6), pp. 713-722.
- [4] Burd, S. D., Gaillard, G., Rooney, E., & Seazzu, A. F. (2011). "Virtual Computing Laboratories Using VMware Lab Manager", *Proceedings of the 44th Hawaii International Conference on System Sciences*.
- [5] Conklin, A. (2006). "Cyber Defense Competitions and Information Security Education: An Active Learning Solution for a Capstone Course", *Proceedings of the 39th Hawaii International Conference on System Sciences*.
- [6] Elluminate. Retrieved May 1, 2011 from [www.illuminate.com](http://www.illuminate.com)
- [7] Gephardt, N. & Kuperman, B. A. (2010). "Design of a Virtual Computer Lab Environment for Hands-on Information Security Exercises", *Journal for Computing Sciences in Colleges*, 26(1), 32-39.
- [8] GoToMeeting. Retrieved May 1, 2011 from [www.gotomeeting.com](http://www.gotomeeting.com)
- [9] Kumar, M. (2006). "Constructivist epistemology in action". *Journal of Educational Thought*, 40(3), pp. 247-261.
- [10] Markowsky, G. & Markowsky, L. (2010). "Consumer Guide to Online Cybersecurity Resources: UMCEG". *Proceedings of the 2010 International Conference on Security Management, SAM 2010, July 12-15, 2010, Las Vegas Nevada, USA, 2 Volumes*.
- [11] Nance, K. L., Hay, B., Dodge, R., Wrubel, J., Burd, S. D. & Seazzu, A. F. (2009). "Replicating and Sharing Computer Security Laboratory Environments", *Proceedings of the 42<sup>nd</sup> Hawaii International Conference on Systems Sciences*.
- [12] National Security Council (2009). "60-day Cyberspace Policy Review: Assuring a Trusted and Resilient Information and Communications Infrastructure". Retrieved May 1, 2010 from [http://www.whitehouse.gov/assets/documents/Cyberspace\\_Policy\\_Review\\_final.pdf](http://www.whitehouse.gov/assets/documents/Cyberspace_Policy_Review_final.pdf)
- [13] Nestler, V. & Bose, D. (2011). "Leveraging Advances in Remote Virtualization to Improve Online Instruction of Information Assurance", *Proceedings of the 44<sup>th</sup> Hawaii International Conference on System Sciences*.
- [14] Padman, V. & Memon, N. (2002). "Design of A Virtual Laboratory for Information Assurance Education and Research", *Proceedings of the IEEE Workshop on Information Assurance and Security, June 17-19, 2002, USMA, West Point, New York, USA*.
- [15] Powell, K. C. & Kalina, C. J. (2009). "Cognitive and social constructivism: Developing tools for an effective classroom", *Education*, 130(2), pp. 241-250.
- [16] Powell, V. J. H., Johnson, R. S. & Turcheck, J. C. (2007). "VLABNET: The Integrated Design of Hands-on Learning in Information Security and Networking", *Proceedings of the 2007 Information Security Curriculum Development Conference*, September 28-29, 2007, Kennesaw, Georgia, USA.
- [17] Schweitzer, D. & Boleng, J. (2009). "Designing Web Labs for Teaching Security Concepts", *Journal of Computing Sciences in Colleges*, 25(2), pp. 39-45.
- [18] Stackpole, B., Koppe, J., Haskell, T., Guay, L. & Pan, Y. (2009). "Decentralized virtualization in systems administration education", *Proceedings of the 9<sup>th</sup> ACM SIGITE conference on Information Technology Education*.
- [19] Wu, Yu (2010). "Benefits of Virtualization in Security Lab Design", *ACM Inroads*, 1(4), pp. 38-42.

# RTFn: Enabling Cybersecurity Education through a Mobile Capture the Flag Client

Nicholas Capalbo, Theodore Reed, and Michael Arpaia

Computer Science Department, Stevens Institute of Technology, Hoboken, NJ, USA

**Abstract**—Cybersecurity is one of the most highly researched and studied fields in computer science. It has made its way into numerous accredited universities as a full-fledged degree program. Students are constantly exposed to new technologies and methodologies through coursework. However there is a shortage of places to practice, in a controlled environment, the skills gained in the classroom. For that and numerous other reasons, the industry has seen a noticeable increase in capture-the-flag (CTF) style competitions. In the same vein, entering a CTF-like competition for the first time is a daunting task for any university. To aid in the organization of resources before, during, and after the competition we present the Rock The Flag network (RTFn). RTFn is a combination of hardware and software, which provides VPN capabilities as well as a central repository for tool tracking and real-time competition information. In this paper, we present an in depth discussion of this tool, its capabilities, and how it can aid in the organization required during CTF-style competitions.

**Keywords:** Cybersecurity, Education, CTF, War-Games, Competition Logging, Collaboration

## 1. Introduction

Cybersecurity is an emerging concentration for undergraduate college students, and a developing concentration for masters candidates, doctoral, and post doctoral researchers. Academia has provided the field with a foundation for creativity, research, and innovation. As the field moves into the realm of undergraduate study, academia fails to adequately prepare students for the advanced technical work required at established security organizations [18]. Advanced degrees and programs build upon the experience of their candidates and students. Undergraduates often do not have such experience, thus it appears that academia is not an ideal venue for practical experience.

To compensate for a lack of experience, undergraduates must rely on programs which integrate that experience, such as labs or emulations. Alternatively, they can augment their undergraduate study with extra-curricular activities or internships. Many undergraduate programs recognize this requirement and integrate courses focusing on providing practical experience [21], [16], [3], [19]. These programs may invite experienced industry professionals [1], or allow

students to use emulation and simulation sandboxes [12]. Bringing practical experience into the classroom is difficult.

It becomes even more challenging to provide red-team experience<sup>1</sup> to undergraduates. It is widely understood that this type of activity on commercially owned networks is against the law (as described in the Computer Fraud and Abuse Act of 1986). Although the goal of cybersecurity education and research is to create defensive strategies, playing the role of the attacker is often necessary. Defenses cannot be created effectively if attack methodologies are unknown. Creating attack taxonomies is the first step for assessing risks and developing defenses. This requirement elicits ethical considerations when providing students with the tools and processes for conducting attacks [1]. While these exercises demonstrate a great deal in a test bed environment, trust must be placed in students to not utilize these tools outside of a controlled environment.

Capture the Flag competitions are emerging as the solution to the issues created when introducing cybersecurity as a field for undergraduate study. These competitions [2], [6], [4], [15], [13] are designed to provide the needed cybersecurity experience that compensates current events, emerging trends, and course work. They provide attack and defense scenarios by facing students against difficult tasks, obscure procedures, and each other. Competitions provide feedback in the form of ranking and a detailed synopsis of the events. They introduce the fast-paced environment which surrounds the cybersecurity industry, and has the potential to teach information security related problem solving from experience [8].

In this paper we enhance this potential by providing institutions a quick solution to compete, perform well, collaborate amongst teams, identify weaknesses, and extract valuable experience during each event. We describe a hardware and software solution called the Rock the Flag network (RTFn). In section 2 we review work related to creating capture the flag competitions, and their impact on collegiate study. Section 3 introduces our Rock the Flag network and provides an overview of its hardware and software suite. This section also outlines goals for capture the flag competition participation. In section 4 we examine the minimum set of hardware components for RTFn; section

<sup>1</sup>Experience that involves malicious behavior, such as penetration testing, or attack design.

5 examines the software components. In section 6 we outline our experiences as an institution introduced to capture the flag competitions and the steps we followed to organize a successful team of students. In sections 7 and 8 we provide our conclusions and future work.

## 2. Related Work

Capture the flag and cybersecurity competitions may utilize Virtual Private Networks (VPN) to enable competition play [10], [17], [2]. The VPN connects teams to either a defensive or offensive network where various oracles provide scoring mechanisms. In a defensive competition teams are often provided with a Virtual Machine (VM) which contains various flaws and security holes [18]. Teams may be scored by how well they can patch, secure, and defend their VM against a scoring system or other teams. If teams are required to defend their VM, as well as attack other team's VMs, the event is considered both an offensive and defensive competition. Competitions may also require only offensive challenges; these events typically involve a set of VMs maintained by the scoring system [14].

RTFn is a unique suite of hardware and software that enables collaboration. It resembles software-engineering and collaboration software, without focusing on development. RTFn does not suggest any methods to improve or change cybersecurity competitions. The tool is partly a response to documented "lessons learned" documents, published by various competition administrators, created to assist teams during competitions. RTFn may not be appropriate for all future competitions, and there are current competitions that will not actively utilize RTFn. We maintain that, in such competitions, RTFn will still enhance a team's performance during these competitions.

## 3. Approach

RTFn presents us with a significant amount of competition improvements and advancements. It is comprised of a combination of hardware and software that work with each other to maximize success in many of the areas of competition that often go overlooked. RTFn can be implemented as either a rack-mount solution or a mobile stand-alone system. Both implementations of RTFn have unique advantages and disadvantages. The rack-mount solution offers a static network address for persistent access by student competitors while the mobile stand-alone implementation offers added portability for off-site competitions.

RTFn was designed to solve the following existing problems related to participating in cybersecurity competitions:

- Universities may not have server space to host tools
- Teams may not have a dedicated meeting area to organize
- Teams experience a lack of consistency and coherence between competition events

- It is difficult to extract learning items or recognize weaknesses during competition

RTFn was constructed using the following goals:

- Organize cybersecurity competition participation a-priori and post-priori
- Enable task-scheduling of competition challenges<sup>2</sup>
- Enable campus involvement, with minimum configuration and communication overhead
- Keep competition-related information secure
- Trend competition outcomes based on problem type, time, skills required, etc.

RTFn has very few requirements. There are a minimal amount of hardware requirements and the software used is adapted from open source solutions. Of the hardware requirements, it is necessary to equip RTFn with a large storage media. This is essential for the use of Virtual Machines, a repository of tools and the storage of data-mined information, challenges and reports. Also, it is imperative to outfit RTFn with a fast processor. This assists in many areas which include, but are not limited to, the running of virtual machines, the acceleration of key generation, running tools that support multi-threaded execution, and off-loading VPN requirements. It is important to keep the requirements to implement RTFn low to improve and promote university involvement in CTF-style competitions in an easy, fun fashion.

RTFn provides the following features: a challenge ownership portal, file uploading, and a real-time collaborative document editing. The challenge ownership portal assists in many areas such as work-load distribution, task completion, and coordination. Competitors can flag themselves as the owner of a specific challenge while they are working to minimize repetition. Marking ownership of a challenge is a highly effective way to accomplish multi-location participation. By marking their progress on a competition, team members will know to work on other challenges and propagate successful coordination and completion of challenges and tasks.

The RTFn also serves as a database of information security tools and scripts. Taking advantage of automated tools is an essential part of efficiently participating in CTF-style competitions. RTFn presents a structured way to categorize and cross-reference specific tools with specific types of challenges.

## 4. RTFn: Hardware Components

In this section we evaluate possible mobile hardware solutions, and provide a recommended configuration. We considered two deployment options for RTFn; one as a rack-mounted server in our university's information technology department's server room, and the second as a stand-alone

<sup>2</sup>We use the term competition challenge throughout the paper. A challenge may be a trivia question, deliverable, or achievement. Typically these challenges are point scoring tasks.

mobile device. We chose to investigate the mobile device option for two reasons: rack-space may not be available to students at all universities, and a mobile option can include a mobile network which will quickly connect a physical lab- or group-environment. The mobile configuration we recommend also has the ability to be rack-mounted.

If students have access to rack-space then using a rack-mounted, dedicated metal, machine is the best option. Some of the software components described in the following section work best when they can be accessed before and after the competition. Whereas, a mobile device may change network addresses on campus and may not be accessible at all times, a stationary device will be able to be persistently reached at the same network address. A goal of RTFn is enabling university participation, thus rack-space is not a viable requirement. Note that RTFn should not be deployed as a virtual machine since it includes a hypervisor; it should be capable of running virtual machines for competitions. We strongly recommend hosting RTFn locally, on a university-owned network, since there is a possibility of logging offensive techniques used during competitions. Related offensive software should be stored for competition use only, labeled, and accessed securely.

RTFn has two hardware requirements: 1) a sufficiently large storage drive for storing competition-traffic captures, competition provided VMs, and associated collaboration data; 2) a CPU fast enough to run a virtual machine, maintain an OpenVPN connection, and generate keys to run a local OpenVPN. We also include optional features: a Wi-Fi B/G network interface, multiple Ethernet interfaces, and routing capabilities. The optional requirements enhance the mobile deployment option. We recommend using a Soekris computer to implement both the requirements and optional features. The Soekris net5501 [20] can be used as a stand-alone computer or racked with a special attachment.

## 5. RTFn: Software Components

In this section we describe the software components of the Rock The Flag network. The components can be divided into two groups: collaboration and reporting. We have identified that improvements to collaboration during cybersecurity competition will positively effect outcome. Based on related work, we also found that report generation, statistic tracking, and performance evaluation will also improve competition play. We describe each software component of RTFn as it relates to collaboration or reporting. Finally we conclude with a discussion of software security.

### 5.1 Collaboration

Robust project management software like Redmine [11], which combines wiki-style documentation and SVN support, present a particularly lucrative solution to information organization. During cybersecurity competition, however, time management is paramount. Wiki-style document editing,

while used almost ubiquitously in information collaboration, is a time consuming process. Multiple participants may be simultaneously trying to update the same page, which may cause versioning conflicts. This type of information sharing also creates an overhead to those who are unfamiliar with the markup language syntax, which could cause time loss during the competition.

To solve this problem, RTFn implements a custom implementation of EtherPad [9], a web-based real-time collaborative document editor with chat support. Several additions are made to the EtherPad code base to include the following features, outlined in the following sections.

- Challenge ownership
- Related file uploading
- Meta-data labeling and challenge tagging

#### 5.1.1 Challenge Ownership

To increase the overall efficiency of challenge completion during competitions, it is imperative for all participants to communicate and gain awareness of workload distribution. Without this awareness, competitors run the risk of duplicating already-completed work. Work distribution provides a sense of organization during the competition and allows competitors to focus their efforts intelligently. Competition challenges are often solved by multiple team players; unfortunately these challenges also suffer repeated work, which wastes a team's valuable time. Team members should not have to work through the same preliminary steps to solve a challenge. Furthermore, a second team member should be capable of picking up where another has finished by utilizing the collaborative document editor.

The challenge ownership feature is focused on improving team performance for competitions that last multiple days. Instead of requiring all students to be physically co-located, RTFn encourages distributed play. Coordination of tasks, and summaries of completed work, are provided by implementing ownership. Such that, if a student begins work on a challenge, they are assigned ownership; once they complete or exhaust their ability to continue the challenge, they can release ownership. This happens discretely, allowing students who play in different locations and at different times to keep their work and assignments synchronized.

RTFn's EtherPad implementation includes a dashboard of the challenges currently being attempted. Figure 1 shows a mock dashboard with 7 challenges; where (Ti) is the challenge title, (Tw) is the challenge type and (O) are the challenge owners. Each challenge shows a time counter, and allows an owner to mark the challenge as difficult or solved. When solved, the counter is paused. This dashboard identifies the "owners" of a particular challenge and allows other participants to quickly jump between questions. This mechanism, however, does not prevent participants from editing challenges being completed by other participants.

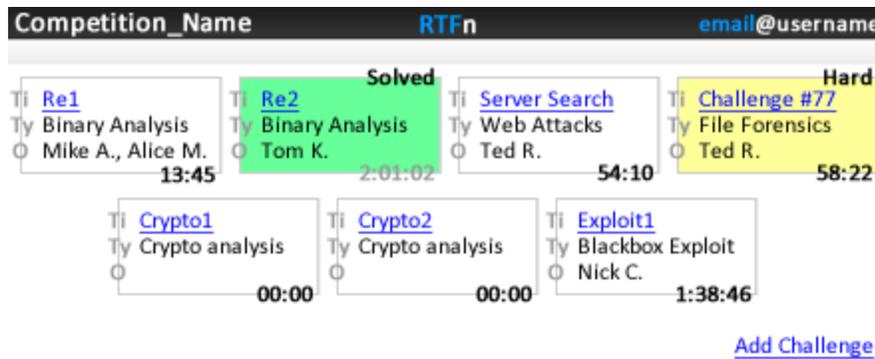


Fig. 1

A WEB VIEW OF THE CHALLENGE BOARD CREATED USING ETHERPAD.

In this scenario, competitors simply share ownership of the challenge, as shown in the first challenge.

### 5.1.2 Related file uploading

The original code base for EtherPad does not include functionality to support file uploading. We implement this feature to further aid in organization during the competition. It is very common to receive virtual machine images, PDFs, JPEGs, and other binary files during the competition for analysis and exploitation. To supplement the EtherPad code base, our file upload addition will be on a per-competition and per-challenge basis. That is, participants will have the ability to upload a file specifically related to whatever challenge they are attempting or for the more-general competition. When competitors switch between challenges, they will be able to view and download any files related to that particular challenge. When players view past competitions they'll be able to review general files, such as instructions, story-line documents, and team guides.

### 5.1.3 Meta-data labels and tags

Historical information retrieval provides insight into several facets of past competitions. Being able to search through past competitions, using data labeling, allows participants to observe past challenge strategies, tools, and general problem solving techniques. This serves as a great starting point for first-time competitors, reducing the lack of familiarity required to solve technically involved challenges. We describe these advantages in section 6. This form of data labeling is also used to trend focus areas of competition challenges. This methodology is discussed in the following section on report outline generation.

RTFn will supplement the EtherPad code base with support for meta-data labeling and tagging. After the completion of a competition, participants will have the opportunity to revisit challenges and tag them with labels. Labels are the higher level area of information security most closely asso-

ciated with the problem. Some example labels could be web application, reverse engineering, code auditing, exploitation, trivia, password cracking, network forensics, cryptography, and social engineering. With appropriate labels in place, our implementation will support a further level of granularity using tagging. Tags will describe useful information regarding the methodology, tools, or related technologies required to complete or related to the challenge. Some example tags could be the related operating system, programming language, vulnerability language, associated CVE, as well as useful tools that were used to complete the challenge.

## 5.2 Reporting

Statistics gathering and report summary generation, performed after a competition, can be just as important and valuable as the preparation performed beforehand. Post-mortem report summary generation offers a holistic perspective into the competition flow [22]. Competition reports may highlight positive and negative actions and provide insight on future decision making. One of the problems RTFn attempts to solve is competition archival and inheritance. Many competitions are held annually, when competing multiple times RTFn can provide players with a refresher summary of past experiences. Report summaries may serve as an important tool for team and competition evaluation. Competitions may also require a report deliverable. RTFn can assist by providing an outline for writing. We implement two new features to EtherPad:

- Report outline generation
- Competition tracking

### 5.2.1 Report Outline Generation

Report outline generation provides the ability to trend competition challenges. By combining challenge outcomes with data labeling during collaboration, RTFn can provide insight on challenge types. A team can identify weaknesses by examining challenge types that are often not completed or

not attempted. A weakness might be a lack of specific subject understanding, experience, or time commitment. Identifying weaknesses is one of the primary goals of cybersecurity competition. This insight can help a team with recruitment, the structure of their practice, and perhaps the feedback provided to the university.

RTFn adds a reporting feature to the current EtherPad implementation, thus allowing competitors to analyze historical details of the competition. First, the reports will contain challenge-specific information such as the number and type of competition goals, with options for presenting higher and lower levels of granularity. After several competitions, teams will be able to trend goal patterns (e.g. offensive and defensive) to better predict and prepare for future iterations. Additionally, reporting will feature a timeline of events. Participants will be able to see a time breakdown of the competition, with a detailed view of how long each participant spent on each question and whether or not their effort led to a solution.

### 5.2.2 Competition Tracking

In order for a university's cybersecurity team to be successful, its participants need to constantly be aware of upcoming competitions. This awareness will allow for better planning and preparation during the weeks preceding the competition. With this knowledge, teams can accurately plan practices, scrimmages, and exercises to flesh out areas that may need more attention going into the competition.

RTFn recognizes the benefit of competition tracking; it implements a calendar add-on to EtherPad. With this functionality, teams will be able to mark important registration and competition dates, and plan their practices accordingly. The calendar framework will also send out email reminders to participants when an event nears. Competitors will also have the ability to export RTFn's calendar in iCalendar format to sync personal calendars.

The calendar feature's real power comes from a managed RSS feed of competitions. We maintain a list of security-related competitions and contests. Local RTFn installations can optionally poll this list and update their calendars. We imagine an interface for this public RSS feed that allows competition organizers to post their events. An example of the XML retrieved from the managed RSS is shown in figure 2, this shows what information is contained for one competition. This should allow smaller competitions to gain popularity. However, the main goal of this managed RSS feed is to provide awareness to new competitors. This removes the overhead of discovering competitions, which aligns with RTFn's goal of removing organization overhead from competitions.

An example of the types of events that may appear in the feed may include:

- CSAW 2010 [15] - September 24-26th
- iCTF 2010 [2] - December 3rd

```
<title>East Coast Cyber-CTF</title>
<description>
  A security contest for high school
  and undergraduate students on the east coast.
  The first round contest will be held online at
  http://ecctf.example and the finals will be held
  in Washington D.C.</description>
<participation type='remote' method='OpenVPN' />
<time start='May 4th, 2011 HH:MM:SS'
  end='May 5th, 2011 HH:MM:SS' />
<duration hours=24 />
<repeat annual=4 />
<pastinfo>
  <stats number_competitors=18 />
  <winner>Computer Security University</winner>
</pastinfo>
```

Fig. 2

EXAMPLE OF XML RETRIEVED FROM THE RSS FEED OF  
COMPETITIONS.

- Plaid CTF 2011 [4] - April 22-24th
- ISTS 2011 [16] - April 1-3rd
- ruCTF 2010 [10] - December 14th
- CODEGATE 2011 [5] - March 3-4th
- Defcon CTF [7] - Mid-Summer

## 5.3 System Security

Because of the offensive nature of CTF-style competitions and the capability for many of the stored tools to be used maliciously, it is imperative to securely store all aspects of the RTFn. RTFn's features work together during competition time to foster successful participation. The adapted EtherPad implementation, the repository of tools, the archive of old reports, the competition facing web servers, have a specific time that they are used in the realm of competition. The collaborative EtherPad software and competition facing web servers are used during competitions; the reports are used after the competition to analyze performance and before competitions to help prepare for future competitions. These features have a distinct purpose and it is important that they do not become leveraged for a counterproductive, malicious nature.

## 6. Campus Involvement

One of the most important features of RTFn is its ability to improve campus involvement and participation in CTF-style competitions. The many components of RTFn support the ease of getting involved in an extra-curricular cybersecurity organization and participating in competitions. The meta-data labeling and tagging system works to facilitate education and training. Also, students using RTFn for the first time are supplied with a wealth of organized data to browse and benefit from. The meta-data labeling and tagging system, as well as the file uploading system supplies students with a repository of challenges from previous competitions.

If a student wanted more ways to become involved in CTF-style competitions, they could browse the reports made by the collaborative document editing software. This will show new team members how specific problems were solved. This idea of community based self education removes the internal feeling of competition between team members by ridding the need of a team leader. It also promotes the idea of community based improvement and team building. The collaborative document editing software also facilitates task continuation and coordination with distant team members, making it easier for people to work together, regardless of where they might be located on campus.

Report outline generation and competition time lines offer students a detailed look into what competition is actually like. Students will be more comfortable with participating in a competition once they are more informed about the structure of it and will be able to create goals for themselves using the reports based on improving their weaknesses. The time line will assist in reducing the learning curve that often comes with CTF competitions by giving students a realistic expectation of time-based requirements.

In addition to making it easier for students to become involved in CTF-style competitions, RTFn also enables students to be more willing to become involved. Students feel more comfortable getting involved in a competition where there exists a plethora of information about what to expect. Often times, students are more comfortable participating from the comfort of their own dormitories, apartments, houses, etc., especially over the course of a several day competition. Given this, students will be inclined to take advantage of the collaborative document editing software features.

RTFn grants team leaders the ability to focus on the competition and assist other students instead of having to act as a systems administrator. The added organizational structure increases ease and fun, making students more inclined to participate. Since the software is open source and all materials are made available to everyone, fair team building is promoted. Also, since one of the biggest necessities of RTFn is collaboration, it is able to promote inclusion of newer team members and keep all students involved.

## 7. Conclusion

Early implementations of RTFn have been deployed on a home router and on a virtual machine running pfsense. From these deployments we created most of the hardware and software requirements outlined in this paper. Our team of students competed in CTF-related events for the first time and evaluated these deployments. Using RTFn, as described in this paper, enhances collaboration and productivity during CTF competitions. The hardware and software combination also enables a university to quickly gain a competitive advantage, record their performance, and evaluate their

weaknesses. RTFn also provides valuable information to help improve undergraduate cybersecurity programs.

## 8. Future Work

To determine the effectiveness of our approach, we will rely on user experience data and performance monitoring. Observation during live competition is the most effective way to generate this data. This is accomplished by placing data monitors at different locations inside RTFn. We plan on offering RTFn to multiple universities participating in capture the flag competitions. To protect data privacy we will clearly explain what data will, and will not, be collected by RTFn. While the type of the data logged by RTFn is not sensitive in nature, participant awareness and disclosure is warranted.

We will monitor the performance of the various hardware components including the network interface(s), RAM, and CPU. For the network interface(s), we will monitor the total number of packets received versus dropped to help us gauge the amount of traffic routed during a competition. From this data, we can make improvements to RTFn to support a more reliable network interface card if required. In similar fashion, we can monitor CPU and RAM usage to better understand the processing load during peak competition involvement. To gauge the effectiveness of our customized EtherPad implementation, we will rely on user feedback; this interaction will help us better understand which are the most useful as well as possible component additions to be made.

We also plan to incorporate RTFn's modified EtherPad code base into a distributable disk image. Then, teams will not have to install the required software and troubleshoot any complications that may arise during the process. This will reduce the overhead for team organizers and ultimately foster more participation.

## References

- [1] AMAN, J. R., CONWAY, J. E., AND HARR, C. A capstone exercise for a cybersecurity course. *J. Comput. Small Coll.* 25 (May 2010), 207–212.
- [2] CHILDERS, N., BOE, B., CAVALLARO, L., CAVEDON, L., COVA, M., EGELE, M., AND VIGNA, G. Organizing large scale hacking competitions. In *Proceedings of the 7th international conference on Detection of intrusions and malware, and vulnerability assessment (2010)*, DIMVA'10, pp. 132–152.
- [3] CMU: CYLAB. Cylab: Confidence for a networked world. Website. <http://www.cylab.cmu.edu/>.
- [4] CMU: PLAID PARLIAMENT OF PWNING. pCTF2011. Website. <http://www.plaidctf.com/>.
- [5] CODEGATE. YUT Qualls. Website. <http://yut.codegate.org/>.
- [6] CONKLIN, A. Cyber defense competitions and information security education: An active learning solution for a capstone course. In *System Sciences, 2006. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on* (jan. 2006), vol. 9, p. 220b.
- [7] DEF CON COMMUNICATIONS, INC. DEFCON, 2009. Website. <https://www.defcon.org/html/links/dc-ctf.html>.
- [8] DODGE, R., HAY, B., AND NANCE, K. Standards-based cyber exercises. In *Availability, Reliability and Security, 2009. ARES '09. International Conference on* (March 2009), pp. 738–743.

- [9] ETHERPAD. EtherPad Foundation, 2011. Website. <http://etherpad.org>.
- [10] HACKERDOM. RuCTF, 2010. Website. <http://www.ructf.org/>.
- [11] JEAN-PHILIPPE LANG. Redmine, 2011. Website. <http://www.redmine.org/>.
- [12] LEE, C., ULUAGAC, A., FAIRBANKS, K., AND COPELAND, J. The design of netsecclab: A small competition-based network security lab. *Education, IEEE Transactions on* 54, 1 (feb. 2011), 149–155.
- [13] LI, P., LI, C., AND MOHAMMED, T. Building a repository of network traffic captures for information assurance education. *J. Comput. Small Coll.* 24 (January 2009), 99–105.
- [14] MINK, M., AND FREILING, F. C. Is attack better than defense?: teaching information security the right way. In *Proceedings of the 3rd annual conference on Information security curriculum development* (2006), InfoSecCD '06, pp. 44–48.
- [15] NYU POLY. Cyber Security Awareness Week. Website. <http://www.poly.edu/csaw>.
- [16] NYU POLY: ISIS. The information systems and internet security (isis) laboratory. Website. <http://isis.poly.edu>.
- [17] O'LEARY, M. A laboratory based capstone course in computer security for undergraduates. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education* (2006), SIGCSE '06, pp. 2–6.
- [18] POTHAMSETTY, V. Where security education is lacking. In *Proceedings of the 2nd annual conference on Information security curriculum development* (2005), InfoSecCD '05, pp. 54–58.
- [19] RIT: SPARSA. Security practices and research student association. Website. <http://www.sparsa.org/>.
- [20] SOEKRIS. net5501. Website. <http://soekris.com/products/net5501.html>.
- [21] UCSB: SECLAB. The computer security group at ucsb. Website. <http://www.cs.ucsb.edu/~seclab/>.
- [22] WAGNER, P. J., AND WUDI, J. M. Designing and implementing a cyberwar laboratory exercise for a computer security course. *SIGCSE Bull.* 36 (March 2004), 402–406.

# Using the Castle Metaphor to Communicate Basic Concepts in Cybersecurity Education

G. Markowsky and L. Markowsky

Department of Computer Science, University of Maine, Orono, Maine, USA

**Abstract** - This paper explores how to use the castle as a metaphor to help students and non-technical users understand some basic concepts of cybersecurity. Castles are symbols of security that are familiar to and easily understood by most people. Important defensive structures for many centuries, castles were designed and built using much ingenuity and effort and are not the simple-minded structures that many people imagine them to be. This paper describes the design of castles in detail and shows that many of the techniques used by castle designers are still relevant today and can provide a concrete embodiment of important cybersecurity concepts.

**Keywords:** security, security architecture, cybersecurity education, active defense, intrusion detection and prevention systems

## 1 Introduction

Castles have long inspired people of all ages. To many people they embody the idea of security. In this paper we examine some of the ways that the castle can be used as a metaphor to teach basic concepts of cybersecurity to a general audience.

In [1], McDougal suggests that there are valuable lessons to be learned from studying the defensive systems of castles and presents some strategies based on these lessons. In this paper we study the defensive systems of castles in more detail and make more detailed comparisons to cyber defense. We also point out the dangers of having too simple an understanding of castles and thereby not benefiting from the lessons learned from the hundreds of years of experience acquired by castle builders.

## 2 A simple view of castles

Figure 1 shows the cover of the February 22, 2010, issue of InformationWeek [2]. Note how primitive the castle is – just a simple wall surrounding three people who are armed with bows and arrows. The castle shown in Figure 1 is more of a liability than an asset since the people in the castle have no windows to look out of and no platforms along the wall that can be used to defend the castle. Also there is a strange figure suspended over the castle by a crane not shown in the picture.

Real castle walls are not simple structures, but intelligently designed defensive systems. Castle walls have



Figure 1. Simplistic View of a Castle

platforms from which the defenders could resist the attackers and get some shelter. The model of the castle in InformationWeek is essentially a model of a prison for the people inside the walls.

There are at least two additional problems with the castle in Figure 1. First, the castle is pictured sitting in the middle of a featureless plane. Real castles typically were placed in strategic locations so that they either controlled some passage or at least had a good view of the surrounding area. Another problem is that the image is reused in the article with the word “outflanked” superimposed on the image. This is completely nonsensical since a circular castle has no flanks and cannot be outflanked. The care with which castles were located and designed leads us to the first lesson in cyber defense: have an overall plan. You should not build defenses in isolation. Like a castle builder, you should understand who your enemies are and how you are likely to be attacked.

People often can't think of “enemies” that they might have in cyberspace. It is possible to have both enemies of a personal nature as well as impersonal enemies for whom you and your organization are targets of opportunity. Businesses should be concerned with all competitors: local, national and international. They should also be concerned about insider threats originating from disgruntled employees and jealous colleagues. Individuals should be concerned with cyberthieves, botnet masters, partners and ex-partners. Even “friends” can be a source of trouble.

### 3 Real castles

Castles have a great deal of individuality because they were built: (a) in places that are geologically very different from one another; (b) at different times; (c) for different purposes; and (d) by people having widely varying resources and time. The photos shown in Figures 2 and 3 highlight the defensive systems of Malbork Castle in Poland.



**Figure 2. Part of the Outer Wall of Malbork Castle**

Figure 2 shows one of the towers that defends the castle's outer wall. Notice the slits, called arrow loops, for firing arrows at attackers who are near the wall. Notice also that the castle's outer wall is designed so that water is drained to the outside rather than the inside of the wall.



**Figure 3. A Guarded Entrance at Malbork Castle**

Figure 3 shows one of the entrances to Malbork Castle. Notice the windows and other openings that overlook the approach to the entrance that enable the castle's defenders to attack enemies approaching the gate from a relatively protected location. Finally, notice that the entrance has gates at both ends. A portcullis is shown in the foreground and the gate at the far end is shown swung open to the left.

The modern firewall functions much like a main wall of a castle. Castle builders understood that any opening in a wall introduces a weakness into that wall. At the same time, it is

not reasonable to build a castle without doors and windows. In the same way, a firewall must have doors and windows so that the computer can communicate with other systems over a network. Openings in the firewall are often known as ports. Services (such as web services and e-mail) have ports that must be kept open in order to be useful.

While castle builders knew the value of entrances, they also understood the vulnerabilities that entrances introduce. Consequently, there were mechanisms to ensure that any attempt to force entry would be strongly resisted. Castles typically had small entrances called postern gates that could be used to escape or to communicate with a boat landing. They often had disguised gates that could be used for raids against the enemy. These gates needed additional defenses to discourage the enemy from following the raiding party too closely back into the castle. A common form of protection was a machicolation. This is basically a collection of slits in the ceiling of an entry way that would permit the defenders to drop objects or pour liquids on anyone in the entry way.

We do not advocate such aggressive defense for the average user. For one thing, attacks of various sorts are illegal and the average user does not want to risk violating the law in defending a system. At the same time users should realize that openings need to be defended. For that reason good firewalls have special rules that define what information flow is allowed through a port. One place where the average user can take some defensive action is to make sure that there are strong passwords on any wireless devices that are deployed including on the control screens. Do not run devices using just the default passwords – this is like having a castle and leaving the door unlocked.

**Table 1. Some Castle-related Terms**

<ul style="list-style-type: none"> <li>▪ arrow loop</li> <li>▪ bailey</li> <li>▪ barbican</li> <li>▪ bartizan</li> <li>▪ batter</li> <li>▪ battlement</li> <li>▪ brattice</li> <li>▪ chapel</li> <li>▪ chemise</li> <li>▪ corbel</li> <li>▪ corner tower</li> <li>▪ covered parapet walk</li> <li>▪ crenelation</li> <li>▪ curtain wall</li> <li>▪ drawbridge</li> </ul>	<ul style="list-style-type: none"> <li>▪ embrasure</li> <li>▪ flanking tower</li> <li>▪ footbridge</li> <li>▪ foundation</li> <li>▪ garderobe</li> <li>▪ great hall</li> <li>▪ hoarding</li> <li>▪ inner curtain</li> <li>▪ inner ward</li> <li>▪ keep</li> <li>▪ lists</li> <li>▪ machicolation</li> <li>▪ merlon</li> <li>▪ moat</li> <li>▪ outer curtain</li> </ul>	<ul style="list-style-type: none"> <li>▪ outer ward</li> <li>▪ palisade</li> <li>▪ parapet walk</li> <li>▪ pinnacle</li> <li>▪ portcullis</li> <li>▪ postern</li> <li>▪ postern gate</li> <li>▪ putlog hole</li> <li>▪ rampart</li> <li>▪ stockade</li> <li>▪ truss</li> <li>▪ turret</li> <li>▪ wall walk</li> </ul>
--	--	---

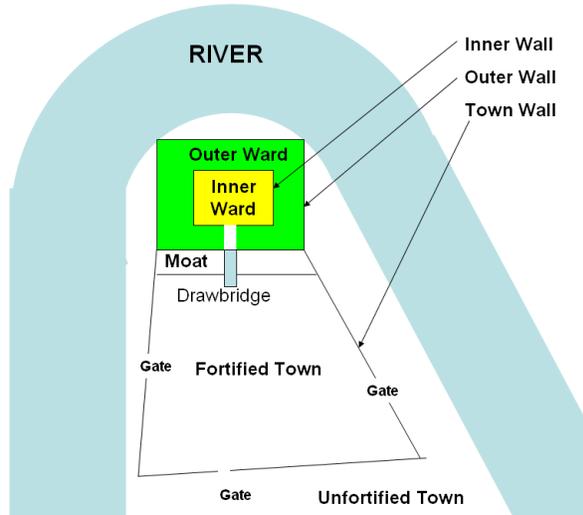
The many castle-related terms in Table 1 further illustrate the complexity of castles. These terms describe some of the more common features associated with castles.

For a more complete list of terms, see <http://www.castlesontheweb.com/glossary.html>.

### 4 The overall plan: defensive zones

The book by Macaulay [3] contains many wonderful illustrations showing how a castle would be designed and built. Many of the steps in designing a castle correspond directly to the steps necessary in designing a secure network.

First, the builder of the castle must decide the purpose and roles of the castle. Some castles were part of a town complex, while others were more like fortresses. They also differed greatly in size and complexity, but all used layered defensive zones. In general, a castle might be part of a larger master plan that basically provided for four defensive zones as illustrated in Figure 4.



**Figure 4. Layered Defensive Zones**

The four typical defensive zones shown in Figure 4 – the unfortified town, the fortified town, the outer ward, and the inner ward – were often separated by three sets of walls that differed in purpose and structure. The first set of walls that an attacker might see were the town walls. These would be substantial walls with many towers. The walls would have platforms for the defenders to stand on to enable them to engage the attackers from above.

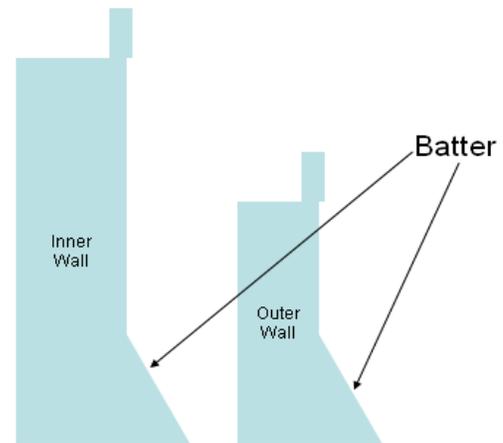
The first protected zone would be the fortified town zone, which would include all the land enclosed within the town walls, but which would be outside the outer castle wall. The outer ward would be the zone consisting of all the territory within the outer wall, but outside the inner wall. The inner ward, of course, would be the part of the castle that would be within the inner wall. Of course, a castle could have more walls than two. Even within the inner ward there could be additional fortified towers and keeps (inner strongholds) for additional security.

Castle designers had no desire for the enemy to get within their castles. Nevertheless, their designs allow for the possibility that the enemy would get in. Like secure networks with layered defensive systems, they made sure that each time the enemy got past one set of defenses, the enemy would encounter yet another set of defenses.

In addition to planning at the town level, castles were often part of a larger plan. In particular, Fedden makes the following observation [4; p.31]: “The Crusader castles would have been formidable enough as isolated units. They acquired additional strength in being linked by an elaborate system of communication with neighbouring strongholds.” Communication at that time was via carrier-pigeon and signaling.

## 5 Details of castle design

There are many interesting details in castle design. For example, castle walls were typically not straight. They had a slanted section near the bottom called a batter.



**Figure 5. Concentric Walls With A Batter**

Figure 5 shows that typically the inner wall would be taller and sometimes thicker than the outer wall. Notice the sloped section of each wall. This was called the batter and it was angled for two reasons. First, any ram hurled against the base of the wall would find itself deflected somewhat so that the full force of the ram would not strike the wall. Second, anything dropped by the defenders from the overhangs, called hoardings, at the top of the wall would be deflected onto the attackers.

Castle designers also introduced such features as drawbridges, twisted passageways, and planks that connected different sections of the wall. These planks could be removed so that if one section of wall was taken, the attackers could not easily get to other sections.

Some particularly interesting details showing the intricacies of castles can be found in [4]. For example, [4; p. 29]: “A besieger wishing to force the entrance at Krak would have had to proceed up a covered passageway and negotiate three elbow turns, at least one portcullis, and four gates furnished with machicolation.”

We should note that the drawbridge has an analog in cyber defense: “pull the plug.” In particular, it is not necessary to be connected to the Internet at all times. Users should consider cutting their connection to the Internet when there is no need for it.

It is interesting to note that an ancient security device survives to this day – *the password*. Since ancient times, people have employed passwords to distinguish friend from foe and to limit access to people who were trusted. Like modern passwords, castle passwords were modified on a regular basis, and different passwords might be used to access different locations within the castle.

## 6 Conquering a castle

Attacks against castles succeeded primarily for the following reasons:

1. Lack of Manpower
2. Psychological Pressure
3. Famine
4. Siege Weapons
5. Insiders and Trojan Horses

Some castles were too large to be defended by the garrison, or troops, that they had at the time of attack. For cyber castles, the lesson here is to make sure that you have enough staff to operate your castle's defenses. In particular, there should be at least one person who is concerned with the cyber castle's defenses. You might need many more people than one, but zero is never an adequate number.

Other castles fell because of the psychological pressure resulting from being surrounded. Fedden describes two instances [4; p.35-36] in which well-defended castles surrendered because of psychological pressure well before there was any physical necessity to surrender. Closely related to psychological pressure on defenders is the use of ruses of various types, including forged letters. Interestingly enough, ruses of various sorts are used today by scammers of all sorts to gain entry into networks and systems.

Famine was eventually successful in a number of castle attacks, and to defend against this threat, some castles had supplies that would last up to five years. For example, [4; p.10]: "The vast cellars at Margat were constructed to hold a thousand men's provisions for a five-year siege." There were not many armies that were willing to wait that long for success. Maintaining a large host in the field for a very long time is not something that most attackers were willing to do.

A number of siege weapons were useful to besiegers of castles. Some of the common weapons and tools were scaling ladders, earthen ramps, siege towers, rams, and bores. Of special interest was the technique of mining under walls and towers and then causing the mines to collapse along with the walls and towers. This technique was defeated by placing the castle on solid rock or in a body of water so the attackers could not mine. Occasionally, castle defenders dug their own mines to intersect those being dug by the enemy.

Another class of weapon that needs to be mentioned is artillery. The term artillery predates the use of gunpowder and refers to various devices like the trebuchet that could hurl large objects against the castle walls and into the castle. It is interesting to note that one of the defenses against artillery was to install artillery in the castle to be used against the artillery of the attackers. With the advent of gunpowder, the castle evolved into the fortress which no longer was used as a primary residence.

Castles, like computers and networks, have also been victims of insider threats. In [5], the authors of this paper surveyed supercomputer cluster operators and found that 9% of survey respondents reported that someone had tried a

physical approach to disrupt computation or to steal data, and 5% were unsure of whether this had happened. Similarly, 8% of respondents reported that someone had tried to bribe or otherwise co-opt one of the cluster staff into helping with compromising security, and 13% were unsure whether this had happened.

Finally, attackers have long used ruses to fool defenders into letting them within the walls. The most famous example of such a stratagem was the Trojan Horse. Interestingly, this name applies to a variety of malware that is commonly encountered, underscoring the link to the classic stratagem.

## 7 Weapons for the Linux cyber castle

Many analogies can be made between traditional castles and Linux "cyber castles," but significant differences exist as well. First, defending a Linux castle does not endanger the lives of the system and security administrators. Also, Linux administrators can audit their computer systems and networks and can even attack their own systems (or on clones of production systems) without fear of damaging the castle.

### 7.1 Design of the castle and surrounding grounds

The overall design of the castle grounds, the layout of the town wall, the outer and inner walls, and the location and protection of the castle entrances together are analogous to implementing a secure network topology and enforcing an effective security policy. The topology will dictate which systems and services will be available within the "town wall" of the network and what must lie within the "outer" and "inner" walls, and like their traditional counterparts, these concentric walls often become better fortified and more restrictive towards the center.

### 7.2 Town wall

An iptables/NetFilter firewall and PortSentry can be used to implement a strong, active, outer defense enclosing all Internet-facing servers as well as the internal network. Nessus can be used to scan for vulnerabilities, Snort can be used to monitor the town for intrusion or attempted intrusion, and Wireshark can be used to monitor network activity.

### 7.3 Outer wall

Internet-facing servers can be placed in a DMZ, and hosts providing services can confine those services within chroot jails or virtual machines such as xen. Services can be further locked down using application-specific configuration files and application-specific security tools, such as ModSecurity for Apache web servers.

### 7.4 Inner wall

Mounting filesystems with minimal access, such as disallowing suid or write access, and performing filesystem security assessment using Tripwire provides a secure base. Also, as noted earlier, insider threats are a real danger for castles of every sort. Monitoring system logs for unusual behavior using a log scanner such as Logcheck can help to

spot an insider threat. Access control lists, or ACLs, can be configured within some services such as Exim (a mail server), or BIND (a domain name server). Of particular interest are Mandatory Access Control systems such as SELinux and AppArmor, which provide fine-grained access control on a particular host. Using MAC systems can both limit the damage done by successful intrusions and prevent some intrusion attempts from ever becoming successful.

## 7.5 Guarded entrances and postern gates

TCP-wrapped services, configured using `hosts.allow` and `hosts.deny`, can be restricted to particular IP addresses or subnetworks. Dynamically configurable authentication, implemented using PAM, and VPNs also guard the entrances to the Linux castle. A known, protected IP address or physical access to a machine can provide a trusted back door into the system, and finally, backup tools such as BackupPC, Bacula, and `fwbackups` can provide an escape route in the unfortunate event that the castle has been successfully attacked and must be abandoned.

## 8 Critical infrastructure

One area that can benefit tremendously from applying even simple castle defense principles is critical infrastructure protection. The Q & A with Joe Weiss [7] makes it clear that even the simple principle of putting up castle walls still needs to be implemented more widely. Critical infrastructure builders provide many entrances into their structures and would benefit from thinking more deeply about protecting these entrances. Joe Weiss [7] describes the use of Bluetooth to provide utility workers with easy access to electrical reclosers (circuits that can connect and disconnect parts of the electrical grid), but this Bluetooth connectivity is provided without enough consideration of what this access could do in the wrong hands. Critical infrastructure, even more than most networks and systems, must be designed to prevent easy access by people who should not have access.

One of the forces working against implementing cybersecurity measures in critical infrastructure systems is the worry that such measures would interfere with the infrastructure's ability to deliver services. An analogous problem was faced by the creators of castles in that they needed to allow commerce and communication while simultaneously providing a high level of security. The fact that castles flourished for many years provides us with an example that it is possible to balance these competing demands.

## 9 Conclusion

Much can be learned from studying the way traditional castles were designed, constructed and defended. Because they are concrete and easily understood, castles can also provide a valuable metaphor for introducing concepts of cyber defense to students and non-technical users. This paper has presented some of these analogies.

Among the lessons that can be drawn from our review of castles and castle warfare are:

- Start with a good overall plan for the castle and all other entities that must be defended.
- Elements of the defense must be active. A completely passive defense will not survive the challenges and repel attackers.
- The cyber castle must be adequately staffed.
- Use defense in depth and make sure that the inner defenses also support the outer defenses. Be sure to have the equivalent of drawbridges and removable planks. Identify points in the security topology that can be used to quickly isolate zones from the network and from other zones.
- Make sure that the cyber castle has a solid foundation.
- Use every means possible to make the attacker's job more challenging.
- Know your attackers. It is important to get some idea of the sophistication of your primary attackers.
- Find a balance between security and service. Castle designers faced this problem and found many successful solutions.

## 10 References

- [1] McDougal, Monty D., *Castle Warrior: Redefining 21<sup>st</sup> Century Network Defense*, Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, Oak Ridge National Laboratory, 2009, [http://www.isiconference.org/2009/MontyMcDougal\\_Raytheon.pdf](http://www.isiconference.org/2009/MontyMcDougal_Raytheon.pdf)
- [2] Davis, Michael A., *Time for a New Strategy*, InformationWeek, Feb. 22, 2010, Cover and pp. 29-34, <http://www.informationweek.com/news/security/management/showArticle.jhtml?articleID=223000132>
- [3] Macaulay, David, *Castle*, Houghton Mifflin Co., NY, 1977.
- [4] Fedden, Robin, *Crusader Castles*, Art & Technics, London, 1950.
- [5] Markowsky, G., Markowsky, L., *Survey of Supercomputer Cluster Security Issues*, Proceedings of the 2007 International Conference on Security & Management, pp. 474-480.
- [6] Trost, R., *Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century*, Pearson Education, Inc., Boston, MA 2010.
- [7] Elinor Mills, *Joe Weiss, crusader for critical infrastructure security (Q&A)*, CNET News, May 10, 2010.

# Red Teaming for Education

Jeffrey C. Scaparra & Jeffrey R. Bullock

Advanced Cyber Concepts Division

Space and Naval Warfare Systems Center Atlantic

Washington Navy Yard, DC, United States

**Abstract** - *The Collegiate Cyber Defense Competitions (CCDCs) offer college students an opportunity to test their network defense skills in an educational but aggressive setting. Their skills are tested by the probing red team with the primary goal of separating the good students from the great defenders. Often, the red teams act in an unorganized, melee-like fashion. Due to the range of student skill levels, both the read teams' size and expertise, network configuration, scope of attack, as well as the overall nature of the competition, the red team should follow a tested methodology that provides a fair, realistic, and educational experience to the students.*

**Keywords:** A Maximum of 6 Keywords. red-teaming, ccdc, education, security

## 1 Introduction

Every year, colleges and universities around the United States participate in the Collegiate Cyber Defense Competitions (CCDCs). In 2011, there were nine regional events – winners of these competitions travel to the National Collegiate Cyber Defense Competition. Often, several states will have qualifying events for the regional competitions.

Coming into the 2011 CCDC season, there was much effort put into standardization of the competition, from a format perspective. The hope is that students at each of the regionals will have more similar experiences that will help to level the playing field among the regions. In the past, regionals have been able to build and design their own event, which in some cases varies greatly from the national event. While this effort to standardize is still ongoing, there has been more cohesion.

The purpose of the event is for student teams to evaluate their ability to defend a network, and thus, an unbiased red team must exist to play the role of “real life threat” to their network. In past years, the red teams at regionals were made up of instructors and local hackers known by the coordinators. Consequently, red team composition led to differences amongst the competitions. The quality and quantity of the volunteers for the red team vary by regional. Some regionals, with as many as 12 blue teams, may have as little as five or six red team members to attack. One of the primary methods for a student to learn is from the red team intrusion reports. The

amount of reporting and types of feedback also vary by region, from “no feedback,” to more commonly a PowerPoint or oral presentation at the end of the competition covering only what the red team members remembered doing. Additionally there were differences in ideology amongst the different red teams and regions. In some cases, the red team wiped boxes before the teams had a reasonable amount of time to secure or identify weaknesses in their computer systems. The tools available for penetration testing do not automatically cross over well for this type of red team methodology. Since one person may gain primary accesses, and because the team skill levels may vary, more collaborative tools are necessary.

This paper is a look at the issues related to running a red team in an educational environment. While this paper looks specifically at the CCDC events, the concepts and ideas should transition to other situations as well. Jeff Scaparra and Jeff Bullock have coordinated and participated on multiple regional CCDC red teams across the United States, including the National CCDC. This paper will describe a suggested and tested methodology to red teaming for education as well as address some of the needed research areas.

## 2 CCDC Background and Purpose

“CCDC provides direct feedback for schools to exercise, reinforce, and examine their security and information technology curriculum.” [1]

The primary purpose for the CCDCs is the educational value. Often, students will learn in theory how to operate and defend a network in a classroom setting. Lab environments allow students to follow a step-by-step procedure to manage a router, install an operating system, configure a web server, manage Active Directory, etc. The CCDCs not only give the students one single experience to tie all of their knowledge together, but also provide them an exciting, competitive, and mostly realistic setting to do so. The competitions test the students' professionalism as well; business injects given to the students task them with writing policy for social networking at their “company,” writing incident reports after an attack has taken place, even quick memos to their “CIO” for higher-level questions – documents that typically need to be written with less technical terminology and more simple, graphical

messages. A lot of work goes into running a CCDC – the roles are typically divided up into different color-coded teams.

### 3 Team Explanations

#### 3.1 Gold

The gold team is responsible for the overall organization and healthy operation of the competition. The gold team is often composed of professors, directors, and veteran students who have an unbiased stake in who wins the competition. Typically, the gold team has the “final say” in the conflicts and tough decisions that must be made prior to and during the event.

Gold team members handle the coordination at the host facility so that the event has the necessary space requirements. Along with this duty entails the public advertisement of the event, to draw enough blue teams from various local colleges/universities to attend the event. Since most events are not sponsored fully by the host facility, the gold team seeks to find sponsorship from those interested in industry and government.

Finally, the gold team will build and monitor the scores, all from a combination of red team attacks, business injects, and the scoring engine. Since the scoring engine relies heavily on the health of the network infrastructure, there is often much coordination between the gold team and the black team.

#### 3.2 Black

The black team is responsible for the network infrastructure that the red/blue teams will operate on. Decisions must be made by a combination of the black and gold teams as to whether the students' machines will be physical or virtual machines – this decision almost always relies on the available funding from sponsors, or the available equipment at the host facility. The black team will prepare a fair and above all equal network for each of the student teams. All of these teams will connect to a hypothetical “headquarters” and they go out to the Internet. The black team also maintains the red teams' “unknown” connection location to the network.

#### 3.3 White

The white team is responsible for judging and monitoring the student teams, they are not competing. They make sure students follow the rules explicitly outlined in the team packets. The white team will sometimes deliver the actual business injects to the team if the gold team has not already automated this process; similarly, there are often business injects that the white team will score on the fly. For example, the gold team may instruct each room's white team member to ask the team to unman their Microsoft Vista workstation so they can attempt to buy something from the e-

commerce website they host, or to send an e-mail. A grading rubric provided to the white team by the gold team will strictly explain how to score based on expected or unexpected results from their actions.

#### 3.4 Blue

The blue teams are the competitors – they are the student teams of seven to eight individuals, enrolled at their respective college or university, that are responsible for defending their network. Some competitions are setup in a structure in which each blue team has their own room; other competitions put all of the blue teams together in order to witness the chaos of the other teams. Teams are responsible for keeping services alive (i.e. Mail, Web, DNS, AD). This task can become daunting when they are given business injects every 15-60 minutes from the white or gold team. Finally, the blue team must do their best at completing these tasks while the red team tries to slow them down to a halt.

#### 3.5 Red

The red team, also not competing, is responsible for providing the “real world threat” from hackers, corporate espionage and script kiddies to the teams. Red teamers are often invited from across academia, industry and government, to provide a realistic arsenal of attacks and exploits equally to each of the blue teams. The red team must submit reports after they successfully:

- retrieve usernames/passwords
- retrieve sensitive information (personally identifiable information or proprietary information)
- obtain root or administrator access to any system
- maintain root or administrator access to any system
- deface web sites
- remove or kill services or hosts

### 4 Scope of Attack

Depending on the rules of the competition, as outlined by the gold team, the red team has different scopes of attack.

#### 4.1 Network

The network is always within the scope of attack, to an extent. If the competition is setup in a way the teams are only responsible for their virtual machines, the red team can only “create havoc” on these machines. If red teamers manage to locate the host machines, they are often “off limits;” however, some gold teams allow the red team to hit the host machine at a certain point through the competition. Again, this decision primarily relies on whether the decision to use virtual machines was for lack-of-equipment purposes, or whether virtualization security is within the scope of the competition.

Another often-deliberated appliance are the routers and switches. Sometimes the red team will have access to the routers and not the switches, and sometimes they are allowed to breach both.

## 4.2 Physical

How realistic does the gold team want the competition to be? And where is the line drawn? Some competitions will allow the red team to enter the student team rooms during the competition, as well as after hours when the teams are not on-site. This action is to simulate a disgruntled employee or burglar entering the building by gaining physical access. Decisions in this realm may lead to the red team discovering post-it note passwords, white-boarded game plans, or even actual computer interaction with an infected thumb drive..

## 4.3 Social Engineering

The weakest points of security in an organization are often the people. Mark Richardson, CTO of a Medical Records Company, says that "we've met the enemy, and it's us." [2] For this reason, the red team is often given the opportunity to social engineer the teams. Sometimes the red teams are given a phone in which they try to impersonate the gold, black or white teams. Sometimes, they will attempt to enter the team rooms, only to be met with a blue team "bouncer" who should immediately deny access. Typically, the line is drawn by not allowing anyone to create fake badges distributed at the beginning of the competition by the gold team; otherwise, a "trust no-one" mentality can be presented to the gold and black teams who are legitimately trying to operate the competition.

## 5 Red Teaming in Phases

Red teaming for educational purposes should give students opportunities to show their knowledge and provide challenges that will help to prepare them for the real world. In order to facilitate this process, a memo entitled the "Red Team Manifesto" went out to the Google Group "Hack CCDC," as well as the coordinators of the regionals, to outline a plan of attack so that a balance might be found to give the students opportunities to find the red team *before* they began destroying the data on the systems.

To allow teams that were better at detection to have an advantage over teams that were not as capable at detecting malicious activities on their network, the red team established a plan of attack that slowly progressed from very stealthy to very loud. This methodology also facilitated red team activities because if the students found all of the red team intrusions and patched all the vulnerabilities on the network, it would make the red team's role very difficult for the remainder of the competition; it also made the competition more realistic.

## Stage 0: Footprinting/Scanning

Stage 0 and Stage 1 start at approximately the same time, but are slightly different efforts. To avoid creating obsessive noise from the red IP addresses, there was an effort to randomize scanning by using a scanner machine that hops around IP addresses and makes the scans available in XML format. Most of the red team tools can import that data, or, red team members can view data with a web browser. This technique is made possible because of nmap's [3] unique xml-output capability. After these scans have taken place, the scan data is used for attack. Scanning continues over the course of the competition since blue teams often change their infrastructure.

### 5.1 Stage 1: Gain Access and Keep it

Stage 1 is usually the entire first day, during the most common competition length of three days. Many of the machines the students receive are full of vulnerabilities and misconfigurations; the goal is to find and exploit them across all the teams as quickly as possible. Once in, the red team plants a variety of backdoors in hopes of retaining access to that machine for the remainder of the competition.

Some red teams operate by assigning each red teamer to a specific student team, sharing accesses once found. This approach can work well if all red team members are at about the same skill level; however, most teams have a range of skill sets. Our suggested strategy is to split up services by familiarity. In this setup, hackers with expertise in web application exploitation could concentrate on the web servers while others were free to explore their niche. Team members were also encouraged to not simply start on team one and progress numerically, but rather decide on a random order and script where possible. Because system users often have poor passwords, it is possible to script an attack that will attempt to log in with common username and password combinations against all teams, followed by placing a backdoor. These types of scripts were used on both Windows and Linux systems with great success. Because they were automated and ran very quickly, the possibility of missing a window of opportunity was small, maximizing fairness.

Many different approaches are used for backdoors and as the competition evolves, so will the strategies of the red team. Teams that have competed years before know what to look for. For Windows systems, backdoors vary from creating accounts, installing meterpreter[4], Core Agents[5], to custom agents written by members of the red team. Additionally, there are cron jobs that send back netcat shells [6] to red team hosts. Similarly, Linux hosts are backdoored with user accounts, SSH keys so that password-less logins are allowed, netcat shells, call back services that restart at boot up, and secondary login services not requiring user/password authentication. Additionally, in an effort to remain silent, logs

are cleansed - histories for the sessions created by the red team members are erased. Some key files are also made immutable, so they cannot be easily changed, including the `/etc/passwd` and `/etc/shadow` files that contain the users that the red team adds. Another method of attack involves taking control of DNS servers, with the goal of blacklisting all update sites - making it difficult for students to get updates to their machines.

The red teams' only other mission on day one is to steal data. Data that is of interest to the red team includes: databases, web site content, usernames, password hashes, documents in the administrator's home directories, and configuration files.

## 5.2 Stage 2: Deny, Deface, Disrupt

If students do not find the backdoors, or plug any of the vulnerabilities in their systems that allow the red team access on the second day, the red team will get a little louder. Teams that notice the red team on day one and patch their systems are consequently at an advantage; however, for the teams that do not notice the red team, they now have the opportunity to see that they are under attack and correct the problem.

This stage usually starts on day two and goes on for most of the day. During stage one red team members are discouraged from altering any data that could potentially affect the students' abilities to complete injects or keep their uptime. By making these alterations during stage 2, the students "business" is affected and they will be more likely to notice. While stage 1 is analogous to an espionage operation, stage 2 is representative of a more hostile attack by adversaries that want to discredit or hurt the bottom line of a company.

During this phase, students see a barrage of defacements. Web pages for the company turn into tributes to Charlie Sheen and the red team. On hosts where red team owns the DNS, the red team begins redirecting common URLs to other places. Search sites, Google, Yahoo!, Bing, and Ask often forward to Baidu, a popular Chinese search engine. Wikipedia requests might bring up a page of the Russian Government; Microsoft might forward to kernel.org. These attempts are solely done to be loud and give the students a hard time to troubleshoot the problem. Other services are altered in ways that make them still run, but not work for outside users.

## 5.3 Stage 3: NUKE, and Annoy

Finally, in stage 3, anything short of a packet flood from a red team computer is allowed. The red team often floods teams' internal network, or brings them down with ARP spoofing. Additionally, any computer that has not been secured is subject to being wiped. Some members of the red team even go as far as to modify the boot loaders so that the kernel images are corrupt and make the boot loader password-

protected. In these cases, the different Linux installs were also renamed to "Red Team Linux" as a way to taunt the students. Better teams create backups and restore their hosts, which take a significant effort. In worst cases, teams request a snapshot of the computer from the white team. This request results in a significant point penalty, and gives them the same insecure system they started with. In these cases the red team would be given instructions to not attack the restored host for 10 minutes after the restore in the hopes that the team would look more closely at the system and secure it.

Systems that the red team still has user-level access to are fair game for attack. Databases are dropped, web files removed, and ransoms for their return are given to the students. Even user-level accounts allow red team members to bring systems down with fork bombs [7]. If the partitioning allows users to fill the hard drive, large files are created to fill up drives. In some cases the red team can pop up messages on Windows hosts and make the computer unusable by simply annoying the user with messages. Stage 3 is a more intense version of the stage 2 attack.

# 6 Reporting and Fairness

Ensuring that the students are able to understand the red team's actions against them during the competition was a primary goal this year. Without reporting, the students cannot learn from their mistakes. In order to collect this information, recording systems such as Google spreadsheets can be used allowing the red team members to easily and efficiently fill out forms to gather the information about an attack.

## 6.1 Importance of Reporting

The function of the red team is not to conduct espionage and attack everything; rather, the goal is to create a real world threat and provide educational value to the competition. In order for the students to learn anything, they must first be able to understand what is happening to them; hence, a good reporting system for red team activities is so important. Nevertheless, there must be a balance. Due to the excitement and quick-pace of the red team activity, it is not always in their best interest to immediately write a report of their accesses, defacements and attacks; nevertheless, their documentation is crucial for a fair and accurate competition. For these reasons the forms have been streamlined and made extremely easy: by having drop-down menus so reporting can be done with a couple of mouse clicks. At regionals that had a SPAWAR-led red team, a Google spreadsheet was used that the gold team was able to access. At the national event, the gold team provided a webpage that was integrated with the scoring engine. While there were some small deviations, the following were recorded:

- Team
- System Compromised
- Level of Compromise

- Method of Compromise
- Time of Compromise

## 6.2 Fairness amongst Teams

In order to promote fairness, we must try out all attack vectors across all of the teams. The team that a tactic gets tried on first is at a disadvantage because they have less time than the rest of the teams to patch that vulnerability. The smaller the time can be made between attacking different teams, the more fair and balanced the red team can be. Because of this issue, red team members are encouraged to carry out attacks on all teams before recording the results. Additionally, other red team members would usually help in attacking other teams after a successful attack.

## 6.3 Conflicts of Interest

In order to have an unbiased red team; they should never be given any information allowing them know what schools they were attacking. Teams are given corresponding numbers, which is how the red team members know them. Consequently, all teams are equal in the eyes of the red team.

## 6.4 Presentations

Although this methodology allows for transparency in the red team reporting, an out-brief is both appreciated and encouraged. While it typically does not go in to full-detail of each team, it does provide some across-the-board items of interest to the teams in regards to common vulnerabilities and mistakes seen. It also gives a chance for the students to ask questions about hacks they noticed during the course of the competition. This presentation, while often exciting for all parties, should not be a “brag and bolstering” of the red team, but rather a “lessons learned” for the blue teams - the competition is not about how well the red team can attack, but how well the blue teams can defend.

## 7 Red Team Tools

Red teaming for education is different than penetration testing in the corporate world. The timeframe is small and the rules of engagement are different. In a traditional penetration test, it is unlikely that a company wants their servers defaced or brought offline. Additionally, during penetration testing, rarely are there multiple systems that the same exact attacks work against. Red teaming for education is all about moving efficiently, furiously and collaboratively.

### 7.1 Collaboration

The red team is often made up of people with specialized skills. As such, it is common place for a few red team members to get lots of accesses; meanwhile, others can help with backdoors, or using that access to gain further access - rather than sit around idle. Tools like Armitage [8], a tool that can be used as a front end to Metasploit [9], have

collaboration support for this type of red teaming. The collaboration server within Armitage allows for agents to callback to a centralized location and anyone on the red team can then use them. One of the really valuable features allows red team members to use that shared access to proxy to other hosts internal to that network. These techniques are used to penetrate NAT and firewalls.

### 7.2 Communication

Communication tools were also very important. VPN users need to communicate with the on-site red team members; similarly, members need to know what others are doing in order to ensure that no two red team attacks affect one another. The two major communications tools that can be used are IRC and Skype. Both of these tools have different benefits and problems.

Skype [10] is a great tool for fast, real-time communications that can be left up in the background. In order to utilize the group video chat, paid add-ons are necessary. Additionally, in some cases in 2011, there were some issues with people dropping off of Skype. However, the biggest problem with Skype is that people have to repeat themselves quite frequently if they are not always readily available. During successful attacks, people were unable to talk about other topics; people periodically took breaks. For these reasons, IRC was also utilized. IRC provides a text-based chat that can be reviewed if someone missed something. It provides a way to have private chats from one member to another. The biggest problem with IRC is that it requires room on the screen to be read and some of the red team members tend to work on small laptops making it a cumbersome task.

## 8 Red Team Core

The makeup of the red team varied from region to region and while this helps with innovation and creative thinking the objective of the regional events should be to prepare the winner for nationals as well as provide a educational experience for the students. Additionally as the competition has grown it has become necessary to have a larger red team and more collaboration. The red teams at the events need more standardization and more unity among the events.

### 8.1 Virtualizing the Red Team

While it is impossible and cost prohibitive to send a core red team to all the events it is possible to provide remote access in order to get a core set of red team that are very skilled. If a core set of red team members can participate in multiple regional events it will also provide consistency across the regions.

This year a virtual red team was tried at two of nine regions as a test pilot to see if it would be feasible due to latency and communications considerations. Using the

different university's bandwidth and openVPN [11] we were able to allow VPN access for up to 10 virtual red team members without issue. The VPN connection went through the contest NAT and VPN addresses were a part of the red teams network. VPN users stayed in communication with IRC and Skype and used cell phone communication as a last resort. Several backtrack machines were provided on site that the remote users were able to SSH to and use as needed.

The VPN users this year were invaluable and these individuals were VPned from all over the country without issue. At one of the regions there were scheduled to be 14 teams and only 6 red team members. Without a virtualized red team the students at that event would not have been given the same red team experience that students at other regions with very large up to 30 red team members existed.

## 9 Conclusion

The competitions are evolving and becoming more complex - a great initiative that must be allowed yet tamed as well. Standardization amongst the regionals will create a fair national competition, as far as competition length, network configuration, red team start time, and scope of virtualization, to name a few issues. Distant competitors may begin competing virtually via VPN, as the At-Large competition operates, allowing for the same educational experience for schools that may have resource-limitations restricting distant travel or funding for these beneficial extra-curricular events.

The students without a doubt have learned from past events. The tools and methods that the red team has used for the past few years are not working as well and like the real world the red teams' tactics will have to evolve. The concentration going forward will be on how to retain persistent access and stay hidden from the users; likely, more custom written software and more advanced techniques will be required. Clearly, the educational institutions that are competing in the competition have adapted their curriculum. The more advanced and cutting-edge the red team becomes, the more secure the defenders of our networks will become.

## 10 References

- [1] "Competition Overview," [http://nationalccdc.org/index.php?option=com\\_content&view=article&id=46&Itemid=27](http://nationalccdc.org/index.php?option=com_content&view=article&id=46&Itemid=27), last accessed May 2, 2011.
- [2] Richardson, Mark. "The Weakest Link : Social Engineering," [http://www.internetviz-newsletters.com/shavlik/e\\_article000204422.cfm?x=a2rpFwQ.a1k9PlsG](http://www.internetviz-newsletters.com/shavlik/e_article000204422.cfm?x=a2rpFwQ.a1k9PlsG), last accessed May 3, 2011.
- [3] NMAP, <http://www.nmap.org>, last accessed May 2, 2011.
- [4] "About the Metasploit Meterpreter," [http://www.offensive-security.com/metasploit-](http://www.offensive-security.com/metasploit-unleashed/Metasploit_About_Meterpreter)

[unleashed/Metasploit\\_About\\_Meterpreter](http://www.offensive-security.com/metasploit-unleashed/Metasploit_About_Meterpreter), last accessed May 2, 2011.

[5] Core Impact, <http://www.coresecurity.com/>, last accessed May 2, 2011.

[6] "Little Reverse Shell Guide," <http://www.plenz.com/reverseshell>, Last Accessed, May 2, 2011.

[7] "Understanding Bash Fork Bombs," <http://www.cyberciti.biz/faq/understanding-bash-fork-bomb/>, last accessed May 3, 2011

[8] Armitage, <http://fastandeasyhacking.com>, last accessed May 2, 2011.

[9] Metasploit Pro, <http://www.rapid7.com/products/metasploit-pro.jsp>, last accessed May 2, 2011.

[10] Skype, <http://www.skype.com/intl/en-us/home>, last accessed May 3, 2011

[11] OpenVPN, <http://openvpn.net/>, last accessed May 3, 2011

# Blending Bloom's Taxonomy and Serious Game Design

L. Buchanan<sup>1</sup>, F. Wolanczyk<sup>1</sup>, and F. Zinghini<sup>1</sup>

<sup>1</sup>Secure Decisions Division, Applied Visions, Northport, NY, USA

**Abstract** - *Using serious games and interactive exercises can provide a safe and effective practice environment for computer network defenders, but development of these games must blend subject matter content, instructional design learning objectives and engaging game design to encourage learners to practice and develop their skills. As part of a program to develop an interactive training platform for the next generation computer network defender, we developed several Flash-based, casual games designed to target different levels of learning objectives as defined by Bloom's Taxonomy, for various skills, subject matter knowledge and tools. This paper lays out a working hypothesis based on that experience: some types of games are actually better suited to certain learning objectives.*

**Keywords:** cybersecurity education, security, Bloom's Taxonomy, learning objectives, serious games

## 1 Introduction

The need for skilled computer network defenders is rapidly growing, both in the commercial sector and in government. Training the next generation of computer network defenders who understand both the tools and the processes of Information Security and Information Assurance (IA) is a challenge being addressed in many different ways.

In most branches of the US Department of Defense (DoD), military personnel with little or no knowledge of computer security or even computer networks rotate into a duty position on a watch floor that handles incident response activities. Personnel generally spend a few weeks or months in a basic IA boot camp designed to teach the very basics of network defense, and they spend the next year learning to do the job with the tools, techniques and procedures used by that service. By the time personnel start to develop relevant skills and knowledge, they are ready to rotate out to the next duty station. Like most enterprises, the DoD needs to train its personnel faster, more effectively and in a more cost efficient manner.

SimBLEND<sup>1</sup> is a research program to develop a platform to assist in training the next generation of computer

network defenders by combining traditional computer based training ("CBT") with visually-intense training aids like serious games and exercises. Much of the entry-level subject matter for computer network defenders, such as ports and services and IP networking, is relatively dull and can be difficult to master. By providing an interesting, engaging and interactive opportunity to immediately review and practice the material covered in a CBT, learners are encouraged to improve their knowledge and skills. SimBLEND uses a traditional learning management system (LMS) to deliver both the CBT and the games, and supplements the LMS with an integrated performance analyzer that evaluates recorded metrics of learner performance in each game or exercise. These metrics are combined with grades from the traditional CBT material such as quizzes to determine an overall grade for the learner that is recorded by the LMS.

## 2 Interactive Cyber Security Training

To understand the issues with training entry-level computer network defenders in the DoD environment, we visited both the Vermont Air National Guard, Information Operations and the United States Air Force 39<sup>th</sup> IO Squadron at Hurlburt Field in Florida to observe the training for entry-level computer network defenders provided by live instructors at those schoolhouses. The courses begin with fundamental networking concepts, such as ports and services, and IP addressing and subnetting, then covers baseline tools to enumerate a network such as *ping* and *dig*, and culminates in network enumeration and vulnerability scanning tools such as *nmap*.

Based on our observation of these different environments, we developed a sample entry-level curriculum. We did not want to address issues of strategy or managing cyber security processes, but focused on hands-on skills. Tools to be covered in the individual classes of this sample curriculum included *whois*, *nslookup*, *dig*, and various DNS and network enumeration and vulnerability scanning tools such as *nmap*. Individual classes were intentionally scoped to cover smaller, very focused topic areas, providing "just in time" training. We developed a demonstration scenario using this curriculum that would highlight the progression of an entry-level computer network defender just starting with basic knowledge acquisition. We considered how to use games to scaffold the learner's progression through low-level network tools that require applying that knowledge, then moving on to active problem solving that would allow the learner to

---

<sup>1</sup> SimBLEND was developed under Air Force Research Laboratory (AFRL) Phase II SBIR contract FA8650-08-C6858. SBIR Data Rights (DFARS 252.227-7018 (June 1995)) apply.

demonstrate mastery of a variety of basic tools and techniques.

## 2.1 Serious Games for Cyber Security

Having identified the specific types of subject matter to be addressed, we began researching available games and interactive exercises for these low-level cyber security concepts and tools, in particular, web-based games that could be delivered from within an LMS and that allow metrics to be integrated into the automated sequencing and scoring. We also planned to demonstrate SimBLEND with a game that would function as a capstone exercise for the curriculum, a hand-on “final exam,” requiring the learner to draw upon the various concepts and tools covered during the classes.

We discovered that in 2009, there were very few serious games publicly available for cyber security training, particularly games that focus on specific, hands-on skill acquisition. Games such as CyberCIEGE [1] and CyberOps [2] are strategy-based games that focus on higher-level best practices and procedures for managing cyber security, not the use of actual tools: the player needs to know to buy a firewall for general protection against threats, but does not need to know how to actually set up a specific firewall to defend against specific threats. In addition, none of the serious games that we found were browser-based, they all required a client-side install, which complicated integration of the game with the LMS that was to deliver both the traditional course material and serious game as part of the CBT.

As we considered the nature of game play and interaction in the service of learning about cyber security, we realized we did not want a web lab, or a graphical simulation of a tool's interface that taught the tool interface. We needed interesting games that would help learners with the core concepts involved with using the tool: how to achieve the best results with a given tool in a specific situation, or even knowing when to use a specific tool. In addition, we wanted games that would create excitement in the learners while using the core subject matter concepts at the heart of the games.

It quickly became evident that not every game type is suited to every learning objective: 3-D games or simulations with avatars are just not well suited to basic knowledge acquisition, which was where we began in the process of cyber security game development. Consider CyberCiege, which is a 3-D game using storytelling of activity in an office environment; the learner makes strategic choices to demonstrate comprehension of best practice and higher-level concepts. Although 3D games represent state-of-the-art technology, using this type of game to assist in memorizing ports and services was too complex, and not very fun for the learner. Using 3D did not seem to actually enhance any of the basic game concepts we developed, which led us to consider a wider variety of game types.

### 2.1.1 Casual Games

We needed appropriate games to demonstrate the larger effort of the SimBLEND integrated training platform, so we leveraged our experience in developing training games for domains [3], and worked with an outside game studio to create our own serious games for demonstrating SimBLEND. We decided to create a series of short, Flash-based games, known as “casual games” in the professional game world. Casual games are ideal for this purpose, as they have simple rules and are easy to use. The player does not have to worry about learning basic controls and game mechanics; the learner would not have to be a “gamer” to succeed at casual games. Casual games are also typically short in duration, removing the time commitment required by more complex games.

Serious games require some effort to develop, even casual serious games. Defining the subject matter, how the knowledge or skill is used, what a tool's interface looks like, when is the tool useful, how individual commands are used, what can go wrong – these all need to be understood to develop an effective serious game. In addition, an entertaining concept is needed that makes the player *want* to play the game: an engaging storyline or interaction device [3]. In the course of developing the game interaction and progression, the instructor or subject matter expert can discover that the game essentially teaches the wrong behavior, or that the game behaves differently than the tool or the progression of a real-world scenario. At the start of designing a serious game, however, the instructor and the game designer both need to understand the core purpose of a game in a specific learning process: is it to help a learner memorize something as a part of basic knowledge acquisition, or should the game help the learner understand how to interpret results from a tool?

We began by identifying what cyber security concepts and subject matter would be used in each CBT and accompanying game, as well as the learning objectives for each class. Having selected different CND concepts, we looked for a game concept that would be something fun to do when the subject content is boring, while still maintaining some connection to the subject matter and would not get the user stuck trying to figure out the game mechanics. Memorizing ports and services is really boring, but a puzzle game where the learner must match items that belong together seemed like a natural fit for this memorization process.

## 3 Hypothesis

This led us to the hypothesis that different types of games are actually better suited for certain kinds of learning objectives, based on the type of interaction inherent in the game type. For example, in first person shooter games, the player must decide what to shoot, and in some instances, which weapon to use. This requires the player to understand the environment and tools available. Puzzle games often have an interaction that requires matching of concepts and the ability to recall information. From this understanding of the

interaction inherent in each type of game, it may be possible to determine what type of learning objective may be most readily served by a particular type of game.

### 3.1 Bloom's Taxonomy

After developing our idea of the learner's progression from low-level knowledge acquisition through understanding how a tool works and then to analysis of when to use each tool, we discovered that this matched existing pedagogical constructs and there was language to talk about this idea. Bloom's Taxonomy<sup>2</sup> is a classification of different levels of cognitive learning objectives that educators set for students. These "learning objectives" in the Taxonomy describe six progressive levels of learning, from the foundation to the pinnacle: knowledge, comprehension, application, analysis, synthesis, and evaluation.

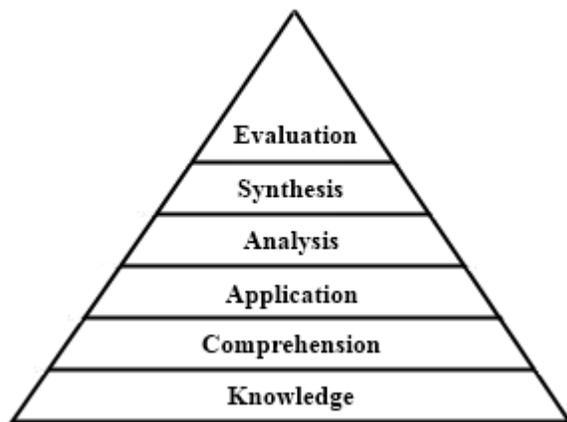


Figure 1. Learning Objectives from Bloom's Taxonomy

Subject matter experts and instructors may understand what the learning objective is for a particular lesson or course, but interaction and game designers generally do not. Having a clear concept of the specific learning objective desired for the game can assist the game designer during initial concept development. Our continued development of demonstration games for SimBLEND was informed by this concept of learning objectives and Bloom's Taxonomy.

### 3.2 Serious Game Classification

Games are classified by genre or game mechanic, such as first-person shooter, adventure, sports game, driving games, horror, puzzles, and simulations [4, 5]. Different types of game taxonomies exist for serious games such as those based on genre or game mechanics, or the presentation

platform or educational purpose [6, 7], but none of the taxonomies we identified address the issue of game design and instructional goals, learning objectives or Bloom's taxonomy.

Serious games design is a relatively new subject for academic research. Much of the existing literature about developing serious games focuses on elements of game design and how to make learning fun, such as interaction development and escalation, like the MDA framework cited earlier. When we began our project, the literature of serious game design had little focus on integrating pedagogical concepts such as Bloom's Taxonomy or learning objectives. Clark [8] mentions pedagogical elements, but refers to elements that appear in the simulation as scaffolding to help the learner, not as part of the underlying design question about what cognitive learning objectives the game needs to teach in addition to the specific subject matter.

The need for a formal design approach that brings together game design and instructional design has been articulated, and frameworks proposed [9, 10]. In these frameworks, integrating learning objectives as part of the early game design consideration ranges from a mere suggestion to a core principle of the framework's design approach. The frameworks do not address game taxonomies, nor do they hint at a potential connection between the type of interaction inherent in a specific game type or genre and learning objectives.

Bloom's Taxonomy did not originally incorporate the psychomotor domain, although this has since been addressed by others. For games designed to increase skills, psychomotor objectives may be very relevant when paired with Bloom's original cognitive learning objectives. A review of the key words describing the various objective levels in the psychomotor domain suggests that there may be a way to connect game interaction and learning objective level. For example, the initial psychomotor learning objective, *perception*, uses keywords such as: chooses, describes, detects, differentiates, distinguishes, identifies, isolates, relates, selects.<sup>3</sup> This list could also read as a set of game mechanic operations.

### 3.3 Cyber Security Learning Objectives

We did not address every level of Bloom's Taxonomy in our game development, but we have explored the range of objectives, considering what type of game interaction would support the basic learning objectives for a given set of skills and tools. The sections below describe how each layer of

<sup>2</sup>B. S. Bloom (Ed.). *Taxonomy of Educational Objectives: The Classification of Educational Goals*; pp. 201–207. Susan Fauer Company, Inc. 1956.

<sup>3</sup>E. J. Simpson. *The Classification of Educational Objectives in the Psychomotor Domain*. Washington, DC: Gryphon House. 1972. As quoted at <http://www.nwlink.com/~donclark/hrd/bloom.html>.

Bloom's Taxonomy was blended with game interaction during the design process.

### 3.3.1 Knowledge

The initial level of Bloom's Taxonomy is the acquisition of basic, foundational *knowledge*. In this stage, the learner should be able to remember ideas and information.

One of the fundamentals of networking and network defense is a thorough knowledge of ports and services. Learning even the most common Well-Known Ports which range from 0 to 1024 and the service associated with each port, is a dull exercise, requiring a lot of rote memorization of pairs. To assist in this learning objective, we designed a matching *puzzle game* around the idea of "connecting the dots." Learners must manipulate tiles to connect a pathway between a numbered port and the corresponding service. The basic manipulation is extremely simple – the focus for the learner is which port and service belong together. The game is timed, and points are acquired for each successful connection. Each progressive level of the game adds more pairs that need to be connected, and while the most common ports are used in the beginning level of the game, less common ports are introduced at more advanced levels of the game.

As a Flash-based game, the ports and services pair data is stored in an XML file, which makes it quite simple to create customized variations of the game. An advanced version adds Registered Ports (1024 – 49151) into the mix, extending the learner's familiarity with a wider range of ports and services. Another version uses common business applications and ports that need to be open or closed in corporate firewalls, and an "evil genius" version includes common malware ports, assisting the learner in learning what activity or configuration issues to look for inside the firewalls.

### 3.3.2 Comprehension

This level moves beyond surface understanding; a learner should be able to interpret, discuss, compare concepts in terms of similarities or differences, or explain the subject in their own words.

To assist in the comprehension of the network scanning tool *nmap*, we developed a game that is very loose variation of the *first person shooter* type, using missiles rather than guns. In first person shooter (FPS) games, the player must try to take down various targets, typically by shooting a gun; variations on FPS games may provide different weapon types that yield a variety of results and scores.

Nmap has a wide variety of command line switches that are used to control the scanning parameter, and it is critical that the operation of the switches is correctly understood as a malformed scan could cause a network disruption, potentially

performing an unintended denial of service attack. By using the FPS concept and making the computers and networks the target, the learner can explore the different command line switches in *nmap* and comprehend how the different switches work (or don't work) with various computer operating systems. The goal of the game is to stop the missiles from falling on U.S. soil; if a missile hits the U.S., the game is over. To stop a missile, the learner must successfully scan (shoot) the computer layers of the missile by selecting the optimal command line switch from a range of choices, however, the learner does not know in advance the operating system or any other characteristics of the computer that is being scanned. The game is timed and different switches cause the scan to run faster or slower as they would in real life. Once the scan is complete, the game displays the *nmap* output formatted just like the command line tool. The player must also be able to correctly understand the scan results in order to answer questions related to the tool or the computer that was scanned.

A similar FPS concept could be used for any subject matter where the learner needs to understand the action and subsequent reaction and/or consequence. For cyber security, another obvious candidate for this concept would be "shooting down" intrusion detection alerts that are not false positives but represent actual attacks and for the game network.

### 3.3.3 Application

At this intermediate level in the Taxonomy, the learner should be able to *apply* a concept, solving a problem by using or examining knowledge and understanding in some manner.

The definition of this learning objective reveals a promising game type: puzzles and problem solving. In the *nmap* missile game described above, the learner must apply their knowledge of ports and services to answer questions about the scan results, such as identifying the computer's OS or function based on ports reported open by the *nmap* tool. The learner must also apply their knowledge of the *nmap* switches to determine what available command line switch options would help "avoid detection" by generating less activity on the network and test different strategies for using some of the most common *nmap* switches under timed circumstances.

### 3.3.4 Analysis

The learner can *analyze* the topic or material, and both distinguish between the parts and make connections at this level of the Taxonomy.

Our capstone exercise focused on the higher cognitive levels of Bloom's Taxonomy. The goal of the game is to ensure that the network supporting a logistics convoy is secure

from attack while the convoy travels to its destination. Through the Flash game, the learner gets a feel for what it might be like to sit on the watch floor of a network or security operations center and evaluate alerts that may represent actual attacks. Provided with information about the network to be defended (network topology, device roles, typical activity for devices, relevant firewall rules) the learner must decide if each new alert is a false positive or real potential attack, and take prompt action. The game interface is divided into several areas: a view of the network devices and their current activity levels; a window with incoming alerts; a view of the convoy and its progress over its route, and a window that simulates a window for command line activity. The learner must use information from these multiple sources, and from tools such as *whois* and *nmap* in the simulated command line window to analyze the potential vulnerability of a device for the attack indicated by the incoming alert.

This simulation game uses the narrative devices of supporting the convoy mission and a defined network to get the learner involved and set the stage for further activity. While scenarios were a part of the other two games, they acted as bookends at the beginning and end of the game. The scenario of defending the convoy network is at the heart of this game. It also cultivates in the learner an understanding that computer network defense is not an abstract thing, but that both business operations and physical (kinetic) missions depend on the network. Using a simulated command line interface restricts the learner to a narrow set of options, which greatly reduces the effort to build the game.

### 3.3.5 Synthesis

Similar to *application*, but on a more sophisticated level, *synthesis* requires a complete understanding of a topic. The learner is able to explain their rationale for choices.

We believe that narrative genre games, particularly the “choose your own adventure” variations, are well suited to the synthesis learning objective. We extended the learning objectives for the capstone exercise described above to address synthesis as well as analysis. After determining if the alert represents a viable attack or a false positive, the learner must also explain the rationale for their decision and indicate why each alert was dismissed or was sent on to a handler for further investigation. To assist learners in achieving this learning objective, in this game a false positive incorrectly determined to be a viable attack is acceptable with no penalty, but true attacks incorrectly identified as false positives incur penalties that increase at an exponential rate.

Another game concept we explored but did not develop involved having the learner determine the priority for patching specific systems in a defined network. This would have involved synthesis of vulnerability information, vulnerability scan tool results, and information about other security settings in the network.

### 3.3.6 Evaluate

At the pinnacle of Bloom’s Taxonomy, learners are able to *evaluate* and put together disparate elements to create something new that is a coherent or functional whole.

We considered an extension to our network defense capstone exercise that would address this final learning objective. After successful completion of the capstone exercise, learners would be asked to develop their own set of alerts for the network environment used in the game. The data set would need to include both false positive alerts and true alerts, and learners would need to provide data supporting the evaluation of the alert as false or true. The data could then be loaded into the game and the learner could play against their own data set, testing the accuracy of their data.

## 4 Conclusions

An interesting discovery was that, unlike simulations, casual games allow the tool interfaces to be abstracted away. They allow the novice learner to focus on the general concepts and skills of the subject matter, and not become distracted by a tool’s user interface. This may be useful in areas where there are multiple tools that are based on the same core concept and perform the same function, such as vulnerability scanning tools.

Ideas for future research in this area include development of a different kind of serious game taxonomy, one that specifies which game types are well suited to deliver individual learning objectives at a particular level of Bloom’s Taxonomy. We have already begun this work as part of a modification to the SimBLEND project. We are developing *ShortCut*, a tool intended to streamline the creation of visually-intensive training aids such as serious games by facilitating collaboration between subject matter experts and interaction or game designers, using an interactive, web-based knowledge elicitation. The knowledge elicitation in *ShortCut* is designed to allow the instructor or subject matter expert to describe not just the subject matter, but with special consideration for information that would be needed by interaction and game designers. We have begun to collect meta data on other existing cyber security games and will identify the intended learning objective types and correlate them with the type of game or game mechanic used to further extend our taxonomy and evaluate our hypothesis.

There can be serious game design considerations that go beyond the learning objectives of Bloom’s Taxonomy. For example, speed may be important to the learning objective and subject matter. It may be critical that the learner gain immediate recall of the knowledge or be able to perform the task very quickly, such as identifying the type of activity that is indicated by port numbers recorded in log data from firewalls or intrusion detection systems. In other cases, correctness and accuracy in the learner’s understanding of the skill or tool may be more important than speed. This was true

for the nmap missile game: correctness and accuracy of the switches and flags in the command line string are more important than speed, because an incorrect switch may adversely affect the tool's operation. As a result, while the game is timed, selecting the switch that yields the most information under the circumstances is the most important factor. The game allows enough time to select another switch, but the player does not receive as many points as making the correct selection first. We have incorporated these additional serious game design considerations into the knowledge elicitation used in our ShortCut tool. The goal is to provide the subject matter expert with the ability to describe the fullest complement of game and interaction characteristics.

It would be desirable to test the effectiveness of different game types over the same training material and using the same data set in the different game types. The results of this research could prove or alter our hypothesis, and assist in the further development of serious game design frameworks that blend subject matter content, instructional design and game design. We welcome a discussion of this hypothesis and hope that others become involved in working towards improving the quality of cyber security education at all levels through the use of serious games.

## 5 References

- [1] Cynthia E. Irvine, Michael Thompson and Ken Allen. "CyberCIEGE: An Extensible Tool for Information Assurance Education"; 9th Colloquium for Information Systems Security Education, 130-138, June 2005.
- [2] Brian Duffy. "Network Defense Training through CyberOps Network Simulations"; Modeling Simulation and Gaming Student Capstone Conference 2008, 2008.
- [3] Markus Lacay and Joe Casey: "Serious Games: Fun vs. Reality", SISO Spring SIW 2011 Conference, No. 11S-SIW-012, April 2011.
- [4] Wendy Despain. "Writing for Video Game Genres: From FPS to RPG". A K Peters, Ltd., 2009.
- [5] Craig Lindley. "Game Taxonomies: A High Level Framework for Game Analysis and Design"; October 3, 2003. [http://www.gamasutra.com/features/20031003/lindley\\_01.shtml](http://www.gamasutra.com/features/20031003/lindley_01.shtml)
- [6] Ben Sawyer and Peter Smith. "Serious Games Taxonomy"; Serious Games Initiatives, February 2008.
- [7] Clark Aldrich. "Learning Online with Games, Simulations and Virtual Worlds". Jossey-Bass, 2009.
- [8] Clark Aldrich. "The Complete Guide to Simulations & Serious Games". Pfeiffer, 2009.
- [9] Brian Winn. "The Design, Play, and Experience Framework"; Handbook of Research on Effective Electronic Gaming in Education (Information Science Reference), Volume III, July 2008.
- [10] G. Gunter, R. Kenny and E. Vick. "A case for a formal design paradigm for serious games"; The Journal of the International Digital Media and Arts Association, Volume 3, 93-105. 2006.

# Challenge Based Learning in Cybersecurity Education

Ronald S. Cheung (cheungr@cs.umb.edu), Joseph P. Cohen (joecohen@cs.umb.edu)

Henry Z. Lo (henryzlo@cs.umb.edu), Fabio Elia (fabioel@cs.umb.edu)

Department of Computer Science, University of Massachusetts, Boston, MA, USA

**Abstract**—*This paper describes the application of the Challenge Based Learning (CBL) methodology to cybersecurity education. The overall goal is to improve student learning via a multidisciplinary approach which encourages students to collaborate with their peers, ask questions, develop a deeper understanding of the subject and take actions in solving real-world challenges. In this study, students established essential questions which reflected their interests in information security, formulated challenges on how to safeguard confidential information from cyber attacks and then came up with solutions to secure their information and network. For guiding activities, students participated in two cybersecurity competitions against their peers from other local universities. In these simulated real-life competitions, students were forced to work together, think on their own two feet and apply their knowledge to defend against cyber attacks. Assessments performed after the study showed improvement in students' computer and security skills, interest in learning security and ability to teach others. Student learning was further reinforced with publication of their research findings and making presentations to their fellow classmates.*

**Keywords:** cybersecurity, education, challenge based learning, CBL

## 1. Introduction

It is well known that cyber threats to the United States are prevalent and they affect our society, business, and government, yet there is no concerted effort among our government and private industries to overcome them. In 2010, the former Director of the National Security Agency, Mike McConnell, testified in the Senate that if there were a cyber war breaking out against our nation's infrastructure, we would lose. He reiterated his grim assessment a year later that we are no better off, though the stakes have risen higher [1]. His concern is realized with recent cyber attacks emanating from servers in China on Google and several dozen U.S. companies. These attackers were able to penetrate the defense of company networks and attempted to steal email accounts, information on weapon systems, and intellectual property.

Top officials in the Defense Department have long believed that the reason why the country's cyber defense is not up to the challenge is due to a shortage of computer security specialists who can battle attackers from other countries.

The protection of U.S. computer systems requires an army of cyber warriors and the current estimate is that there are only 1000 workers skilled in this area. However, to meet the computer security needs of government agencies and large corporations, a force of 20,000 to 30,000 skilled specialists is needed [2]. In response to these heightened concerns, the Senate Commerce Committee recently approved the Cybersecurity Act (S.773) which recommends actions the government should take to improve the nation's cybersecurity preparedness. Among them, the government should fund research leading to the development of new security technologies, promote public awareness of cybersecurity issues, and encourage the growth of a trained and certified cybersecurity workforce [3].

Universities are slow to react to the need of cybersecurity education. It is very common for a computer science major to go through four or five years of undergraduate schooling without taking a single required class on security[4]. Consequently, they graduate without knowing anything about it. At the University of Massachusetts Boston, the Computer Science Department offers an ABET accredited curriculum which covers traditional courses in programming, compilers, operating systems and others. These courses tend to be theoretical and they do not deal with real-world problems in security. Recently, the department has added a more hands-on BS in IT program that offers a course in Network Security Administration. Since this is a new course, enrollment is limited. Furthermore, CS majors interested in cybersecurity often cannot take it because they lack the required IT prerequisites. The goal for this study is to apply innovative student learning methodologies to teach cybersecurity to a group of motivated CS/IT students who are interested in the topic.

## 2. Challenge Based Learning Methodology

Research has shown that student-centered learning approaches are efficacious in improving student learning [5]. In particular, the challenge based learning (CBL) methodology proposed by Apple Computer Inc., which employs a multidisciplinary approach in encouraging students to use their knowledge and technology to solve real-world problems, has reported to yield outstanding results [6]. The challenge approach works because most students are familiar with the concept since they have watched multiple reality TV shows

that are based on it. The common theme is that contestants are presented with a challenge that requires them to draw on prior learning, acquire new knowledge, work as a team, and use their creativity to arrive at solutions. Another reason why this concept is successful is that the participants are highly motivated by the common goal of potentially winning a big reward afterwards.

The challenge concept has been applied to the development of cybersecurity skills among high school and college students. One example is the U. S. Cyber Challenge sponsored by the Center for Strategic and International Studies (CSIS), the SANS Institute, the U.S. Department of Defense (DoD), universities and private industrial firms [7]. It is both a national cybersecurity talent search and skills development program. High school students compete on-line in the Cyber-Patriot Competition sponsored by the Air Force Association where they learn how to control computer networks, defend and protect computer systems from cyber threats and hackers. High school, college and graduate students participate in the DoD Cyber Crime Center (DC3) Digital Forensics Challenge and the NetWars competition. The DC3 Digital Forensic Challenge is an on-line event that tests students on individual scenario-based, investigative tools, techniques and methodologies. The competition fosters innovation among students and encourage them to provide technical solutions for computer forensic examiners in the lab and in the field. The NetWars is an interactive security challenge that tests students' security knowledge and capture the flag skills. Successful contestants in these competitions are immediately recognized and invited to attend regional security camps, national challenges, or given grants or scholarships to study cybersecurity.

Apple Computer Inc. has applied CBL to the collaboration project, Apple Classrooms of Tomorrow (ACOT), between public schools, universities, and research agencies with great success [8]. In this study, we adapt the CBL methodology to teach practical cybersecurity education to a team of nine CS/IT students with different backgrounds of computer education. Some members are sophomores and some are seniors. Most students have no prior formal training on cybersecurity. They enroll in this study in addition to taking their regular course load.

The CBL framework, as shown in Figure 1, is implemented in this study as follows:

## 2.1 Big Idea

The team considered the topic: Cybersecurity, which has broad meanings and importance to the students and society.

## 2.2 Essential Questions

The team came up with the following questions that reflected their interests and the needs of the community:

- What kind of information does one need to keep secure?  
The classification of information would dictate the

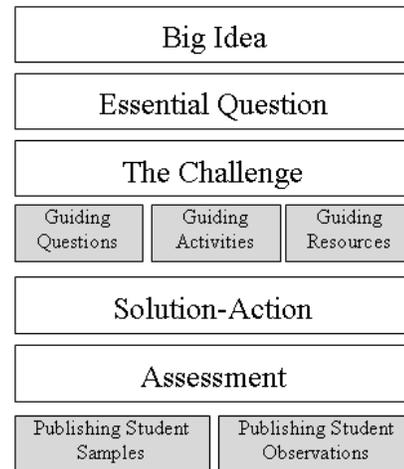


Fig. 1: The CBL Framework

security, management, use and disposition of these data. For those that have been classified as Confidential, such as Personally Identifiable Information (PII) and Protected Information, federal, state laws/regulations or organization rules may govern how they should be protected.

- What does one do to safe guard Confidential information?

Depending on whether the threat is internal or external to the organization, the methods of safeguarding information are different. In the case of internal threats, one may have to take precautions against social engineering tactics. In dealing with external threats, the most effective way is to secure the network and computer systems.

## 2.3 The Challenge

For each essential question, a challenge was formulated that asked the students to come up with a specific answer or solution. In our study, the students came up with:

- Keep confidential information safe
- Keep network safe from cyber attacks

## 2.4 Guiding Questions

Students generated questions that they would need to discover solutions for in order to meet the challenges. Some guiding questions were:

- What are social engineering tactics? How does one guard against them?
- For sensitive information such as administrator passwords, how can they be changed frequently without the excessive burden of remembering the changes?
- How does one know that the organization is being attacked?

- What are the techniques of configuring firewalls to secure the perimeter of the network?
- What techniques do attackers use to penetrate a network's defense?

## 2.5 Guiding Activities

The students held weekly discussions with the coach, learned network security techniques from our university IT security experts, attended seminars, practiced on their own time the installation of different computer operating systems and software applications, and practiced configuration of network services such as Domain Name Service (DNS), Network Information Services (NIS), mail server and firewall etc.

In order to gain practical knowledge, the students competed in the Northeast Collegiate Cyber Defense Competition (NECCDC) [9] and the MIT Lincoln Laboratory/CSAIL Capture the Flag competition (MITCTF) [10] against their peers from universities in the Northeast region. In these two competitions, students got a chance to practice what they had learned. They defended against cyber attacks as well as generated attacks onto others in a simulated live networking environment.

## 2.6 Guiding Resources

Students did their research using books, class lecture notes, papers, the Internet and expert opinions in developing solutions to their guiding questions. They watched videos on the Internet to learn how to fend off social engineering tactics. They purchased equipment, set up and maintained a small standalone network that allowed them to practice network security exercises.

## 2.7 Solutions

Students devised situation-specific solutions as well as general solutions during the two competitions. Critical aspects of computer system configurations such as firewalls, bound ports, logging, updating, and user accounts were itemized and worked on by the entire group. Students learned how to whitelist needed ports in the firewall, scrutinize event logs for possible break-ins, and update operating systems and applications.

The group debated on user account management and came up with a solution in protecting passwords from people stealing them via social engineering tactics. In general, the most effective method against them is to change passwords frequently. Unfortunately, this increases the burden of users having to remember many different ones. Some users resort to writing them down on a piece of paper or their notebooks, and these are easy targets for people to steal them using social engineering means. To alleviate this burden, students came up with a password selection card from which the password could be derived using a code sequence. This solution greatly reduced the chance of attackers stealing the

password because they had to pilfer the selection card, the code and the way to interpret the code in order to construct the exact password. At the NECCDC, this approach was openly recognised as a good idea.

## 3. Cybersecurity Competitions

The two major activities in which students applied the knowledge they had learned were the NECCDC and the MITCTF competitions. The NECCDC is an annual competition to train students on managing and protecting an existing network infrastructure from a group of unbiased "Red Team" attackers. These attackers comprised of Information Assurance (IA) professionals who were very experienced in computer security. In NECCDC 2011, eleven universities from the Northeast region competed. Each team was given an identically pre-configured computer network which simulated that of a working business. Teams earned points by maintaining the availability of services and integrity of the systems. Participants were not allowed to attack the networks of the "Red Team" or other student teams.

In preparation for the event, the University of Massachusetts Boston team set up a network of computers using the topology provided in the rules. Two learning groups were formed based on the students' expertise. One focused on maintaining services and the other on network/system security. During weekly meetings, methods to defend the system/network and install various services were researched and practiced. As the days for the competition approached, the team's focus shifted towards formulating a strategy, and created lists of tasks needed to be completed. Specific roles were assigned to each of the members and a hierarchical communication structure was established.

At the competition, each team was presented with an identical network of computers, switches and routers. Students were given instructions (or injects) by a member of the "White Team" acting as a liaison for the company. Examples of these injects included generating audit reports, setting up a network printer, installing new software and services, and updating existing packages. From the very beginning, the team was bombarded with an onslaught of attacks from the "Red Team". The need to simultaneously maintain business services and defend the network against attacks created a stressful, fast-paced learning environment. After a three-day struggle, the NECCDC competition ended with the University of Massachusetts Boston team placing in the last place.

After the competition, the team got together and did an assessment. The general consensus was that the team learned a lot from the experts on defending the network. This included utilities such as, netstat, ncat, lsof, operating system internals and others. Also, this competition pointed out knowledge we did not know; for example, securing the Cisco router and switch against attacks, knowing whether or not our systems were compromised, and how to setup a

spanning port to track all traffic on the network. The students realized that there were communication problems during the competition. As a result, a smaller core team was formed and it consisted of motivated students with higher networking knowledge. Meetings became more effective in exchanging ideas. The team was better focused and spent time on studying web applications, researching Linux vulnerabilities and properly configuring services.

The MITCTF competition, hosted by MIT Lincoln Laboratory and MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), was also focused on educating and increasing students' awareness on cybersecurity. In 2011, there were thirteen teams in the competition and the goal was to test students on their knowledge of cyber offensive and defensive techniques. They defended and attacked a plug-in based Content Management System (CMS) which simulated a working business website hosted on a local server. Before the competition, MITCTF ran training sessions on the CMS details, cyber defense basics, web-based attack vectors, and also provided a downloadable virtual machine image for students to practice on.

At the competition, each team was given a virtual machine image where flags, consisting of a string of random characters scattered throughout the file system, would rotate every five minutes. Opposing teams attempted to capture these flags and submit them for points. Grading was based on the availability of the sites, the number and integrity of flags captured. Throughout the competition, teams were required to install new plug-ins to the CMS. Each new plug-in introduced new vulnerabilities, requiring patches to be implemented and exploits to be developed on the spot. Failure to do so could potentially allow opposing teams gaining access into systems and wreak havoc.

Prior to the competition, using the virtual machine image provided by MITCTF, students studied the provided code and discovered system vulnerabilities and improper configurations. They also spent time on developing scripts to secure their own system and exploit others. During the competition, the team was able to break into other systems using these scripts and caused havoc to them. These scripts included SQL Injections, Cross Site Scripting, and a PHP vulnerability on a calculator plug-in allowing execution of arbitrary code. After two days of competition, the University of Massachusetts Boston team was placed second among all thirteen teams.

#### 4. Student Learning Results

Research has shown that group dynamics plays a crucial role in student learning [11]. In our CBL study, the group formed had various levels of technical expertise. Students were encouraged to participate in open discussions and work together in smaller groups based on their expertise. Meetings served as a conduit for students to research topics, voice their questions and opinions on topics which they had no

prior knowledge. This was an important aspect in improving student learning.

After performing assessments, outcomes of our study are summarized as follows:

In Figure 2, most students reported that their computer skills increased at the end of the study. For example, one student started training without a strong background in Linux and is now proficient enough to teach other students on how to configure Linux. This perceived increase in computer skills can be interpreted in two ways: they acquired new knowledge, or applied what they already knew in different ways.

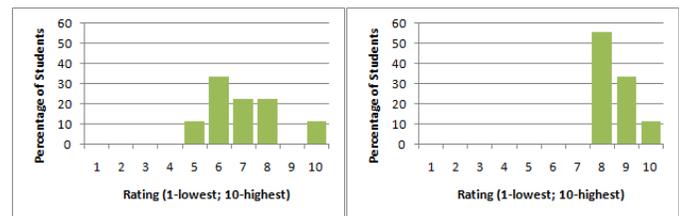


Fig. 2: Student's self-reported computer skills, before (left) and after (right) the study.

There was an improvement in perceived computer security skills as shown in Figure 3. Very few students prior to the study knew anything about computer security. Afterwards, they all seemed to have understood what the field of security involved and felt that they had gained knowledge in this area. The improvement in skills could be influenced by the frequent interaction with students that had high technical expertise, as well as industry experts, such as the "Red" and "White Team" members during the competitions.

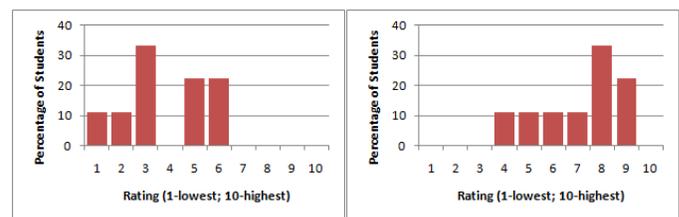


Fig. 3: Student's self-reported security skills, before (left) and after (right) the study.

There was a sharp increase of student interest in computer security after the study as depicted in Figure 4. This may be due to the frequent meetings where students learned from each other. Another reason is that students saw how their knowledge was applied in a real world environment. Several students from the team, after the study, formed a new student group with the purpose of spreading security knowledge and interest among other fellow students in the CS department.

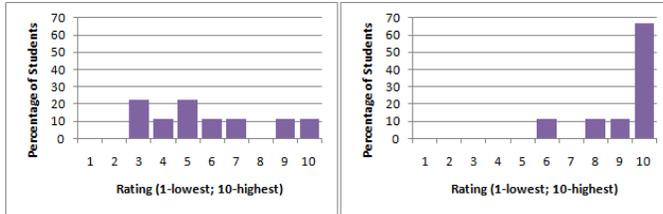


Fig. 4: Student's self-reported interest in computer security, before (left) and after (right).

Figure 5 shows that about half of the students, after going through the study, felt they could teach computer security and half could not. Although all students gained computer and security skills, some still did not feel comfortable enough to teach others.

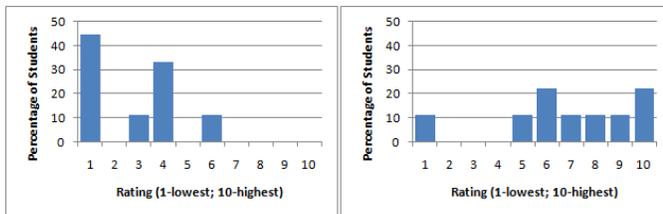


Fig. 5: Student's self-reported ability to teach others security topics, before (left) and after (right).

Although the students came with a range of technical abilities and initial interest, Figure 6 shows that all students who participated in the study benefited greatly from the CBL experience. These benefits included knowledge gained by networking with industry professionals, improving computer and security skills, and applying these skills in a practical, real world environment.

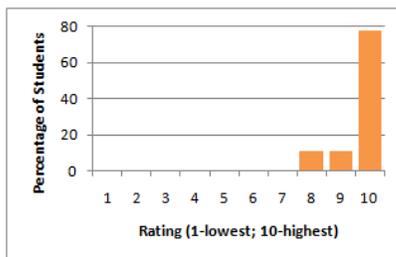


Fig. 6: Student's self-reported benefit of the study.

## 5. Student Observations

Our study is based on one student group comprising of students with different skill levels. Though the sample size is small, we believe the CBL methodology is beneficial to teaching cybersecurity education. Our student group exhibits the six team basics required for high team performance [12].

In this study, we formed a small group of nine students. Group members had complementary skills in computer programming and course knowledge. Some students had working Windows and Linux knowledge, while others had none. The team shared a common purpose of increasing their computer security knowledge. Team members knew they had to achieve a common set of specific performance goals. In this study, these performance goals were specified by the competition organizers. For both NECCDC and MITCTF, students were aware of the services they needed to install and maintain. Throughout the study, students agreed on a common working approach. This included meeting and discussion in a democratic manner. Students with technical expertise tended to guide in this area by suggesting topics to research and troubleshoot configurations. Students voted to decide which direction to take. For example, if no students were familiar with a piece of software application or tool, then the group jointly determined the best course of action to take. All group members felt they were mutually accountable for the success of the team. In our case, the group was highly critical of each others' performance. Students divided up technical functions such as email configuration, central authentication, and DNS; they were then held responsible for that function. If a student did not master the configuration and security for that function, other students would remind him that he needed to do so. Each student expected other students to become experts in some technical area or some configuration of a particular system after they had studied it.

In our study, students found the guiding activities in participating in the NECCDC, and MITCTF competitions most beneficial. The competition provided a real-world cyber defense situation that our students practiced their knowledge on. Students were forced to work in an intense atmosphere that they had to band together to work as a team in solving a problem. Each team member contributed to the solution based on his individual training. Also, what students found most stimulating were discussions with security professionals from industry afterwards and learning from them techniques on securing the network. They also appreciated the opportunity to network with company recruiters and students from other universities in sharing their experience.

Though the CBL methodology seems to improve overall student learning, its benefits vary from one student to another based on their interest and motivation. As compared to that of conventional teaching methodologies, CBL student learning depends more heavily on self-study and peer instruction efforts. Those who are not sufficiently motivated to learn new concepts or technologies on their own have less to gain. Furthermore, those who have less interest tend not to show up at the meetings as often. Since these activities are mostly student-organized, there is no penalty for not showing up except for the fact that these students will learn less. Also, the presence of indifferent and unmotivated individuals

hinders the progress of the group.

The loose structure of the teams in CBL, though beneficial to some team members, may not work for others. The lack of an instructor-student hierarchical structure may not give enough direction for some students to follow. As a result, they lose the motivation of attending meetings. Student ability is also an important factor in learning a highly technical subject such as cybersecurity. For example, the students need to have basic knowledge in networking and scripting in order to configure the firewall to fend off attackers. These are not only key security concepts, they are also vital skills for system administration. Without them, they cannot learn how to secure a system in a short time. Consequently, more experienced students have to spend time teaching the less knowledgeable ones. This slows down the team's progress and reduces their overall learning. These observations suggest that the CBL methodology, especially on teaching a highly technical subject like cybersecurity, can achieve a better outcome if all team members start with the basic prerequisite knowledge.

Although our CBL study does not need much instructor intervention, it requires additional resources, such as equipment and support staff to support the activities of the group. For example, to build a practice network, the students need dedicated computers, routers and switches. This equipment is often deployed at irregular times and their malfunction requires off-hours support. Also, because this study deals with computer security, students have to negotiate firewall policy with the university's IT department so that the practice network will not be blocked from the Internet. These difficulties highlight the importance of additional support resources and their flexibility in support hours that are needed to effectively apply CBL to cybersecurity education. However, we believe that the potential gain in student learning justifies the extra effort to overcome these obstacles.

## 6. Conclusions

This paper has described the application of the Challenge Based Learning methodology to cybersecurity education. By formulating challenges based on students' interest in securing information and systems, students worked together as a team on devising solutions to meet the challenges. Students in this study practiced what they had learned in two cybersecurity competitions. Formative assessments performed showed that students benefited greatly from the CBL approach, though the amount of benefit varied from one student to another. The students were able to improve their computer skills, security knowledge, ability to teach others and interest on the topic of cybersecurity. Though the approach may require additional support resources and may require meetings at irregular hours, the increase in student learning justifies the extra effort.

## 7. Acknowledgments

This research was supported by the Spring 2011 Program on Instructional Innovation (Pi2) grant, the College of Science and Mathematics, University of Massachusetts Boston. The authors would also like to thank the UMB-NECCDC team members for their participation in the research.

## References

- [1] M. McConnell, "To win the cyber war we have to reinforce the cloud", *Financial Times*, April 24, 2011. [Online]. Available: <http://www.ft.com/cms/s/0/078bf734-6e9b-11e0-a13b-00144feabdc0.html>
- [2] K. Evans and F. Matters, "A Human Capital Crisis in Cybersecurity", *Center for Strategic and International Studies (CSIS)*, Nov. 15, 2010. [Online]. Available: [http://csis.org/files/publication/101111\\_Evans\\_HumanCapital\\_Web.pdf](http://csis.org/files/publication/101111_Evans_HumanCapital_Web.pdf)
- [3] J. Vijayan, "Cybersecurity bill passes first hurdle", *Computerworld*, March 24, 2010. [Online]. Available: [http://www.computerworld.com/s/article/9174065/Cybersecurity\\_bill\\_passes\\_first\\_hurdle/](http://www.computerworld.com/s/article/9174065/Cybersecurity_bill_passes_first_hurdle/)
- [4] (2011) "Security Lessons still lacking for Computer Science grads", *InfoWorld*. [Online]. Available: <http://www.infoworld.com/t/application-security/security-lessons-still-lacking-computer-science-grads-769>
- [5] G. O'Neill, T. McMahon "Student-Centered Learning: What does it mean for students and lecturers", *University of College Dublin*. [Online]. Available: [http://www.aishe.org/readings/2005-1/oneill-mcmahon-Tues\\_19th\\_Oct\\_SCL.html](http://www.aishe.org/readings/2005-1/oneill-mcmahon-Tues_19th_Oct_SCL.html)
- [6] L.F. Johnson, R.S. Smith, J.T. Smythe, R.K. Varon "Challenge-Based Learning: An Approach for Our Time", *The New Media Consortium*, Austin, Texas. [Online]. Available: <http://ali.apple.com/cbl/global/files/Challenge-Based%20Learning%20-%20An%20Approach%20for%20Our%20Time.pdf>
- [7] (2011) "U. S. Cyber Challenge", *Cybersecurity Workforce Development Division, Center for Internet Security*. [Online]. Available: <http://workforce.cisecurity.org/>
- [8] (2011) "Challenge Based Learning- Take action and make a difference", *Apple Computer Inc.*. [Online]. Available: [http://ali.apple.com/cbl/global/files/CBL\\_Paper.pdf](http://ali.apple.com/cbl/global/files/CBL_Paper.pdf)
- [9] (2011) *Northeast Collegiate Cyber Defense Competition (NECCDC)* on March 4-6, 2011 in EMC Corporation, Franklin, MA. [Online]. Available: <http://www.ccs.neu.edu/neccdc2011/index.html>
- [10] (2011) *MIT Lincoln Laboratory/CSAIL Capture the Flag Competition* on April 2-3, 2011 in MIT, Cambridge, MA. [Online]. Available: <http://mitcf2011.wikispaces.com/>
- [11] D.R. Forsyth, *Group Dynamics*, 5th ed., Wadsworth Publishing, 2009.
- [12] J.R. Katzenbach, D.K. Smith. "The Wisdom of Teams", New York, NY: HarperCollins, 2003.

# The Assembly and Provisioning of a Red Team

Daryl G. Johnson

Networking, Security and Systems Administration Department  
Rochester Institute of Technology  
Rochester, NY USA

**Abstract** – *As the value and merit of red team exercises in both academic and corporate settings continues to grow, the need to share experiences with staffing, organizing and supporting the red team becomes increasingly important. This paper documents the Northeast Collegiate Cyber Defense Competition's (NECCDC) Red Team captain's experiences and lessons learned over the past four years. The paper will begin by identifying the skills and attributes needed for a Red Team and a process for selecting and recruiting members. The methods employed to form a cohesive working group from the members in the time available prior to the event will be discussed. The resources necessary for the Red Team to be effective and how they were provided is examined. We will look at how to promote planning and organization within the team focused on specific strategic goals and objectives of the Red Team. There are several duties during the event for a Red Team captain that will be examined and cautions that will be explained. At the end of the competition, the style and delivery of the after-action-report can have a profound effect on the Blue Teams. Experience with different approaches over the years will be examined. Recommendations for Red Team/Blue Team exchanges that can maximize the learning outcome for the students will be provided. Finally this paper will provide a summary of the experiences for others seeking to form and organize a Red Team either for a competition or an internal educational event.*

**Keywords:** red team, security, education.

## 1 Introduction

The threats to an organization's information infrastructure today have never been greater as illustrated by the FBI/CSI Computer Crime Survey. From the often quoted Sun Tzu we have "If you know the enemy and know yourself you need not fear the results of a hundred battles." [1] Professor Pascale Carayon describes Red Teaming as "an advanced form of assessment that can be used to identify weaknesses in the security of a variety of systems. The red team approach is based on the premise that an analyst who attempts to model an adversary can find systemic vulnerabilities in a computer and information system that would otherwise go undetected." [2] As

the value and merit of Red Team exercises in both academic and commercial settings continues to grow the need to share experiences with staffing, organizing and supporting the red team become vitally important. This paper documents the Northeast Collegiate Cyber Defense Competitions (NCCDC) Red Team captain's experiences over the past four years. The many issues influencing the selection of skills and capabilities, the organization and planning, and the execution of the event from the Red Team perspective will be examined.

## 2 Assembling the Team

The selection of the individuals for any activity requiring highly skilled members is critical to their combined success. Red Teams are especially sensitive because of the high degree of specialized skills and the pressure of the competition itself.

There are three characteristics that we looked for in recruiting Red Team members. First, *passion* for the security field is the best motivator when a difficult situation or road block presents itself. It pushes the individual to perform above and beyond their limits.

Second, *skill, preparation and dependability*: can they do the job? Are they willing to work long hours for no pay? Can they deliver what they promise? The best indicator I have found is references, recommendations and experience and I rely on all three routinely.

And third, do they exhibit the characteristics of *cooperation, camaraderie, and team focus*? Are they wild horses that cannot or will not pull together for the team but run off by themselves? Do they see themselves as part of something bigger, i.e. the team? Can they see the goals of the team and work towards them? As with the previous set of characteristics, references, recommendations and experience are important but here I call mainly upon the trusted returning members.

## 2.1 Diversity of Skills

Being an expert is important but having the right mix of skills is critical.[4] Having the right blend of talents and two-deep coverage on the team in vital areas can make all of the difference for red team success.

The skills necessary for a red team member cut across many areas and are changing every year. New languages, OS distros, applications, and networking gear add to the challenge each year. Some of the skills currently on the list are: Windows, Linux and Cisco platforms, vulnerability. Additionally, exploit development, exploit execution, persistence, stealthy techniques, web application exploits, and social engineering are valuable.

In comparison, the blue team must consider in its candidate selection and training the duplication of skill sets across their membership. This goal is primarily driven by the possibility that a member might be lost due to illness or as part of the exercise. Two-deep coverage on the red team is chiefly driven by the benefits derived from the mutual support and greater problem solving capabilities gained from “an extra set of eyes” and “a different point of view” of a difficulty. Therefore selecting at least two red team members focused or at least proficient at every skill area has proven itself a valuable goal.

## 2.2 Camaraderie

Not the most technical characteristic but amity and solidarity are none the less very important to the red team. The members of the team must not only respect each other’s technical skills but appreciate the opportunity to red team together. Disrespect or even antagonism can severely impact the performance of the team as a whole.

Since typically the Red Team comes from a wide geographical area, they may not know each other socially. The best indicator is how they worked as a team member during the previous year’s exercises. However, with new members that may not be possible. Soliciting feedback from established members is critical. Some of the best indicators come from Twitter and other social networking sites. The tenor of their posts, how others respond, and what other say about the candidates can reveal much about their character and how they might work as a team.

## 3 Provisioning the Team

Whether you are going camping in the Rocky Mountains or making dinner, you need certain resources and equipment to be successful. A red team has requirements as well that help ensure that they accomplish their goals.

### 3.1 Resources Required

The most obvious requirement is a computer. The workstation that the red team member works on is their main tool and a very personal one. Typically the red team members are required to prepare and bring their own workstation including any and all software they might require. Frequently they bring more than one and sometimes a server or networking gear as well.

There are more mundane resources needed by the red team. Besides your basic pens, paper, whiteboards, and markers, we have found several other valuable resources. In short lived, fast paced exercises, intra-team communication is crucial to getting the most out of the team. A bag of USB sticks helps quickly move data and tools around. A networked printer in the red team room for documentation and reports is useful. Keeping track of who has what IP address within the red team, what is known about the blue teams, who is focusing on what aspect of which blue team, and a host of other information can be facilitated by whiteboards, poster boards on the walls and lots of duct tape (lots). But this year, the best tools utilized by the red team for organizing and keeping track of both the blue and red teams was Armitage.[5] This GUI interface for Metasploit with its team collaboration support provided a great platform for intra-team documentation and coordination of effort. Armitage facilitated the coordination of members skilled at target acquisition, exploitation, persistence and score-able information harvesting.

### 3.2 Support Structure

In addition to the resources mentioned, the 2011 red team was supported by an individual on the red team dedicated to system support. This was one of the improvements requested by several of last year’s red team. With the compressed time frame of the exercise it was felt that an individual who could maintain support services such as a red team web, DNS, DHCP and other services as identified for the rest of the red team would aid in keeping the red team members focused on the attacks.

One of the time consuming and distracting tasks for the red team was recording and submitting scoreable accomplishments. The system support individual prepared and managed a system to make it easier for the red team members to construct a report of a new exploit or duplicate a similar existing report and modify it. This system also helped to sure that all required information and evidence was included in the report to make sure that it was grade-able by the white team.

## 4 Team Planning

Six to nine months before the event, the recruiting of red team members begins. It has to start this early to get on peoples calendars before other commitments. Even then their commitment can be superseded by employer priorities or family demands (new additions to a family do take priority). In four years of planning, there has always been at least one member whose plans get thwarted. Therefore, it is advisable to recruit at least one extra member for the red team.

The security community is a relatively small and remarkably close society. Coupled with the need for camaraderie and that the group will be working very closely and intensely for three long days, soliciting suggestions from returning members for new recruits is a big plus. They can also provide feedback on potential new members. This activity solidifies the members ownership of responsibility for the teams overall success and provides the red team captain with a much broader view of the perspective market place for new members.

### 4.1 “...Know thyself”

As the membership in the red team is incrementally established, team building activities can begin. Several mechanisms for intra-team communication have been tried: wikis, Google groups and docs, etc. Everyone's life is busy and you are asking these folks to volunteer a nice chunk of their time (much of it personal) with no remuneration other than some fame and bragging rights. Communication has to be easy. It has to be normal. None of the tools mentioned was used by a large enough segment of the team to become adopted. Plain old email has year after year ended up being the communication platform of choice that everyone could live with.

The first item of business is to introduce all of the members to each other. The captain typically starts with a short bio, background and skills. The rest of the team follows with their contribution. The captain should collect all of these as late joining members will need to be brought up to speed.

The next phase is planning a strategy for the event. The captain might start with some questions for the team such as: Do we assign red team members to each blue team or to each target type? Much of the planning is only instigated by the red team captain. Once started often the red team members direct the planning themselves with minimal steerage from the captain.

### 4.2 Clarifying the Goals of the Red Team

Red teaming is thrilling. The hunt and capture aspect is exhilarating. One problem that has been seen in previous red team exercises is the loss of focus on what is the red team's actual goal. That question is often answered with “Well, breaking into the blue teams systems of course!” The problem with that answer is that it is neither accurate nor realistic. An attacker in the wild would break into a system but that would not be their goal. Their goal is to secure a reward. That reward might come in the form of compensation for items acquired after breaking in such as credit card numbers, PII or trade secrets. Their goal is something on the other side of the door they forced open.

For the red team the goal is to score points. Those points come can come from breaking in but it does not stop there. Often the red team can become focused on the exploit and lose sight of the more realistic goal of obtaining database contents, credentials, PII and confidential documents. The exploit is professionally satisfying and therefore can itself become the focus. The red team needs to be encouraged to look beyond the exploit and focus on scoring as many points as possible.

Part of the planning of the red team is answering the question: “Now that we are in do we pillage or burn?” There is a part of most folks that want to “rm -rf /” when they get privileges. And although that does score points because you could do it and the blue team loses more points because they are down and miss service checks, is it the best approach if the goal of the red team is to score as many points as possible? The red team has tossed around this question many times and evolved an heuristic approach to the issue which will undoubtedly change again. Using the law of diminishing returns, once the score-able points gleaned from a system nears zero and no additional avenues of attack present themselves, plant as many backdoors as possible and bring the system down. Nearing the end of a day when it is no longer possible to recover, burn the system. The mental and morale strike of having a system down over

night is a tough hit. At the beginning of the last day wipe all systems possible so that recovery is futile.

## 5 Challenges Faced

Probably the greatest challenge to the red team is time. Realistically an attacker would be able to perform reconnaissance stealthily over a long period. The time compression of the event makes stealth difficult. It also makes recovery of lost persistence within the blue team costly in terms of points scored. The best defense against time for the red team is planning and preparation.

### 5.1 The Unknown

Time is not the only challenge to the red team. Typically few facts are known about the target infrastructure and nothing is known about the business injects that will be employed. The inclusion of injects that involve forensics on boxes, VoIP, SCADA, non-typical network devices or OSs, or other unusual services become difficult to exploit without time to overcome the often steep learning curve.

## 6 Duties of the Red Team Captain

Besides the recruiting, organizing, and provisioning responsibilities already mentioned, the red team captain works with the white team before the event to develop a working relationship and provide input on competition development.

It is helpful to red team morale if they can all stay at the same hotel. The red team often worked late into the night on exploit development and planning. Last year we were able to procure a meeting room at the hotel for part of the time to facilitate this after hours work.

Even with the scoring system, the red team captain must read and validate all of the scoring reports before they are forwarded to the white team. Last year we thought that the new red team scoring system would eliminate the captain validation step but it did not and likely cannot. Mistakes will be made and issues of misunderstanding of score-able events and reporting consistency between members of the red team will continue to require the captain's scrutiny.

Relative to scoring, the red team captain should work with the white team to clarify how the scoring will work. The red team members need to know what are considered score-able items or activities. The scoring function should emulate the relative value of various assets and difficulty of acquiring them. Without this

knowledge the red team must rely on guesses and assumptions.

One of the duties that keep the red team captain constantly busy throughout the event is taking questions from the red team to the white team for clarification and ruling. Is it allowed to do ...? Can we get points for ...? Does this rule mean ...? About a third of the red team captain's time during the event is spent resolving red team questions.

Another third of the red team captain's time is spent answering questions from the white team. Explaining what a scoring reports means is frequent but other white team questions come up as well. Is this alarmed event because of a red team action? Did the red team do ...? Did you folks brick ...? We are seeing ... - is that you folks?

### 6.1 Cautions for the Red Team Captain

The red team captain is typically going to be as much of a "techie" as the rest of the red team members. The attraction to "join in the fun" and perform exploits and dig for score-able things is magnetic. My experience is that when the red team's captain "plays" the duties that fall solely on the red team captain's shoulders do not get the attention necessary and both the red team and the competition can suffer. The caution is "even if you think you can wear two hats at once, your head is not really that big!" If you accept the role of red team captain, also accept that you will not be performing exploits.

## 7 After Action Report

Besides the competition aspect, the CCDC events are a powerful and unique learning experience for the students. Attending blue teams come with all levels of preparation and skill. Part of the value of the red team is to give the blue teams both encouragement to continue in the security field and comeback next year as well as feedback on what they did well and how they can improve in the future.

Traditionally the red team has conducted an after action debriefing to all of the blue teams at once at the end of the event. The report has been part introduction to the red team members and a general overview of what the red team has accomplished and observed. The problem was that the presentation was limited in time and could only be very general. Also new teams can get overwhelmed and possibly discouraged because of inadequate preparation and experience. These teams need encouragement and support to not give up.

Last year at the 2011 NECCDC, we tried something different with two teams that struggled during the competition. Their coaches asked us to stop in and talk to them giving some pointers and specific feedback. We then conducted our typical all team debriefing. Our experience with the two teams meeting with them individually was an epiphany. The change in their attitude and morale was striking. We spent 15 minutes with each of the two teams. We were able to provide specific feedback about their team. We also answered lots of their specific questions about what we did and what they could do to better prepare. At the end there was no question that they had benefitted from the competition and were coming back next year.

## 8 Conclusions

Our experience with assigning red team members to skill and application specific areas, instead of to a blue team, has served us well. It made the most effective use of our skills and attention.

It is imperative that the red team through the red team captain be involved with the white team before the event, consulting on the design and development of the exercise. A blind red team mainly tests the red team's skills not the blue team's skills and preparation.[6]

Our experience with the one-on-one debriefings with the blue teams has convinced us of their value to the students. We are recommending next year that time be allocated after the event to allow the red team to meet individually for 15 minutes with each of the blue teams. It is our experience that this provides better feedback and more values to the students.

## 9 Acknowledgments

This paper could not have been written without the shared experiences of many people. I would like to acknowledge the work of the 2011 NECCD Red Team: Jonathan Claudius from Trustwave, Laura Guay from SecureWorks, Raphael Mudge from Delaware Air National Guard, TJ OConnor from US Army, Ryan Reynolds from Crowe Horwath, Jason Nehrboss from CSC/Bath Iron Works, Joshua Abraham from Rapid7, Will Vandevanter from Rapid7, Gerry Brunelle from Boeing, Todd Leetham from EMC, and Silas Cutler from RIT. I also need to thank Tom Vachon from Kayak as the white team captain and Themis Papageorge from Northeastern University who organized and hosted the competition and many others.

## 10 References

- [1] Sun Tzu, *The Art of War* (Tribeca Books, 2011).
- [2] P. Carayon and S. Kraemer, "Red Team Performance: Summary of Findings University of Wisconsin-Madison & IDART: Sandia National Laboratories" (2004).
- [3] Furtună, et al, "A structured approach for implementing cyber security exercises", 2010 8th International Conference on Communications (COMM),
- [4] Dodge, R.C., Jr.; Ragsdale, D.J.; Reynolds, C., "Organization and training of a cyber security team," *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance*, 2003.
- [5] Raphael Mudge, "Armitage - Cyber Attack Management for Metasploit." [Online]. Available: <http://fastandeasyhacking.com/>. [Accessed: 18-May-2011].
- [6] P. Herzog, "OSSTMM 3 – The Open Source Security Testing Methodology Manual" [Online]. Available: <http://www.isecom.org/mirror/OSSTMM.3.pdf>. [Accessed: 18-May-2011].
- [7] G. B White and D. Williams, "The collegiate cyber defense competition," *Proceedings of the 9th Colloquium for Information Systems Security Education*, 2005.
- [8] P. Sroufe, S. R. Tate, R. Dantu, E. Celikel, "Experiences During a Collegiate Cyber Defense Competition," *Journal of Applied Security Research*, Vol. 5, No. 3, 2010, pp. 382–396.
- [9] J. Mattson, "Cyber Defense Exercise: A Service Provider Model," in *Fifth World Conference on Information Security Education*, 2007, 81–86.

# Dynamic Threat-resistant Node Identification in Sensor Networks

David Pearson, Sumita Mishra and Yin Pan

Department of Networking, Security and Systems Administration  
Rochester Institute of Technology, Rochester, NY, USA

**Abstract** – *There are numerous challenges in ensuring secure communications within modern Wireless Sensor Networks. In particular, valid node identification in sensor networks is a crucial aspect of maintaining data confidentiality and non-repudiation. Statically-assigned identities provide simple capabilities, but they also allow attackers to easily predict IDs—thus creating a susceptibility to sensor impersonation attacks. Dynamic identities, however, can allow for uses beyond that of a simple name for a device. This work focuses on proposing a solution that not only allows for dynamically-assigned node identities within a network, but also for defense-in-depth capabilities when used in conjunction with a security scheme.*

**Keywords:** Sensor networks, location schemes, sensor security, dynamic node identification

## 1 Introduction

Wireless Sensor Networks (WSNs) have been implemented in a variety of industries, including military/government, medical, and manufacturing sectors [1]. These dissimilar environments make standardization of the WSN size difficult. Due to a combination of this dissimilarity and the resource constraints of the sensor nodes, WSNs are a class of wireless network which can benefit dramatically from implementing location-based node identification.

There are also numerous challenges when trying to ensure secure communication within WSNs. Due to their generally reduced data transmission rate, the capabilities of high-traffic algorithms implemented for security in other platforms with more processing capabilities are significantly hampered when applied to WSNs. Additionally, nodes can in many cases be prone to physical tampering, as the devices may be found in remote locations. The most distinct caveat, however, is directly tied to one of the network's most essential advantages – long-lasting availability. Since these nodes are deployed to be autonomous for their entire lifetimes, it is imperative that they waste as little of their resources as possible [2]. Such resources include processing and radio capabilities which otherwise would greatly shorten the node's usable lifetime.

To effectively address the issues of secure and dynamic location-based node identification, these factors must be taken into account. Currently, applications of location-based identification implement a variety of technologies, but none have been discussed as a major solution due to issues with the

fundamental specifications of WSNs. Further, little research has been conducted in the topic of secure and dynamic node identification. Similar constructs have been used in other forms of electronic identification, perhaps most clearly in username and password verification within client-server systems.

This paper proposes applying a method of node identification for wireless sensor networks known as *dynamic threat-resistant (DTR) node identification*. Section 2 will discuss related work in this topic followed by the necessary assumptions in Section 3. The primary mechanisms used to apply DTR node identification will be described in Section 4. Section 5 will explain the benefits associated with implementing DTR node identification. Section 6 will provide test models for the application. Section 7 concludes the paper.

## 2 Related Work

As mentioned in the previous section, a large amount of research has been performed on the topic of identification of sensor nodes based on location, while limited research has been conducted on dynamic node identification. The most significant related works will be discussed below.

In [3], the authors explain that the sensor nodes receive their location information from a GPS sensor which is embedded in each node of the network. They point out that by having such a system, the issue of knowing the precise location of each node has been eliminated. While this seems to be a reasonable solution, some of the fundamental characteristics of a WSN are not considered. From a physical point of view, the denseness of a WSN is not standard. Therefore, nodes may be inches apart (or less) in some cases, which would prove to be too precise for many GPS implementations. Further, the sensor nodes would need to have a clear line-of-sight in order to communicate with the GPS, which is something that cannot be guaranteed. From a resource perspective, the power consumption of transmitting and receiving to/from some positioning device at a regular interval would not be feasible. Further, the cost and size of a node with increased sensor capabilities would conflict with the desire for smaller and cheaper nodes [4].

The proposition of node identification and security using location-based keys is discussed to different depths in [5] and [6]. In both, the authors describe utilizing a group of mobile robots to learn location information and create location-based

keys. Unfortunately, having the assumption that an adversary cannot or will not attack until these robots have completed their task is unrealistic in mission-critical applications.

In [7], the authors describe a boundary node-based solution. In this, there are a few powerful beacon nodes which send a signal (including their location information) to the network. Each receiving node crafts packets for each of its neighbors with the received information and its own information. When the original beacons receive these packets, the data is parsed and boundary nodes are elected based on a voting scheme.

Again, this solution overlooks some of the fundamentals of WSNs. It assumes that nodes are able to store and transmit a large amount of data, which is currently unreasonable. Further, no security is discussed, which exposes this proposal to a wide array of attacks.

The method of Approximate Point-in-Triangulation-based identification is discussed in [8]. In this approach, a handful of high-powered beacon nodes broadcast information. The end nodes within range triangulate based on the strongest three beacons, and therefore identify themselves as within that region. Not only is this method lacking security and computational awareness, but it also has the potential to create false positives and negatives.

Dynamic node identification was proposed in [9]. The authors discuss using a randomly assigned and semi-unique value to serve as the address within a node's network. This method involves assigning a new random value to *each* transaction within network transmission, regardless of whether or not the node has previously transmitted data. While it does have merit in pointing out the usefulness of locally (as opposed to globally) unique addressing within most WSNs, it does not take security into account. If each transaction involved a different ID, non-repudiation would be impossible. Additionally, it would likely be reasonably costly to produce a new random ID each time a transaction occurs on the network.

### 3 Assumptions

In order to provide a level of security and capability to such a scheme, some assumptions must be made. It is assumed that any WSN on which DTR node identification is used has deployed TinySec symmetric key encryption for all nodes [10]. By using such encryption, each node will be able to safely transmit its unique identification information within the network.

The second assumption is that the encryption scheme has not been compromised. It is imperative that it can be relied upon for the confidentiality and integrity of data traversing the network. Moreover, it is assumed that once the encryption scheme has been compromised, the network is no longer reliable.

A third assumption is that each node's environment has a form of uniqueness related to the sensors being utilized. This will be discussed in detail in the following section.

Finally, the assumption must be made that the network contains a certain number of cluster nodes which are higher-powered and much less resource-constrained than the remainder of the sensor nodes. These nodes, referred to hereafter as clusterheads, are the collection points in each region of a hierarchical WSN.

## 4 Mechanisms

To clarify the capabilities of using such a technique, it is essential to discuss its underlying mechanisms. In order to eliminate the cost of superfluous sensors which only serve the purpose of providing node location (such as a GPS sensor, for example), this model utilizes each sensor's immediate environment as an identifying feature.

In theory, each sensor's environment is unique. This uniqueness extends through a variety of different realms, including light information, sound, vibration, and more. Though such differences may not be visible or audible to a human without the aid of technology, they are in fact present, and can be used to identify each node in a distinct manner.

By capturing a "snapshot" of this uniqueness (hereafter known as the *First-Sensed Unique Identification Data*, or *FSUID*) at the very first instant that a node is powered on, a distinct identifier for the sensor node will be produced.

Once the unique identifier is created for a sensor node, it is sent to its neighbors and to the clusterhead node(s). All nodes which are 'directly connected' to the node store the identifier, while the rest simply pass it on to the clusterheads (or drop it if it has already been passed along).

Furthermore, the proposed solution also re-uses the FSUID as a random seed to an encryption algorithm designed to hide the FSUID within the node's filesystem. Doing so prevents (or greatly inhibits) the ability of an attacker to masquerade as or hijack a legitimate device within the network.

Another mechanism of DTR node identification is that it can provide node authenticity. In order to detect a node compromise, a challenge timer is used by the clusterheads to schedule when it will send a challenge packet to a node (see Figure 1). This packet requests the following two values from nodes:

1. A flag in the unencrypted header stating that an encrypted FSUID is appended to the layer 3 frame.
2. The encrypted FSUID itself.

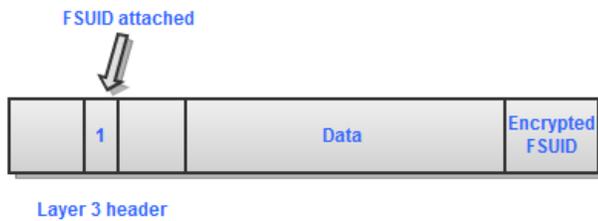


Figure 1: Challenge request frame information

The purpose of attaching the former value is to alert the receiving node(s) that this information not only includes transmission data, but that it also includes the response to a challenge packet by a node utilizing DTR node identification. If this value were set to *false*, the packet would only include standard data.

The latter value sent in the response packet must include the challenged node's valid FSUID. If the flag is set and the FSUID has either not been appended, has been corrupted (either through malicious means or otherwise), or the time period to which the node can answer expires (due to a collision or a node sleeping, for example), the clusterhead will send a second challenge packet to the device. If the node does not respond correctly to the second attempt within the specified time, the clusterhead transmits a broadcast message to alert other sensor nodes of the likely compromise. The suspected victim node is blacklisted from all information tables. This can be likened to a user answering secret questions in order to verify his or her identity to a server.

## 5 Benefits

Implementing a scheme such as DTR node identification has a variety of benefits both over location-based and static identification proposals in the field. To point out these advantages, they will be discussed based on their merits to the specific research topic. In logical order, location-based advantages are discussed first.

DTR node identification is versatile and sensor-independent. The uniqueness of the initially-sensed data is theoretically the same in practically all applications (light, audio, vibration, etc...), and therefore, no additional sensors must be added to the mote.

Second, this application involves something simple that the sensor node must measure in order to perform its duty correctly. This eliminates the issue of having to perform additional communication to find the sensor's location.

Third, there is no additional monetary cost associated with such a scheme. Because the application is sensor-independent, further costs elicited by other proposals (such as those including GPS) can be avoided.

Fourth, the amount of storage required is very low (though dependent on the size of the clusters). Since only a direct neighbor's location information and node ID is stored into a table on each end node, the increased work load is minimized.

In relation to the merits of the dynamic capabilities and its underlying mechanisms, there are numerous benefits as well. One inherent characteristic of utilizing dynamic identification is that the unique identifier becomes much harder to predict using a logic-based attack. Additionally, brute force methods would prove useless, as the challenge process only allows two attempts.

Because the FSUID is used as a random seed to hide itself within the filesystem, and because challenge packets are transmitted at a regular interval, it is difficult for an attacker to compromise and maintain access to a node on a WSN. This benefit provides an added level of authenticity to the topology.

Again, the dynamic characteristics allow a network to be setup in the field dynamically, instead of manually before or during deployment.

Finally, DTR node identification is flexible enough that it may have applications in other fields of computing. For example, it is possible that a computer can use a built-in webcam as a light sensor to provide itself with an FSUID. This could be implemented as an extra layer of defense in an 802.11 network, thus providing the access point(s) with another method of node authentication.

## 6 Test Cases

In order to provide situations where DTR node identification proves effective, the authors have derived test cases for some common network attacks—node compromise and masquerade. These scenarios will briefly describe the aim of the attack and how it fails to succeed within the proposed methodology.

### 6.1 Node Compromise

In a node compromise, an attacking entity's goal is generally to gain complete control of the node. This attack venue is attractive because the node is a legitimate device in the network, and therefore it is less likely to be discovered.

Using DTR node identification, the adversary will potentially be able to compromise the node for some amount of time. However, the challenge packet mechanism described in section 4 will eventually discover the intrusion (see Figure 2). Because the success of this particular test case relies heavily on the length of time between challenge requests, it is advisable to use the shortest interval possible in relation to normal traffic patterns within a particular wireless sensor network.

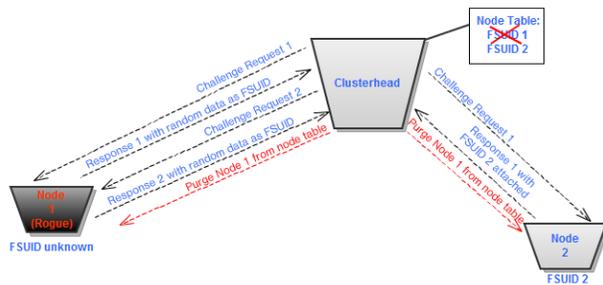


Figure 2: Node compromise challenge sequence

## 6.2 Masquerade

Masquerade attacks occur when an attacker releases a node which is supposed to mimic a legitimate node within the network. By doing so, the attacker can create a denial of service to the mimicked device, cause a change in the routing mechanism of the network, or inject false or malicious information into the network, for example.

By implementing DTR node identification, this attack is generally not possible. Because the network is encrypted using TinySec, the attacker must break through this mechanism before reaching the network functionality. At this point, the attacker would also need to collect and break the FSUID of the node to be impersonated, which is also encrypted (see Figure 3).

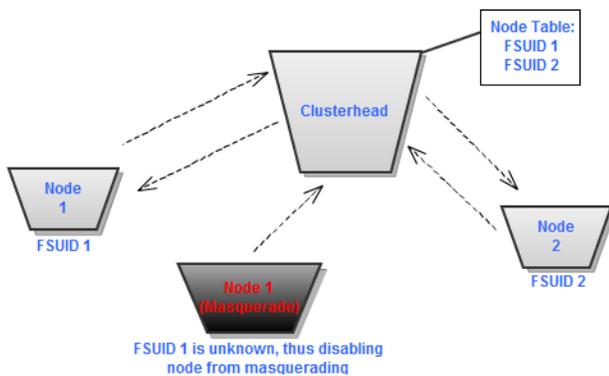


Figure 3: Masquerade attack fails without FSUID

## 7 Conclusion

It is easy to see that as WSNs become more widespread, the usefulness of location-based and dynamic identification will continue to increase for certain deployment types. Moreover, while the usefulness of such an implementation increases, the importance of employing a security scheme will also increase. It is for that reason that the authors feel it is necessary to work toward creating not only a cheap and lightweight location-based identification solution, but also one with the goal of keeping the adversary out and the data secure.

## 8 References

- [1] A. Wood and J. Stankovic, Denial of Service in Sensor Networks. *Computer* 35, 10, 54-62, Oct 2002.
- [2] G. Pottie and W. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51-58, May 2000.
- [3] S. Arisar and A. Kemp, A comprehensive investigation of secure location estimation techniques for WSN applications. *Security and Communication Networks*, 4: 447-459, 2011.
- [4] A. Srinivasan and J. Wu. A Survey on Secure Localization in Wireless Sensor Networks. *Encyclopedia of Wireless and Mobile Communications*, 2008.
- [5] K. Ren, K. Zeng, and W. Lou, "Secure and Fault-Tolerant Event Boundary Detection in Wireless Sensor Networks," *IEEE Trans on Wireless Communications*, , vol.7, no.1, pp.354-363, Jan.2008.
- [6] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol.24, no.2, pp. 247- 260, Feb. 2006.
- [7] X. Du, D. Mandala, W. Zhang, C. You, and Y. Xiao, "A Boundary-Node based Localization Scheme for Heterogeneous Wireless Sensor Networks," *Military Communications Conference, 2007. MILCOM 2007. IEEE*, vol., no., pp.1-7, 29-31 Oct. 2007.
- [8] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. 2003. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th Annual international Conference on Mobile Computing and Networking MobiCom '03*. ACM, New York, NY, 81-95. Sept 2003.
- [9] Elson, J. , & Estrin, D. (2001). Random, ephemeral transaction identifiers in dynamic sensor networks. *21st International Conference on Distributed Computing Systems*, 2001.
- [10] C. Karlof, N. Sastry, and D. Wagner, TinySec: a link layer security architecture for wireless sensor networks, *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04)*. ACM, New York, NY, USA, 162-175, 2004.

## **SESSION**

# **MISSION ASSURANCE AND CRITICAL INFRASTRUCTURE PROTECTION, STMACIP'11**

## **Chair(s)**

**Prof. Michael R. Grimaila**



# Availability Based Risk Analysis for SCADA Embedded Computer Systems

Stephen M. Papa, William D. Casper and Suku Nair

HACNet Labs, Computer Science and Engineering Department, Bobby B. Lyle School of Engineering  
Southern Methodist University, Dallas, TX 75275, USA

**Abstract** - Information Technology (IT) Security is often focused on Confidentiality, Integrity and Availability of software and data (information) contained in networked computers, servers and storage devices. In embedded industrial control or Supervisory Control and Data Acquisition (SCADA) systems the security focus must be on the protection of the availability of the system's functions. This basic paradigm change of maintaining availability is not subtle and system protection cannot be met with the application of IT security measures alone. This paper advocates focusing on maintaining Availability and using Confidentiality and Integrity to secure an embedded control or SCADA system from attack. A risk assessment process to identify attacks on each system element's availability is the first step to determining where protection mechanisms should be applied.

**Keywords:** embedded systems, security, availability, risk assessment, SCADA

## 1 Introduction

IT security including DMZs, firewalls, access control, third party authentication, least privilege, and other classical security mechanisms are essential in the protection of computer systems and these mechanisms create a foundation to protect a system that includes embedded computers [3]. Any networked system that has not implemented IT protection mechanisms is relatively easy to attack. However, traditional IT security protection mechanisms alone are not sufficient in ensuring the embedded control system's equipment configuration remains intact and the system availability is protected.

This paper will use SCADA systems as examples of an existing embedded and distributed control system requiring protection (note this can be applied to new systems as well). Figure 1 shows an example of a SCADA system and identifies the areas where enterprise IT security should be focused and where embedded security should be added. Reference [2] provides a good description of SCADA equipment and how these systems are networked and controlled. Using the convention described in this reference all remote field interface devices including Programmable Logic Controllers (PLCs) will be referred to as Remote Terminal Units (RTUs). A communication network connects the SCADA application server, Front End Processor (FEP), Historian (database) and RTUs. The network may include any combination of wire-line and RF physical components, communication standards and protocols. Included in the network are proprietary and standards based interfaces such as field control busses,

Ethernet, phone line modems, RF Modems, etc. Servers use the network to communicate commands to RTUs and receive status from RTUs. A Human Machine Interface (HMI) computer provides the system operators system status and critical alarm information. The HMI interfaces to the RTUs via the SCADA application server and Front End Processor [7].

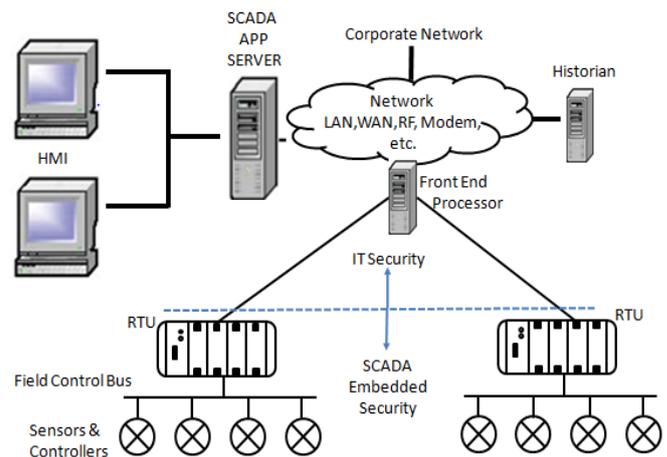


Figure 1: Example SCADA System Network

To determine the security mechanisms needed to protect a given piece of SCADA edge equipment (HMI, FEP, RTU, etc.) a risk assessment of the entire system focused on availability should be performed. Equipment protection requirements should be based on a risk assessment that accurately reflects the relative risks of each piece of equipment if under attack. Attacks may be via a network interfaces, other system interfaces, or by a person who has physical access and the desire to modify or change the equipment configuration. This paper will describe a risk assessment approach focused on SCADA embedded equipment where maintaining availability is the primary objective for adding security mechanisms.

## 2 Risk Assessment Approach

### 2.1 Overview

Figure 2 shows an overview of the risk assessment and vulnerability reduction process described in this paper. Prior to applying this process it is necessary to form a team with detailed and complete knowledge of the system being evaluated. Team members should include operators, maintainers, systems engineers, design engineers, operations engineers, IT security engineers, key suppliers

of equipment and members of management. Management needs to be part of the process or at least they should be periodically briefed by the team, otherwise, it is difficult for them to understand the scope of the vulnerabilities in the system. Finally if there are requirements for government regulatory of licensing requirements, then a person from the regulatory agency should be asked to participate at key reviews to allow for feedback during the risk assessment and vulnerability reduction process.

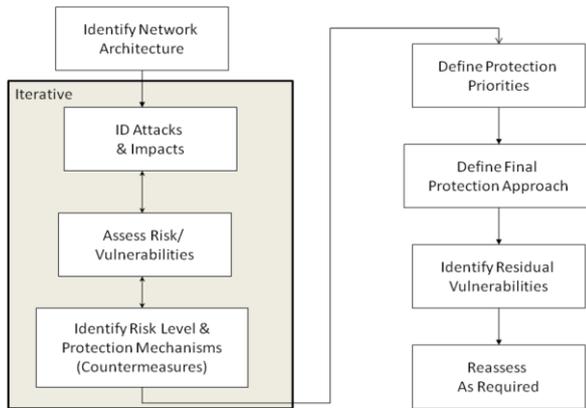


Figure 2: Risk Assessment and Vulnerability Reduction

## 2.2 Identify System Architecture

The first step in the risk assessment and vulnerability reduction process is to completely identify all equipment and how it is attached to the system network. This includes all hardware elements, internal and external network connections (control, internet or company intranet), and any existing security mechanisms that may already be in place. For each hardware element the expected software configuration should be identified. This configuration would include boot software, Operating Systems, network software, and all application software (for a SCADA system this would include RTUs, FEPs, HMIs, etc.). Network connectivity may include Ethernet, serial interfaces, wireless, modems, and other control buses/interfaces for edge devices. Existing security mechanisms may include firewalls, IT access controls, network encryptors, etc.

As these elements are identified detailed system architectural drawings should be created or updated. For edge devices it is necessary to identify all of the critical control elements within the device, and the expected configuration of each sub-element. The critical control elements include the connection mechanism by which the edge device is connected to the network and other access interfaces that the edge devices have. For example, if the edge device is connected to the network via an Ethernet connection and the edge device also has an RS-232 serial port available on it both of these connections should be considered as critical control elements. The RS-232 serial port could be used as an access point into the edge device to attack or disrupt the normal operation of the edge device. Any access point into the edge device is a critical control mechanism. Identifying all of the critical control

elements is an iterative process, and must be completed prior to proceeding with the risk assessment.

## 2.3 Identify Attacks and Impacts

An attack may originate from inside or outside of the organization; however, any threat implies a desire to perform a malicious act. The risk analysis threats and attacks are defined as follows:

1. **Insider Threats:** Authorized personnel within the organization (employees, contractors, etc.) who have either physical access to equipment or network access.
2. **External Threats:** Unauthorized personnel with network access only or unauthorized personnel who gain physical access to equipment by circumventing the physical protection mechanisms.
3. **Attacks on Availability:** An attack to prevent an element of the system from performing its designed functionality resulting in undesired loss of functionality, inoperability of equipment, regulatory compliance violations, loss of service, or creation of any safety or health hazard.
4. **Attacks on Confidentiality:** An attack to observe or obtain software, firmware, data or other system information not normally available to the attacker. Insider threats may have access to the system but not the specific software or data while external threats should not have access to the system, the software, or the data.
5. **Attacks on Integrity:** An attack to temporarily or permanently modify software, embedded firmware (e.g. FPGA configuration data) or data used or generated by an element of the system.

Based on the attack type the specific goals of the attack must be identified. The root concern of an attack on a SCADA system is the loss of availability. Potential loss of availability may include typical network disruptions (external attacks) such as interruption of communication (Denial of Service), replaced network messages (Man In the Middle attack) resulting in a network disruption or incorrect command sequence to an edge device, or insider attacks including modification of the hardware or software configuration to cause immediate or delayed failure of other erroneous functionality. SCADA IP Network vulnerabilities and attacks are fairly well defined by IT security experts [1][3][5][6][8][9][10][11][12]. NERC and the U.S. Department of Energy had published an annual description of the top ten vulnerabilities for SCADA [1]; however, the last publicly available published list is from 2007. Additionally, NCS provides a list of network based attacks, motives, and impacts of the attack if successful. Of the NCS identified attacks the following are primarily availability attacks:

1. Intercepting messages and providing false information to take control of the SCADA system,
2. Insertion of software, hardware or firmware to take control of the system. This may include planting a "Trojan" or other malware. The goal is to take control of the system and issue controlling commands that normally originate from operators. While the inserted or Trojan element may also be an attack on the integrity of the system, the SCADA concern is focused more on the impacts to the availability of the system caused by inserted element.
3. Modify status from remote system elements to mislead Operators. The goal may include causing the operator to believe the control processes are either operating normally when these are not, or that the system element needs to be shut down [2].

Other publications [4][6][10] have identified network attacks based on the attacker having the ability to create network messages or intercept messages. These attacks may include data intercept and manipulation, DoS, Address spoofing (to act as an edge device), sending unsolicited responses (false status), protocol/packet replay attacks, log data modifications, or other unauthorized control. Regardless of the embedded control system (SCADA or other control system) the end goals may include taking control of the system, causing the RTU or operator to make incorrect corrective commands, or disrupting the processes being controlled [4]. Many of these attacks are possible when the system has no message or user authentication, no encryption, minimal error handling and exception handling. This type of attack may be possible for a person who has direct access to equipment or only network access.

Network Attacks are not the only way to cause loss of availability. A physical attack has different characteristics due to the fact the attacker may be able to by-pass IT security or physical security and make modifications that are not detectable using typical security protection measures. An attacker having physical access to equipment may also take control of the system directly, insert a Trojan, or change information to cause loss of system availability.

In addition to loss of system functionality the result of either attack type can be loss of production, loss of service or creation of a safety hazard. This can result in a potential for loss of revenue, fines, lawsuits, loss of customers, or a combination of any of these. All of these affect the enterprise's short and long term profitability and may affect the organizations ability to continue in business.

IT security mechanisms are widely known and provide significant security if deployed correctly. However, if an organization fails to heed the warnings of the IT security department or fails to install and maintain proper IT security mechanisms the system in question may contain

vulnerabilities that can be exploited. For the purposes of the risk assessment a worst case scenario would be to assume that the attacker is able to exploit weaknesses in the IT security and is able to monitor the network, intercept messages, and modify or insert data. The attacker may have physical access to edge equipment, an RTU for instance, and make modifications to its configuration. The risk assessment process will help identify the critical equipment that needs to be protected from these worst case attack scenarios.

## 2.4 Assess Risks

A successful availability attack will result in loss of system functionality. A failure analysis provides an accurate indication of the resulting risk of an attack that causes equipment to fail, not perform its intended operation, or that causes other system degradations. Fault trees and event trees may exist for a given system, and redundant or fail-safe equipment may already be in place to protect against failures. This failure analysis should be re-used or created as required to understand the risks associated with an availability attack.

An availability attack risk is a measure of the likelihood that a successful attack can be performed that causes a loss of availability of one or more redundant or non-redundant system edge devices coupled with the consequences incurred by the system's loss of availability. An attack may include any combination of modifications of the existing configuration (system hardware, software or firmware), changes in control device status to the operator, or changes to the operator commands to the controlling device. The effect of an attack, or consequence of occurrence, may include interruption of system operation, disablement of all or part of the system, or modifications to cause damage or permanent failure of equipment. Other consequences of an attack may result in a disruption of service, loss of production, or creation of safety hazards

The assessment process must identify risks to equipment so that protection measures can be identified that will reduce vulnerabilities associated with potential attacks. This process should help identify and prioritize where to apply protection mechanisms to reduce the likelihood of a successful attack. The steps to identify risks are as follows:

1. Select the system element (edge device or interface) to evaluate
2. Determine the effect of a failure of this element
3. Determine the threats and type of attacks that could cause this element to fail
4. Determine likelihood and consequence of occurrence of the attack being successful
5. Score the risk for the system element under evaluation
6. Determine protection mechanisms to reduce risks

These steps should be repeated for each system element. This risk assessment will focus on the SCADA edge devices performing system control functions to

identify vulnerabilities and to identify the relative risk to availability for that system element. As part of this process attacks to system elements performing critical functions are identified. Attack types are described later in this section.

For each attack an initial Consequence of Occurrence ( $C_o$ ) and Likelihood of Occurrence ( $L_o$ ) must be established. The  $C_o$  and  $L_o$  are estimates of the result of the attack and the likelihood of the attack occurring. The initial assessment is performed with the assumption that no protection mechanisms are present in the embedded SCADA equipment. This first pass establishes a baseline for comparisons when new or existing protection mechanisms are added to the system elements and the risk is reanalyzed.

$L_o$  is inversely proportional to the cost of the attack method. For this measurement basis cost is a combination of the financial cost an attacker incurs to execute the attack plus the skills and knowledge the attacker needs to possess to be successful in their attack. Therefore an attack that requires a high cost to the attacker has a lower probability of being attempted, or in other words has a low  $L_o$ . Likewise an attack that is inexpensive or easy to perform has a higher probability of being attempted, and therefore has a high  $L_o$ .

The  $L_o$  level can be categorized based on the possibility that the system element may be attacked and the following criteria is proposed as one way to differentiate between levels:

1. Low  $L_o$ : A very high cost or special knowledge is required to attack. Time to attack is prohibitive. No network or physical access to equipment is available.
2. Minimal  $L_o$ : A high cost and time is required for an attack. Technology exists for the attack but is not readily available. Effort required to access network or equipment. The attack requires special knowledge.
3. Moderate  $L_o$ : A moderate cost and time is required for an attack. Technology exists for attack and is available. Effort required to access network or equipment.
4. High  $L_o$ : A small cost and little time are required for an attack. Technology exists for the attack and is readily available. Some effort required to access network or equipment.
5. Very High  $L_o$ : Very little cost and time is needed to attack. Technology to attack exists. It is relatively simple and cost effective to attack. Access to network or equipment is readily available.

The  $C_o$  estimate is based on a loss of availability (system performance or functionality) and the potential impact of that loss of system availability. The following criteria are proposed as a way to differentiate between

levels which may include one of the following five categories:

1. Low  $C_o$ : No impact to system or service is incurred, no loss of equipment, no loss of market share.
2. Minimal  $C_o$ : Negligible impact to system or functionality, negligible reduction in market share.
3. Moderate  $C_o$ : Moderate reduction in functionality or system operation, reduces organization's market share. Negative impact to public image.
4. High  $C_o$ : Results in potential loss equipment or functionality, or significantly reduces organization's market share. Significant negative impact to public image.
5. Critical  $C_o$ : Results in total loss of equipment and service. Eliminates organization from market place due to extreme negative impact to public image or financial responsibility to repair the damage caused by loss of service or functionality.

Certain attacks may result in hazards or safety issues and the  $C_o$  of the attack could result in loss of human lives, or undermine the organization's ability to continue in business, especially in SCADA systems. An organization may experience public and financial impact due to the loss of system functionality from a successful attack.

Finally, the  $L_o$  is independent of the  $C_o$  and each can vary from low to high as proposed above. Key factors in determining the  $L_o$  is estimating the cost to perform the attack, the effort required, the level of access required, and any special knowledge required ensure a successful attack.

Once the initial  $L_o$  and  $C_o$  are established a risk score is created. Figure 3 shows an example of a risk scoring matrix. There are many approaches to risk assessments and each has a method to compute a risk score. These methods range from rigorous mathematical analysis (Dempster Schafer Evidence Theory, or Bayesian Networks) to fuzzy logic or neural networks based algorithms [13][14][15][16][17]. In general these methods can be difficult to understand and use in practice due to the complexity of the mathematical algorithms utilized. Significant training may be required and the time spent modeling the system may result in the risk analysis never being completed. A simpler method will result in quicker results in establishing the system risks and protection goals and is usually adequate for understanding the risks relative to each other. In the process outlined here the purpose in identifying risk is to assist in prioritizing which parts of the system should be modified to protect against attacks. Based on application of this risk process the risk score defined below is relatively simple, easily modified, and useful in most risk assessments. The risk score can be computed by a weighted sum as follows:

$$\text{Risk Score} = A * L_o + B * C_o \quad (1)$$

The weighting factors (A and B) are selected based on the system, and their sum is equal to 1.0. In the example

provided  $A = 0.4$  and  $B = 0.6$ . In most SCADA cases  $B$  will be greater than or equal to  $A$  due to the fact that the consequences related to a SCADA system failure can be life-threatening, so even a lower likelihood of attack may warrant more protection since the consequence of failure could be catastrophic. The  $L_o$  levels and  $C_o$  categories were given whole number constant values in the included risk scoring example due to the ease of understanding that a change in level or category results in a change in  $L_o$  and  $C_o$  values, and whole number differentiation is easy for the majority of people to understand. While this may seem arbitrary it is a common practice in risk analysis efforts to have a normalized table to allow easier understanding of the relative risk level between one risk element and another risk element. The resulting risk score is not a mathematical probabilistic analysis of risk, rather it provides a relative quantification of risk for use in comparing other system elements where the same criterion is used for scoring the elements in the system. The risk score should be an indicator of the level of trust needed and the priority for this element to be modified to support the protection of the elements critical functionality. It is also important to note that adding protection mechanisms may reduce the  $L_o$ ; however; the  $C_o$  cannot change with the addition of protection mechanisms. To change the  $C_o$  a change in the design to improve the systems safety or fault tolerance is required (and not the subject of this paper). A  $L_o$  goal should be established prior to selecting protection mechanisms that are being considered to counter the identified attacks. Setting the goal in advance will provide a basis for selecting a design, and lets the designer know when to stop applying protection.

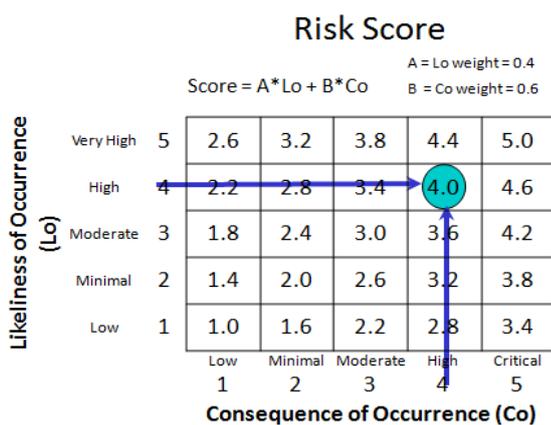


Figure 3: Scoring the Risk

### 2.5 Identify Trust Level and Protection Mechanisms

Based on the risk score an appropriate level of protection should be established. A low risk score is indicative to a low  $L_o$  and  $C_o$  and a minimal level of protection is needed for the system element to be trusted, while high risk scores require more protection to be trusted to be available.

### 2.6 Trust Levels

A Risk Score may be mapped into a Trust Level (TL) which in turn determines an appropriate set of protection

mechanisms to be used. For each TL a set of system protection mechanisms should be implemented to provide a consistent level of assurance that the system element will operate during or after an attack. The following proposed trust levels allow differentiation between risk scores while providing guidance for the types of proposed protection mechanisms to mitigate the risk. These proposed trust levels are loosely based on the FIPS 140-2 security levels for cryptographic modules [18] and illustrate the balance needed between risk versus consequence since additional protection often leads to increased cost.

The first Trust Level (TL1: a risk score less than 2) represents a very low risk of loss of availability and therefore no specific protection mechanisms are required. The second Trust Level (TL2: a risk score less than 3) represents a low risk of loss of availability and protection mechanisms should be added if already available but not enabled or if it is added at a low cost. The third Trust Level (TL3: a risk score less than 4) represents a medium level of risk for loss of availability and protection mechanisms should be considered including use of encryption for messaging, verification of integrity of all commands and status, reporting of invalid commands and verification of integrity of software, firmware or hardware at power up or reset.

The fourth Trust Level (TL4: a risk score less than 5) represents a high risk of loss of availability and inclusion of protection mechanisms should be considered. TL4 protection mechanisms include use of the protection mechanisms from TL3 plus the following:

1. verification of integrity of software or firmware prior to use, and during run time,
2. maintaining safe operation during an attack,
3. monitoring external interfaces,
4. monitoring and protection of hardware and key material,
5. verification of hardware configuration,
6. two factor authentication of users connecting directly to equipment,
7. test and evaluation on purchased software to ensure secure development practices were followed.

The fifth Trust Level (TL5: risk score = 5) represents a critical risk of loss of availability and inclusion of protection mechanisms should be considered including use of the protection mechanisms from TL4 plus the following:

1. TL4 plus the capability to restore system configuration, maintain availability and operation of system during attack,
2. hardware is designed specifically to meet high level of protection,
3. protect against instrumentation, external memory is encrypted to ensure confidentiality, etc.

### 2.7 Define Protection Priorities

Prioritize protection based on system approach and add protection based on system elements with the highest risk score (high  $C_o$  &  $L_o$ ). Due to budget constraints it is unlikely that all equipment at all risk levels will be

included in the final list of protected equipment. Selection of protection mechanisms should be performed to ensure the best coverage of elements with the highest risks, and if all protection mechanisms cannot be implemented then partial protection should be evaluated to see if the resulting risk reduction is worth the investment that is to be made. Any unprotected or partially protected system element results in a residual vulnerability (partially unprotected system element). It may require multiple passes through the risk assessment process to find the mix of protection mechanisms that are within budget and meet the organization's availability protection needs. The individual risk areas will be identified by assessing the attack vectors and risk scores for each of the system elements and comparing the elemental risk scores relative to each other. Higher risk scores for an element indicate that element is a higher risk area.

## 2.8 Identify Final Protection Approach

The final protection approach should result in a coherent and complete security architecture being identified, and should be based on budget and other organizational constraints (labor, planned system down time, etc.). A phased plan to add protection mechanisms should be created. As system elements are replaced or a new system is built then adding protection mechanisms that support the architectural approach should be required.

While much of the discussion above was protection centric, all of the basic protection requirements (protect, detect, report, respond, restore) should be considered when defining the security architecture. There are dependencies when implementing these goals. For instance the system must first detect an attack before being able to report it, however, protection is independent of these two goals. Finally, in most cases the cost increases as more of the goals are implemented and so design trade-offs are required.

In general protection priorities must be established. With an organization with unlimited budget all of the protection mechanisms identified for each system element can be implemented and the system will have a high level of security. However, the reality is that the budget may run out before the system is fully protected, and so the highest risk hardware elements should be protected first.

Existing equipment with limited resources (memory, processing, interface bandwidth, etc.) may require replacement, however, this may be hard to justify without understanding the risk reduction and potential cost avoidance versus the replacement cost.

If budgets are very tight then the first step should be to identify low cost options that provide the highest risk reduction. Some simple short term (IT Security) measures may include disconnecting the SCADA network from the internet and corporate intranet (good in theory); because an air-gapped network can remove all external attacks (insider attacks are still an issue). This may also include

constraining operator PCs to SCADA control/status functionality only. Providing separate computers for e-mail, internet access, etc. on inter/intranet may be less costly than protecting the entire SCADA system against network attacks. Another option is to tighten Firewall rules or create DMZs to minimize network connectivity risks.

New systems may make it easier to include protection mechanisms due to the fact that there are no existing equipment constraints to work around. The extra cost for the protection mechanisms may still be an issue, and it may be hard to justify protection to any organization's management focused on cost (what management is not?). Cost avoidance must be quantified by the decrease in risk, and the cost of the consequence of a successful attack. Decreasing the likelihood of an attack would decrease the likelihood of the expenditure that resulted from a successful attack for that risk element. Reducing an element's risk from one that has a high baseline risk to one with a low baseline risk after the protection mechanisms are appropriately identified and implemented would result in cost avoidance assuming that the protection mechanisms chosen successfully prevent the attack from occurring. The cost of the added protection should be justified by understanding the reduction in risk and potential cost savings that results from implementing the protection mechanisms.

For existing systems the thought of replacing "perfectly good" equipment is a difficult position to take. For these systems adding equipment to provide some protection may be the best solution. For example, for RTUs that use PC-104 stackable hardware it may be possible to insert an encryption card to protect messaging to/from the host computer. Ultimately a system architectural approach to protection must be defined with an approach that prioritizes the addition of protection mechanisms based on the highest  $C_o$  and  $L_o$ .

Finally, the total cost of protecting a system should never exceed the cost associated with the potential attacks. In other words apply common sense and do not let fear of an attack cause you to spend more protecting a system than the total projected losses due to an attack. This must include additional operational costs, maintenance costs and equipment costs.

Therefore the final approach is based on the following steps:

1. assessing the baseline risks to the various elements of concern without an added protection,
2. mapping these baseline risks to the resultant proposed trust level for each element,
3. determining the desired trust level based on a balance of cost, risk, and protection mechanism viability,
4. reassessing the overall risks to the system assuming the protection mechanisms are implemented,
5. verifying that the final risks are now at an acceptable level given cost and protection mechanism viability constraints,

6. and finally documenting the residual vulnerabilities.

## 2.9 Identify Residual Vulnerabilities

Residual vulnerabilities are those that due to priorities, lack of suitable protection technologies, budget or other constraints are not protected. Other possibilities for not protecting an element include when it is technically unfeasible, or if the protection mechanisms are too costly to implement (risk reduced is not worth the cost of protection). In any case these remaining vulnerabilities are called residual vulnerabilities. It is helpful to have a list of residual vulnerabilities and any unimplemented protection mechanisms that reduces them so that future upgrades or equipment replacements have a starting point for defining protection requirements. If there are licensing or regulatory agency approvals then the initial risks, protection mechanisms added and residual vulnerabilities may be useful for inclusion in regulatory or other assurance documentation when seeking approval of the system's protection approach.

## 2.10 Reassessment

As the network changes, new equipment is added, the attacks change, or risk scores change ( $L_o$  or  $C_o$  changes) then the risk assessment process should be repeated on all or parts the system. Follow-on assessments should be easier if there is adequate documentation from the last assessment, and it is usually more accurate than memories of the participants. As the system is changed documentation of the changes may also assist in providing assurance information for any required regulatory or licensing assessments

## 3 Conclusions

IT security is important, however, it does not fully address embedded system security, physical or network attacks on embedded computer equipment, or have a primary focus on maintaining system availability. IT security may also fail to protect the system when the attack originates from within the SCADA network. Edge devices such as RTUs need to be hardened to protect the availability of the device to ensure an attack cannot cause loss of production, service or create a hazard to people in or near the system under attack.

For SCADA systems an Availability based risk assessment should be performed to reduce system vulnerabilities. A failure analysis can be used to help identify the system behavior that may result due to network and physical attacks. A risk score is created based on this assessment and a trust level based on the risk score was also provided to identify protection mechanisms required to reduce availability attack risks. The addition of protection mechanisms should be prioritized based on high or significant risk ratings (high  $C_o$  and  $L_o$ ).

If the  $C_o$  and  $L_o$  of an attack on any system element is significant or high then the cost of not protecting the equipment under evaluation may result in severe or even catastrophic results to system or the enterprise.

## 4 References

- [1] "Top 10 Vulnerabilities of Control Systems and Their Associated Mitigations – 2007", North American Electric Reliability Corporation Control Systems Security Working Group, U.S. Department of Energy National SCADA Test Bed Program, March 22, 2007
- [2] National Communications System, Technical Information 04-1, "Supervisory Control and Data Acquisition (SCADA) Systems", October 2004
- [3] NIST Special Publication 800-82 Revision 2, "Guide to Industrial Control Systems Security", Final Public Draft, September 2008
- [4] Jack Wiles, "Techno Security Guide to Securing SCADA", Syngress Media, 2007
- [5] Vinay M. Ijure, "Security Assessment of SCADA Protocols", Lightning Source, Inc., 2008
- [6] Ronald L. Krutz, "Securing SCADA Systems", Wiley, 2006
- [7] Gordon Clark and Deon Reynders, "Modern SCADA Protocols: DNP3, 60870.5 and Related Systems", Elsevier Ltd, 2008
- [8] "The Role of Authenticated Communications for Electric Power Distribution", Pacific Northwest National Laboratory, U.S. Department of Energy, November 8-9, 2006
- [9] NIST Special Publication 800-53 Revision 2, "Recommended Security Controls for Federal Information Systems", Information Security, December 2007
- [10] Mariana Hentea, "Improving Security for SCADA Control Systems", Interdisciplinary Journal of Information, Knowledge, and Management, Volume 3, 2008
- [11] Jeff Dagle, PE; Presentation: "Potential Mitigation Strategies for the Common Vulnerabilities of Control Systems Identified by the NERC Control Systems Security Working Group", Pacific Northwest National Laboratory, 2010
- [12] Aakash Shah, Adrian Perrig, Bruno Sinopoli, "Mechanisms to Provide Integrity in SCADA and PCS Devices", International Workshop on Cyber-Physical Systems Challenges and Applications (CPS-CA), June 2008
- [13] Dong-Mei Zhao, Jing-Hong Wang, Jing Wu, Jian-Feng Ma, "Using Fuzzy Logic and Entropy to Risk Assessment of Information Security", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005, IEEE
- [14] Lu Simei, Zhang Jianlin, Sun Hao, Luo Liming, "Security Risk Assessment Model Based on AHP/D-S Evidence Theory", 2009 International Forum on Information Technology and Applications, IEEE
- [15] Xiao Long, Qi Yong, Li Qianmu, "Information Security Risk Assessment Based On Analytic Hierarchy Process and Fuzzy Comprehensive", The 2008 International Conference on Risk Management & Engineering Management, IEEE
- [16] Yu Fu, Yanlin Qin, Xiaoping Wu, "A Method of Information Security Risk Assessment Using Fuzzy Number Operations", 4<sup>th</sup> International Conference on Wireless Communications, Network and Mobile Computing, IEEE 2008
- [17] Chaoju Hu, Chunmei Lv, "Method of Risk Assessment Based on Classified Security Protection and Fuzzy Neural Network", 2010 Asia-Pacific Conference on Wearable Computing Systems, IEEE
- [18] Federal Information Processing Standards Publication (FIPS) 140-2, "Security Requirements for Cryptographic Modules", National Institute of Standards and Technology, May 25, 2001

# Mission Assurance Implications for Federal Construction by Building Information Modeling Implementation

Krishna R. Surajbally<sup>1</sup>, Peter P. Feng<sup>1</sup>, Ph.D., P.E., William E. Sitzabee<sup>1</sup>, Ph.D., P.E., and Patrick C. Suermann<sup>2</sup>, Ph.D., P.E.

<sup>1</sup>Graduate School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA

<sup>2</sup>Air Force Center for Engineering and the Environment  
Kelly AFB, Texas, USA

**Abstract** - *The increasing use of Building Information Modeling in the commercial sector has affected construction in the federal sector. The Architecture, Engineering, and Construction Industry performs design and construction for federal agencies and using Building Information Modeling will impact the federal construction process. Building Information Modeling is a design process that operates in an information technology environment. It contains dynamic and interactive features that allow for greater efficiency in the design and construction of a facility. However, the use of Building Information Modeling in federal construction does present some degree of risk because of these features. The authors identified potential security risks associated with its implementation by studying the United States Air Force Military Construction process. As risks were identified, mitigation measures were recommended. Federal agencies involved in construction must be cognizant of these risks and their related costs*

**Key words:** Building Information Modeling, Military Construction, Federal Construction, Mission Assurance, Critical Infrastructure, Risk

## 1 Introduction

As the use of Information Technology (IT) in construction becomes more widespread in the Architecture, Engineering, and Construction (AEC) industry, it will impact federal construction [1]. Most communication between AEC firms and federal agencies are done electronically; these communications include solicitation, proposals, contract documents, and project designs. Solicitation is done by advertising project requirements on the website <https://www.fbo.gov/> [2]. AEC firms interested in the project must register for access to the site and can submit their proposal electronically. When an AEC firm is selected for a project, electronic exchange of contract documents between the AEC firm and federal agencies begin [3]. This communication grows to include the project design drawings, bill of material, schedules, cost estimates, personnel information, and other construction information. The security of these sensitive communications relies heavily on the IT infrastructure and security protocols [4].

Since Building Information Modeling (BIM) is considered the next generation of design technology [5], its functionality depends on the robustness of the IT system that supports it [6]. BIM will also be affected by existing IT infrastructure and security protocols. The consequences of compromised BIM data will be more significant than conventional design since BIM contains much more details of the project in a single model. These details include a 3-dimensional (3D) model of the facility, bill of material, schedules, cost information, and interactive design attributes. The risks associated with facility design using BIM is of greater concern to federal agencies like the Department of Defense, who invests a significant amount of money on critical infrastructure, which is expected to be secure and not prone to security breaches.

The Department of Defense Military Construction (MILCON) Budget represents a large amount of construction business for the AEC Industry with an annual average of over \$15 billion for the past 10 years [7]. AEC firms attempt to be most efficient in their construction process to be competitive for federal construction projects. At the same time, federal agencies seek to obtain the best value for the Government when awarding contracts for construction projects [3]. BIM offers a method to effectively design a facility while maximizing work performance during construction [5]. For these reasons, the AEC Industry and federal agencies, including the United States Air Force (USAF) and United States Army Corps of Engineers (USACE), are currently implementing policies to require the use of BIM in the design and construction of federal facilities [8]. BIM offers the ability to design a 3D model of a facility that exists in a dynamic, interactive environment. BIM models are very detailed and show every aspect of the facility [5]. The power of BIM to produce precise and accurate design details poses a security risk to the federal construction process by allowing details not available in conventional 2-dimensional (2D) design to be made public. The concern is greater if the facility is one that handles confidential or classified information such as embassies and intelligence operations [9].

This paper examines the potential risk to mission assurance involved in implementing BIM in the federal construction process. The study will focus on the MILCON

program used by the USAF where the USACE performs the contracting and project management function for military services. The USACE coordinates with the selected AEC firm to execute the construction of the facility on behalf of the USAF [6].

## 2 What is BIM?

BIM is a design process that produces an informational model of a facility; BIM is not considered a product [5]. The model is “a computable representation of the physical and functional characteristics of a facility” [5] and is made up of objects represented by graphical lines, shapes, and symbols. The objects contain attributes with specific properties such as product information, solid or void spaces, material specification, and space orientation. These objects allow the model to be conceptual in nature or detailed enough for construction. BIM also provides tools for selecting, extracting, and editing the objects' characteristics. The ability to select and manipulate objects in the model allows for the viewing of specific sections of the model from different orientations. The selected objects remain consistent in size and location in all views. This consistency eliminates these types of errors that occur in 2D modeling.

BIM also defines objects parametrically so that they serve as parameters in relation to other objects. This feature allows for universal editing of an object's property; if a change is made in one object, parametric-related objects would automatically change based on the properties programmed in the original objects [5]. Any change is updated in the entire model; these changes can be as complex as material specification or simple as paint color.

The properties of objects in BIM are also computable, which allows for cost estimation, creation of bills of material, and clash detection before any construction begins. The computability feature can also be used to analyze energy use, lighting, acoustics, heating, and other features that will exist when the facility is complete. These capabilities allow for better collaboration during the design process whereby owners and the AEC firm can explore configuration possibilities. Since BIM exists in an dynamic and interactive environment, designers, engineers, contractors, and subcontractors can view the model in real time and determine how changes would affect their part of the construction.

The current process of facility design uses 2D drawings, which are created by computer-aided design (CAD) software. These techniques produce an electronic 2D drawing that can be transmitted electronically or printed on paper [5]. The use of conventional design methods do present some risk, because it contains information needed to construct a facility. However, the risk is considered low since multiple sets of drawings are required to produce a complete model.

The conventional design exists in electronic or paper format and is not as easily visualized as in a 3D model.

Although the 2D design process creates and represents all the information needed for the facility, this information exists in separate and distinct drawings. It is difficult to conceptualize a facility and its component systems from a set of 2D drawings. The 2D modeling process does not have the ability to select or deselect various attributes of the design. Designers and engineers incorporate changes in 2D design by editing paper copies and submitting them to the design owner or CAD operator for revision.

The capabilities of BIM may affect the way in which the federal construction process works, so its implementation must be considered at all steps in the construction process. The USAF MILCON process includes Requirements, Programming, Funding, Solicitation, AEC Evaluation, Award, Project Validation, Design and Construction, and Project Management [3], [10]. Risks can be identified by studying these steps.

## 3 MILCON Process

Figure 1 shows the major steps of the MILCON process, which are broadly classified into three phases for this study: the Conception Phase, the Planning Phase, and the Execution Phase. The areas of control for the USAF, the USACE, and AEC are shown on the horizontal tracks. These three entities play specific roles in the construction of federal facilities and must follow federal contracting and construction requirements. The USAF, as the main customer, is involved in all aspects of the MILCON process. The USACE becomes involved once the project is funded, and the AEC firm is involved in the Solicitation, Project Validation, and Design-Build Steps [3], [10].

Figure 1 also shows the steps of the MILCON process where BIM is used; these steps are highlighted by hatch marks and only these steps will be evaluated and given a risk rating. As risks are identified, the corresponding ratings are determined by using the levels of Threat, Vulnerability, and Consequence for that particular step.

### 3.1 Conception Phase

The Conception Phase consists of the Requirements, Programming, and Funding Steps. BIM is not used in these steps.

*Requirements:* Translates a need or request for a facility into quantifiable requirement documents.

*Programming:* Develops justification for the facility request and seeks approval.

*Funding:* All MILCON projects must be approved by the United States Congress [3]. For this reason, the project request is vetted and prioritized at several levels before it can be funded. After Funding, the project moves to the Planning Phase.

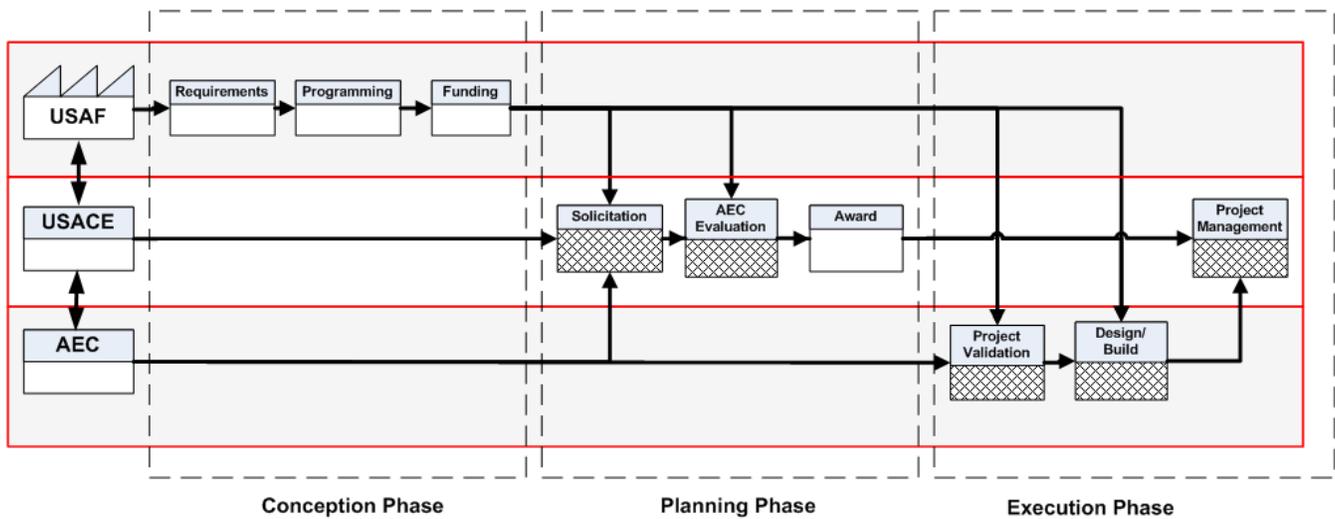


Figure 1. MILCON Process Flow and Areas of Control

### 3.2 Planning Phase

The Planning Phase includes the Solicitation, AEC Evaluation, and Award Steps. In this Phase, BIM is only used in the Solicitation and the AEC Evaluation steps.

*Solicitation:* Advertises the project and requests bids from interested AEC firms.

*AEC Evaluation:* Representatives from the USAF and the USACE evaluate bids from the AEC firms and selects the bid that is deemed the best value to the US Government.

*Award:* Announces which AEC firm was selected for the project. The AEC firm signs a contract with the Government, through the USACE, for the design and construction of the facility.

### 3.3 Execution Phase

The Execution Phase consists of Project Validation, Design-Build, and Project Management Steps. Extensive BIM use occurs in this phase of the MILCON process.

*Project Validation:* Details of the project and contract requirements are coordinated among the USAF, the USACE, and the AEC firm.

*Design-Build:* The facility design begins and certain type of construction, such as site preparation, may start. The design may go through several iterations before a final design is approved.

*Project Management:* USACE manages all aspects of the construction and coordinates with the USAF to resolve any issues.

## 4 Risk

The risk of using BIM in federal construction occurs at several key steps in the process. The risk is determined by the factors involved at these steps or by the entity controlling the steps or a combination of factors and control. The risk of compromise of the facility design can be due to the IT requirements of BIM within an agency or communication between agencies. Risk also arises from contractual requirements of the federal acquisition process, which requires communication of design information that introduces additional risk [3], [10]. The risk involved in using BIM in the MILCON process will vary for each phase of the MILCON process and will also vary from one agency to another. From a security perspective, the USAF and USACE primary concern is the compromise of the design and construction of the facility. However, the AEC firms may be more concerned with protecting their design and bid details from their competitors.

BIM is not involved in the MILCON process until interested AEC firms submit proposals for the construction project. From this point, the risks involved in using BIM begin. However, until the Government selects an AEC firm, the risk rests mainly on the AEC firm submitting the proposal. Personnel representing the Government must be aware of the risks presented by BIM use at this point and ensure security measures are in place. The risks associated with the steps under the USAF control are minimal since BIM is not used in these steps. The risk associated with the steps under the USACE and USAF is higher, because the Solicitation, AEC Review, and Award Steps require the use of BIM. After the contract is awarded, the risk factors become more significant since BIM becomes a major part of the construction process and is most critical at the design phase where it is used extensively.

Risk is an integration of threat, vulnerability, and consequence:

“Threat is a measure of the likelihood that a specific type of attack will be initiated against a specific target.

Vulnerability is a measure of the likelihood that various types of safeguards against threat scenarios will fail.

Consequence is the magnitude of the negative effects if the attack is successful” [11].

The Volpe Center represents this relationship with the following formula [11]:

$$Risk = Threat \times Vulnerability \times Consequence \quad (1)$$

Recent research has shown that there are limitations to this formula when considering terrorists attacks and that risk is a function of its three components [12]:

$$Risk = f(Threat, Vulnerability, Consequence) \quad (2)$$

However, the depth of research needed to study this concept

Table 1. Ratings for Risk and its Components

Possibility of Occurrence <i>Threat, Consequence, Vulnerability</i>			RISK <i>Threat + Vulnerability + Consequence</i>	
Possibility	Value	Rating	RISK Value	Risk Rating
Very Unlikely	1	Very Low	1 - 3	Very Low
Unlikely	2	Low	4 - 6	Low
Possible	3	Medium	7 - 9	Medium
Likely	4	High	10 - 12	High
Very Unlikely	5	Very High	13 - 15	Very High

is beyond the scope of this paper and the authors used a modified version of Formula (1):

$$Risk = Threat + Vulnerability + Consequence \quad (3)$$

The risk for each step in the MILCON process is evaluated using this formula where values are assigned to the probability of occurrence of the Threat, Vulnerability, and Consequence based on the Likert Scale [13]. The values and

Table 2. Risk associated with BIM use

RISK	MILCON Process Steps Using BIM				
	Solicitation	AEC Evaluation	Project Validation	Design-Build	Project Management
<b>A. Unauthorized Interception</b>	Threat = 2 → L Vulnerability = 2 → L Consequence = 2 → L  RISK = 6 → LOW	Threat = 2 → L Vulnerability = 3 → M Consequence = 3 → M  RISK = 8 → MEDIUM	Threat = 2 → L Vulnerability = 2 → L Consequence = 2 → L  RISK = 6 → LOW	Threat = 4 → H Vulnerability = 5 → VH Consequence = 4 → H  RISK = 13 → VERY HIGH	Threat = 4 → H Vulnerability = 4 → H Consequence = 2 → L  RISK = 10 → HIGH
<b>B. Unauthorized Distribution</b>	Threat = 2 → L Vulnerability = 2 → L Consequence = 2 → L  RISK = 6 → LOW	Threat = 2 → L Vulnerability = 3 → M Consequence = 3 → M  RISK = 8 → MEDIUM	Threat = 2 → L Vulnerability = 3 → M Consequence = 3 → M  RISK = 8 → MEDIUM	Threat = 4 → H Vulnerability = 5 → VH Consequence = 4 → H  RISK = 13 → VERY HIGH	Threat = 4 → H Vulnerability = 4 → H Consequence = 2 → L  RISK = 10 → HIGH
<b>C. Unauthorized Access</b>	Threat = 2 → L Vulnerability = 2 → L Consequence = 3 → M  RISK = 7 → MEDIUM	Threat = 2 → L Vulnerability = 2 → L Consequence = 3 → M  RISK = 7 → MEDIUM	Threat = 2 → L Vulnerability = 3 → M Consequence = 2 → L  RISK = 7 → MEDIUM	Threat = 4 → H Vulnerability = 4 → H Consequence = 4 → H  RISK = 12 → HIGH	Threat = 2 → L Vulnerability = 2 → L Consequence = 2 → L  RISK = 6 → LOW
<b>D. Unauthorized Alteration</b>	Threat = 3 → M Vulnerability = 3 → M Consequence = 4 → H  RISK = 10 → HIGH	Threat = 3 → M Vulnerability = 3 → M Consequence = 3 → M  RISK = 9 → MEDIUM	Threat = 2 → L Vulnerability = 3 → M Consequence = 2 → L  RISK = 7 → MEDIUM	Threat = 4 → H Vulnerability = 4 → H Consequence = 4 → H  RISK = 12 → HIGH	Threat = 2 → L Vulnerability = 2 → L Consequence = 2 → L  RISK = 6 → LOW
<b>E. Multiple Designs</b>	Threat = 3 → M Vulnerability = 3 → M Consequence = 4 → H  RISK = 10 → HIGH	Threat = 3 → M Vulnerability = 3 → M Consequence = 3 → M  RISK = 9 → MEDIUM	Threat = 2 → L Vulnerability = 3 → M Consequence = 2 → L  RISK = 7 → MEDIUM	Threat = 5 → VH Vulnerability = 5 → VH Consequence = 5 → VH  RISK = 15 → VERY HIGH	Threat = 5 → VH Vulnerability = 5 → VH Consequence = 5 → VH  RISK = 15 → VERY HIGH
<b>F. Server Compromise</b>	Threat = 4 → H Vulnerability = 4 → H Consequence = 5 → VH  RISK = 13 → VERY HIGH	Threat = 2 → L Vulnerability = 2 → L Consequence = 2 → L  RISK = 6 → LOW	Threat = 2 → L Vulnerability = 2 → L Consequence = 2 → L  RISK = 6 → LOW	Threat = 5 → VH Vulnerability = 4 → H Consequence = 5 → VH  RISK = 14 → VERY HIGH	Threat = 4 → H Vulnerability = 3 → M Consequence = 3 → L  RISK = 10 → HIGH
<b>G. Alteration Errors</b>	Threat = 3 → M Vulnerability = 3 → M Consequence = 4 → H  RISK = 10 → HIGH	Threat = 2 → L Vulnerability = 2 → L Consequence = 2 → L  RISK = 6 → LOW	Threat = 2 → L Vulnerability = 2 → L Consequence = 2 → L  RISK = 6 → LOW	Threat = 5 → VH Vulnerability = 5 → VH Consequence = 5 → VH  RISK = 15 → VERY HIGH	Threat = 3 → M Vulnerability = 3 → M Consequence = 3 → M  RISK = 9 → MEDIUM
<b>H. Management Errors</b>	Threat = 2 → L Vulnerability = 2 → L Consequence = 2 → L  RISK = 6 → LOW	Threat = 2 → L Vulnerability = 2 → L Consequence = 2 → L  RISK = 6 → LOW	Threat = 4 → H Vulnerability = 4 → H Consequence = 4 → H  RISK = 12 → HIGH	Threat = 5 → VH Vulnerability = 4 → H Consequence = 5 → VH  RISK = 14 → VERY HIGH	Threat = 4 → H Vulnerability = 4 → H Consequence = 5 → VH  RISK = 13 → VERY HIGH

ratings are shown in Table 1. The cumulative value obtained from the formula is then used to determine a value for the risk and its corresponding rating. These values and ratings are also based on Likert Scaling and are shown in Table 2.

#### 4.1 Interception of design

*Threat:* Since BIM exists in an electronic format, there is the possibility that it may be intercepted during transmittal.

*Vulnerability:* The use of electronic communication makes any transmission vulnerable to interception. The degree of vulnerability will depend on the network security of each user; the vulnerability will be based on the weakest security system.

*Consequence:* The consequence of interception can range from loss of confidentiality of the design and contract information to deliberate sabotage of the BIM model. These situations may arise from competition among contractors or acts by terrorist groups.

*Recommended Mitigation Measures:* AEC firms should employ electronic security protocols to reduce the chances of the design being intercepted. The communication link should be secured to deny unauthorized access to the BIM server. Personnel who evaluate bids must employ strict control over the BIM information to avoid compromising any bids.

#### 4.2 Unauthorized distribution

*Threat:* Unauthorized Distribution may allow a facility design to reach unintended or unauthorized people or groups.

*Vulnerability:* Distribution to unapproved or unknown parties may occur because of the ease of sending data electronically. This distribution may occur inadvertently or deliberately.

*Consequence:* Unauthorized distribution can result in the loss of confidentiality of design and contracting information. Since unauthorized or unintended recipients may not be identified, the facility information is deemed compromised.

*Recommended Mitigation Measures:* A distribution list of people authorized to send and receive BIM information should be created and updated periodically. The number of people on this list should be kept at a minimum and only include people who need to send and receive BIM information. The information itself should be encrypted so that in the event unauthorized parties receive it, they will not be able to access it.

#### 4.3 Unauthorized Access to BIM Design

*Threat:* Unauthorized access is a widespread threat, because it can occur anywhere the design exists and may result in unauthorized distribution and changes to the existing design.

*Vulnerability:* Unauthorized access to the design is possible if adequate security protocols are not in place. The degree of vulnerability depends on the security of the locations where the design information is kept.

*Consequence:* Unauthorized access can compromise the design and contracting information. Unauthorized distribution and design changes may occur and not be discovered.

*Recommended Mitigation Measures:* The layers of security that protect BIM information must be in place and evaluated frequently. Since all access points must meet the required security level, there should be security protocol agreement at the initial meeting of the USAF, USACE, and the AEC firm.

#### 4.4 Unauthorized Alteration of Design

*Threat:* Unknown or unauthorized changes in the design can occur without detection. This may be inadvertent or deliberate.

*Vulnerability:* Once there is access to the BIM information, anyone who has the knowledge can make changes to the BIM design. The degree of vulnerability is based on the vulnerability of access to the design.

*Consequence:* Unknown or unauthorized changes in the design can ultimately result in a design and facility that was not originally conceived or approved. However, any significant alteration will eventually be discovered so the effect will not be substantial.

*Recommended Mitigation Measures:* A design team should be designated. This team can produce read-only designs for people who do not need to make design changes. The design team should be the only body authorized to make changes to the design. Additionally, there should be a log to record who accessed the design and document any changes that were made. A backup system should be installed to memorialize several past designs in case changes need to be undone.

#### 4.5 Multiple Designs

*Threat:* Since BIM operates in a dynamic and multi-user environment, there will be multiple users making design alteration within their purview. This will result in multiple versions of the design and there will need to be a process to incorporate all changes and resolved any clashes.

*Vulnerability:* Multiple versions of the design is very common since there are multiple users contributing to the model. Designers will focus on their specialty leading to multiple version of the design.

*Consequence:* Each construction specialty will edit their parts of the design resulting in numerous versions. With multiple designs, there will be some degree of confusion and unnecessary work to incorporate all the different versions in

a single model. This can lead to construction clashes and the unnecessary performance of work based on the wrong design.

*Recommended Mitigation Measures:* Individual design changes must be discouraged or prevented. The project team must approve any changes to the original BIM design. Additionally, the person who makes the final decision and approves all changes must be identified. The project team should designate a design entity that is responsible for control the master copy of the BIM design. This entity should approve and perform any changes to the design. The legal status of who "owns" the design must also be resolved to prevent any litigation that may arise after construction [6]. The possibility exists that individuals who make changes to the design may stake some claim to the final design.

#### 4.6 Server Compromise

*Threat:* Since BIM exists and functions on an electronic platform, this platform must be networked to allow for collaboration and coordination of multiple users. The facility design is susceptible to compromise if there is uncontrolled access to this platform.

*Vulnerability:* In order for BIM to be dynamic and interactive, it must operate on a server to allow multiple users. With multiple users accessing the BIM design, the chance for compromise by alteration, distribution, or destruction increases.

*Consequence:* The compromise of the BIM server can result in the alteration and distribution of the facility design. The design can also be damage, destroyed, or deleted.

*Recommended Mitigation Measures:* The BIM server can be protected by having effective IT infrastructure and security protocols in place. Additionally, personnel must be trained and vetted to ensure they know how to operate the system and can be trusted.

#### 4.7 Alteration Errors

*Threat:* Revisions and edits presented by the three agencies may result in a design containing errors and may affect the construction of the facility.

*Vulnerability:* With a relatively large amount of people from three different agencies making or suggesting numerous changes and updates to the design, there is the possibility that errors may be included. These errors may go unnoticed since there are multiple people working on the design.

*Consequence:* Alteration errors will produce a facility design that contains flaws. If these inaccurate changes are not detected, the construction process and the actual facility or its composite sections may be also be flawed.

*Recommended Mitigation Measures:* The three agencies must designate a 'design team' to coordinate all alteration

and edits to the design. This team should be responsible for compiling, tracking, and performing all edits.

#### 4.8 Management Errors

*Threat:* With three separate agencies involved in federal construction, there is the possibility that there will be conflicting directions from different personnel. This can result in duplication of effect, errors in design, and possible legal claims.

*Vulnerability:* Numerous inputs from different agencies can result in errors. The more input there is the greater there is the chance for error.

*Consequence:* Management errors will result in design flaws and possibly flaws in the facility itself. Additionally, since there are legal contractual requirements for federal personnel, the AEC firm can file claims against the Government for following unauthorized directions.

*Recommended Mitigation Measures:* The project team must identify who the key decision-makers are and what their level of responsibility and authority is. These decision-makers should coordinate all requirements with their agencies before making presentations to the other agencies.

### 5 Risk and Cost

The amount of risk the project team is willing to accept affects the project cost. Figure 2 shows a relationship of risk planning and cost using three curves. Curves 1 and Curve 2 work in concert with each other. If a project team plans for a high degree of risk, initial cost for mitigation measures will be high and if a risk event occurs, there will be low consequential cost. Conversely, if a project team plans for a low degree of risk and a risk event occurs, there will be high consequential cost since fewer mitigation measures are in place. Curve 3 is the combination of costs from Curve 1 and Curve 2. The intersection of Curve 1 and Curve 2 lies within the best value region of total risk costs. Management Teams should balance Curve 3 with other project costs.



Figure 2. Relationship between Cost and Risk Planning [14]

This concept can be shown by considering Risk D - Unauthorized Alteration: an unwanted change to the design may go undetected and cause other aspects of the design to be flawed. Once the mistake is discovered, it will take additional man-hours and time to correct the flaw. The mitigation cost in this example will be to invest in backup systems to record the design at various stages and hire additional personnel to control design inputs. The consequential cost is the time and man-hours spent on tracing the change and correcting design errors caused by it. The project team must decide what risks and cost they are willing to accept.

## 6 Conclusion

The AEC have advocated the benefits of BIM and federal agencies have begun to formulate policies for its implementation. Federal construction will be impacted by this implementation since BIM design involves changing certain procedures that are used in current construction process using 2D design. While BIM has tremendous benefits in the production and collaboration of a facility design, there are some risks associated with its use. Some of the features that make it more efficient than conventional design are the same features that produce vulnerabilities. These risks are especially important for federal construction when critical infrastructure is involved and greater security is needed. The vulnerability in each step of the process that involves BIM must be analyzed to identify and mitigate any potential risk. The project team and the facility owner must decide how much risk they are willing to undertake and how much money they are willing to mitigate risks. They must balance these costs and risk throughout the project since each phase presents different levels of risk. On the surface, it may appear that the majority of risks associated with BIM seem IT related; however, the federal construction process has unique requirements that must be considered when using BIM. The authors understand that this paper is an initial assessment of the risks associated with BIM implementation; however, it provides a framework for further investigation by other federal agencies.

## 7 Disclaimer

The view expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government.

## 8 References

- [1] P. C. Suermann, "Evaluating the Impact of Building Information Modeling (BIM) On Construction", Dissertation, University of Florida, 2009.
- [2] Federal Business Opportunities, March 16, 2011 [Online]. <https://www.fbo.gov/>
- [3] United States Army Corps of Engineering, "The MILCON Project Process," Presentation, 13 May 2010, [Online]. <http://www.nan.usace.army.mil/sbc2010/pdf/presentation/milcon.pdf>
- [4] Information Assurance Technology Analysis Center, "Vulnerability Assessment, Fifth Edition," September 25, 2009.
- [5] C. Eastman, P. Teicholz, R. Sacks, and K. Liston, "BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors", Wiley, Hoboken, NJ 2008.
- [6] C. Furneaux and R. Kivvits, "BIM—Implications for Government," Cooperative Research Centre for Construction Innovation, Brisbane, Australia, 2008.
- [7] Office of the Under Secretary of Defense (Comptroller), "Construction Programs (C-1)" Department of. Defense Budget, Fiscal Year 2000 – 2011.
- [8] United States Air Force Center for Engineering and the Environment, "Design Instructions Memorandum, Attachment 7: AF BIM Guide Specifications," 10 Oct 2010, [Online] [http://www.wbdg.org/docs/afcee\\_attachf\\_bimrequire\\_db.doc](http://www.wbdg.org/docs/afcee_attachf_bimrequire_db.doc)
- [9] Public Law 107–296, "Homeland Security Act of 2002, H. R. 5005–11, Title III—Information Analysis and Infrastructure Protection," 25 November 2002
- [10] United States Air Force, "Designing and Constructing Military Construction Projects", Air Force Instruction 32-1023, 21 April 2010.
- [11] Volpe Center (2003) "Risk Assessment and Prioritization", [Online]. <http://www.volpe.dot.gov/infosrc/journal/2003/pdfs/chap1.pdf>.
- [12] L.A. Cox, "Some Limitation of 'Risk = Threat x Vulnerability x Consequence' for Risk Analysis of Terrorist Attacks," Risk Analysis, Society for Risk Analysis, 2008.
- [13] W.M.K. Trochim, "Likert Scaling," Research Methods Knowledge Base, 2006.
- [14] R.G. Bea, Modified 'Cost Benefit Curve,' Class Lecture, University of California, Berkeley, 2009.

## Towards a Low-Cost SCADA Test Bed: An Open-Source Platform for Hardware-in-the-Loop Simulation

Nicholas Wertzberger

(presenter)

University of Nebraska, Omaha

[nwertzberger@unomaha.edu](mailto:nwertzberger@unomaha.edu)

(402) 554-3979

6001 Dodge Street,  
Omaha, NE, 68182

Casey Glatter

University of Nebraska, Omaha

[cglatter@unomaha.edu](mailto:cglatter@unomaha.edu)

(402) 554-2072

6001 Dodge Street,  
Omaha, NE, 68182

William Mahoney

University of Nebraska, Omaha

[wmahoney@unomaha.edu](mailto:wmahoney@unomaha.edu)

(402) 554-3975.

[cs2.ist.unomaha.edu/~bmahoney/](http://cs2.ist.unomaha.edu/~bmahoney/)

PKI 282F

6001 Dodge Street,  
Omaha, NE, 68182

Robin Gandhi.

University of Nebraska, Omaha

[rgandhi@unomaha.edu](mailto:rgandhi@unomaha.edu)

(402) 554-3363

[faculty.ist.unomaha.edu/rgandhi/](http://faculty.ist.unomaha.edu/rgandhi/)

PKI 177A

6001 Dodge Street,  
Omaha, NE, 68182

Kenneth Dick.

University of Nebraska, Omaha.

[kdick@unomaha.edu](mailto:kdick@unomaha.edu)

(402) 554-4932

[cs.unomaha.edu/faculty/people/dick.html](http://cs.unomaha.edu/faculty/people/dick.html)

PKI 173D

6001 Dodge Street,  
Omaha, NE, 68182

### Abstract

*Cyber-physical systems are now ubiquitous. In the realm of critical infrastructure protection, it is important to have the capability to test and experiment on a simulated system so as to avoid potential failures which might have real catastrophic impact. Hardware-in-the-loop (HIL) simulation allows for the testing of target controllers with a simulated system in lieu of physical versions. But building these systems can cost anywhere from a few thousand dollars to over a hundred thousand dollars, based on the desired environment. In this paper, we present our work on an open and low-cost platform being developed as part of the Infrastructure Security Research Lab. We expect our research and development outcomes to be offered as a fully open-source option for HIL simulation.*

*Index Terms -- Hardware in the Loop, PLC Testing, Open Source, Critical Infrastructure, SCADA Simulation*

### 1. Introduction

Computational systems that monitor and control the critical infrastructure are significantly different from traditional Information Technology (IT) systems. Primary differences include visible cyber-physical sensing and actuation, real-time application needs, different system failure requirements, and hardware embedding of a large fraction of the system. Typical IT

labs in educational institutions do not provide opportunities to experience such system characteristics. In addition, critical infrastructure research and experimentation requires access to equipment which can provide a realistic interaction with control hardware and software, yet maintain human and environmental safety at all times. Developing dedicated test beds with equipment that spans different sectors of the critical infrastructure (e.g. water supply, power grid, oil pipelines, traffic control, sewage, etc.) can be a significant monetary investment.

We are faced with these issues at the critical infrastructure research lab in development at the University of Nebraska, Omaha. The primary design objective of the lab is to be able to switch between control system environments like one would expect from water utility infrastructure to an electric distribution grid to even a simulation of a large industrial factory floor. A dedicated “mock” environment is unacceptable because of the plasticity in lab configuration required to meet the needs of a variety of critical infrastructure sectors. In this paper we present our results from the investigation and use of a method for flexibly simulating critical hardware systems, called hardware-in-the-loop (HIL) simulation.

HIL is an effective cost-cutting measure for the design and testing of a wide variety of systems including engine control [1], automotive design [2,3], railroad infrastructure [4], and has been used for some time in the development of flight control systems [5].

The technique offers a way to test critical systems with a model instead of putting an untested system into the physical environment. It involves connecting control devices, which are often Programmable Logic Controllers (PLCs), with a data acquisition and control component. This model works well for simulating Supervisory Control and Data Acquisition (SCADA) systems, which traditionally are made of PLCs, and whose infrastructure is either too expensive to re-create or too dangerous to test directly. HIL components simulate portions of the environment which the SCADA systems are designed to control. Such simulations are commonly carried out using specialized test components developed by vendors such as National Instruments [6], which interface seamlessly with simulation environments authored in application packages such as LabView [7], Simulink [8] or others [9,10]. This setup allows for a detailed simulation to be created. However, these hardware and software components have significant initial setup and licensing costs, which may be prohibitive for educational labs. Furthermore, the level of detail allowed by these simulation systems may also be unnecessary in some situations. For example, a black box cyber security system assessment may not be concerned with how a system specifically functions, only that the functionality can be compromised.

A large fraction of a critical infrastructure test bed cost is devoted to acquiring acceptable test hardware. Efforts to develop economical solutions for critical infrastructure simulation that are engineered in software still require high cost hardware. Thus, test bed setup may stand to become considerably more economical through the use of an open hardware platform [10]. This need for a cost-effective configurable test bed motivates our work in the development of a new open-source system for HIL simulation. The developed solution is highly configurable and considerably lowers the initial costs associated with setting up a SCADA lab environment. This low cost allows for a dedicated simulation board for each PLC in the test environment. Ethernet control offered by our HIL solution can be used to communicate with a central server as well as integrate the test bed with existing IT lab equipment.

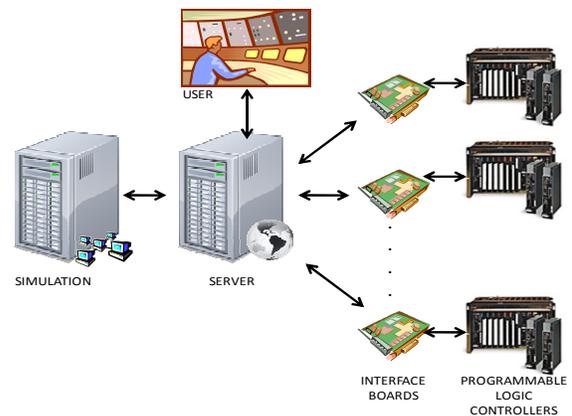
The rest of the paper is organized as follows. Section 2 provides a high level overview of the system. Section 3 delves into the interface hardware designed and the decisions made when creating it. Section 4 looks at the software for the interface boards, noting the open source software model used in its design. Section 5 talks about the control server, how it manages the connected interface boards, and the

interface provided into this server. Section 6 talks about J-Sim, and the reason for choosing this software as the first simulator backend. Section 7 talks about another application of the hardware interface board. Sections 8, 9, give a summary and discuss possible future work, respectively.

## 2. System overview

The HIL simulator system is made up of three parts: the interface board, the control server, and the simulator. These three subsystems form the complete integrated HIL simulation system shown in Figure 1.

The interface boards are responsible for interacting with PLCs. Each board must share a common ground with the device it is interacting with, and communicates with the control server through TCP/IP. On startup, a board obtains an IP address via DHCP and begins sending a message to a pre-determined server address every second until it is acknowledged. It then listens for instructions. A keepalive message is sent every four seconds to assure that the server is reachable.



**Figure 1. System block diagram**

The server is responsible for processing keepalive messages, sending update requests, receiving changes in PLC state, and translating these messages into meaningful data for the software simulator. All interactions of the server with the software simulation environment are via a network socket as shown in Figure 1. This allows the simulator to either exist on the same server or on another server, depending on processing requirements.

The communication between the server and the simulator follows a protocol where each simulation board connected can be enumerated, addressed, and controlled via JavaScript Object Notation (JSON) encoded strings. The example exchange depicted in Figure 2 demonstrates the addressing of all of the

available PLCs, and illustrates how each is instructed to start sending updates for any changes to hardware pin 0:

```
{action:"enumerate"}
{response: "ok", data: [101,102,103]}
{action:"notify-on-change",
 pins:[0], targets:[101,102,103]}
{response: "ok"}
```

**Figure 2: Simulator-controller exchange**

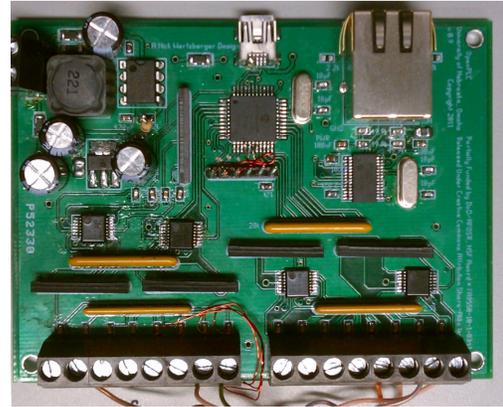
This format has been chosen due to its easy readability and the availability of libraries for supporting this protocol. The JSON-encoded string format also simplifies integration with anything from a simple script to a full simulator as a backend.

In our current lab setup, we rely on an Autonomous Component Architecture (ACA) based simulation engine called J-Sim [11]. This simulation paradigm allows the creation of a model at the desired level of abstraction for the phenomena under study. Such flexible simulation architecture is appropriate for most critical infrastructure security education, where the focus is more towards how to secure a system with some arbitrary level of granularity. ACA based simulations have been used successfully to study large scale network behaviors as well as simple contract-based interactions between self-contained components. However, ACA does have limitations. Since the simulation engine works entirely in software on a host operating system, its real-time assumptions are weak. For example, if the desired test depends on a highly time-critical operation, such as that in an anti-lock braking system, the current system would not be appropriate.

### 3. Hardware overview

The physical interface board, shown in Figure 3, is designed to interface directly to the hardware input/output (IO) area of various PLCs. The design is capable of interfacing with PLCs at voltages up to 24V and currents of 40mA. The devices in the lab which this board is designed to interface with are run off of a 24V power supply, making 24V the maximum voltage this board would have to interface within its target setting.

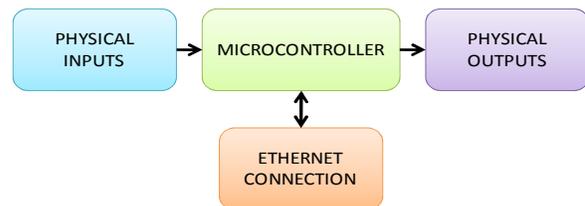
The hardware was designed with cost-effectiveness in mind while maintaining desired functionality. We decided the device must be Ethernet controllable, and capable of interacting with the hardware IO used on most major PLCs.



**Figure 3. The interface board (first version)**

Our design consists of a 3.3V control and communication module as well as a 6V to 24V physical input and output module. The block diagram for communication pathways in our system is shown in Figure 4.

The entire hardware base was designed using an open-source hardware design package called gEDA [12]. This ensures that any future changes to the board do not require the use of proprietary design software. In addition, the schematics were broken into five different sections: Ethernet, microcontroller, physical input, physical output, and power regulation. Because power regulation does not communicate with any module, it was left out of the diagram in Figure 4.



**Figure 4. Hardware Block Diagram**

The microcontroller module consists of a PIC18F4550 made by Microchip [13], including the necessary supporting hardware for USB communication. This chip has 2kB of RAM and 32kB of ROM, as well as 256 bytes of EEPROM, and is clocked at a frequency of 48MHz for a throughput of 12MIPS. Because of the low amount of RAM on this system, all software subsystems are designed with memory consumption as the first concern. The PIC18F4550 was chosen due to its low cost and built-in USB interface module.

The Ethernet Interface is made up of Microchip's ENC28J60 integrated MAC + PHY [14], which has an 8kB buffer for storing Ethernet frames, and supporting

hardware. This device can also generate certain parts of an Ethernet frame, such as the CRC, reducing the overhead for the main controller.

The physical input and output blocks use operational amplifiers and voltage dividers to translate between the desired voltage ranges. In order to simulate an arbitrary output voltage, several outputs were set up as class D amplifiers. This offered more versatility in the outputs, but also lowered how quickly the board can respond to stimuli.

#### 4. Software overview

The software for this device is designed with an attention to running in a memory constrained environment. The entire codebase is written in ANSI C89 and compiled with the SDCC compiler [15] in order to make it as portable as possible; future architecture changes may require other compilers.

The system is broken up into eight different software modules. The system is also layered with the intent that if someone did want to swap out the underlying hardware, only the bottom-most layers would be affected, allowing the code inside the *core* and *logic* modules to only be concerned with executing a defined behavior. The layout of the system is shown in Figure 5.

The *dhcpc*, *uip*, and *usbstack* modules represent existing open-source libraries which are integrated into the system. An effort was taken to keep the names the same as those used in the source. The *dhcpc* and *uip* modules are part of *uIP*, an open source TCP/IP stack designed for embedded applications [16], and the *usbstack* module is part of *Honken USB*, which was specifically designed for the chip we used [7]. Care has been taken to modify these modules as little as possible to allow easier updates in the future.

A technique called protothreading is used in *uIP* to lower the memory and processing overhead of the code used to parse and manage TCP/IP packets. This technique simulates multi-threading without the memory and processing overhead of a context switch. This is accomplished by storing the data needed by a function into a static data structure. The data structure keeps track of the last case statement executed in a large switch block that runs through the entire function. This implements user-level threads, and lowers the memory overhead usually required for a kernel-level thread implementation. Because of *uIP*'s use of this technique, some modules are designed for protothreading in order to fit with the rest of the system.

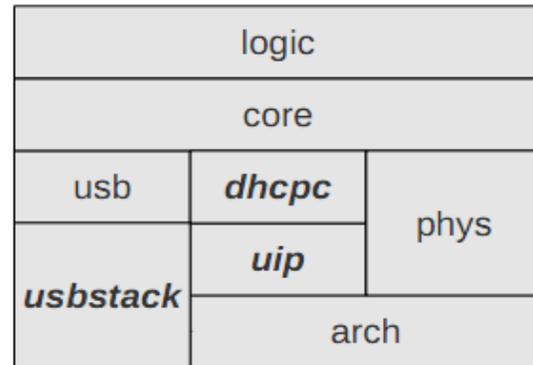


Figure 5. Software Block Diagram

The *arch* module is responsible for all hardware-level communication, with the exception of *usbstack*, which works directly with the PIC18F4550. This consists of any calls out to hardware IO or to the Ethernet MAC + PHY chip. In the event that a new microcontroller is used with this board, the *arch* and the *usbstack* modules should be the only ones that need to be rewritten.

The *usb* module offers a lightweight interface into the *Honken USB* library. It is there to assure that the open-source module being used for USB communication is easily replaceable.

The *core* module is the glue that connects the *logic* module to the rest of the system. The *logic* module is the control system for this stack. In the current setup, the *logic* module does little outside of converting commands into output voltages and converting input voltages into status updates, which are then sent to the server. However, in future branches of this system we hope to enhance this module in order to assure more real-time reactions to input.

#### 5. Control server overview

All interface boards are run by a single server, defined to be at address 201 on a class C subnet. This control server acts as a façade for discovering and controlling PLCs on a given subnet.

The control server offers a JSON-encoded string based interface, making communication easy to understand and debug. The server has support for the following commands: *enumerate*, *set-voltage*, *get-voltage*, *notify-on-change*, and *signal*. Every command other than *enumerate* acts on an array of PLCs. Such multicast commands allow a simulation to observe their influence on all included PLCs at once.

The *enumerate* command is used to determine the number of PLCs connected and their identifiers. If the

*enumerate* command is successful, an array is returned showing the list of PLC identifiers that are available.

The *set-voltage* command is used to set output voltages on specific pins. Pins 0 through 3 are capable of supplying analog voltages, while pins 4 through 7 can only supply digital output. If pins 4 through 7 are set to voltages greater than 0, they are considered ON.

The *get-voltage* command is used to poll devices for the input voltage on a set of pins. These are read as analog voltages for all input pins and are returned in the order the pins were enumerated.

The *notify-on-change* command is used to tell a PLC to send notifications to the server every time it senses a change in the given set of input pins. This is the preferred method for interfacing to a PLC, as it allows for immediate reaction and follows an event driven model.

The *signal* command is used to determine which physical PLC represents which identifier in the server. By sending a *signal* command to an interface board, the signal light will begin blinking.

The control server is loosely coupled with the simulator over a network socket. It is the responsibility of the server to either allow or deny incoming connections to the corresponding network port.

### 6. J-Sim simulator system

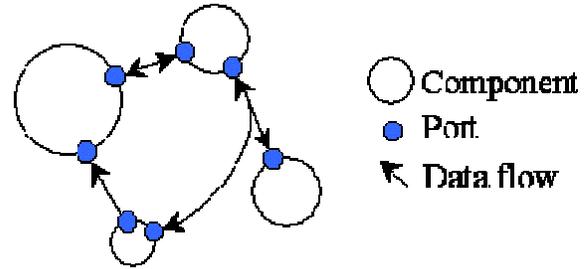


Figure 6: Component-based architecture [18]

J-Sim is a Java simulation environment that is based on Autonomous Component Architecture, or ACA [11]. J-Sim was chosen specifically for its ACA modeling paradigm and the ease of cyber-physical component creation and integration. At the heart of an ACA simulation model are its component specifications. A simulation model in J-Sim is represented as a composition of multiple autonomous “components” interfaced via “ports”. Any real-world system can be emulated by simply specifying components (and later implementing them) to represent actual objects such as a generator or PLC. The overall structure of an ACA system can be seen in Figure 6 [18]. The benefit of ACA is that these objects can be developed and tested independently of each other and integrated with the whole system later. This allows for

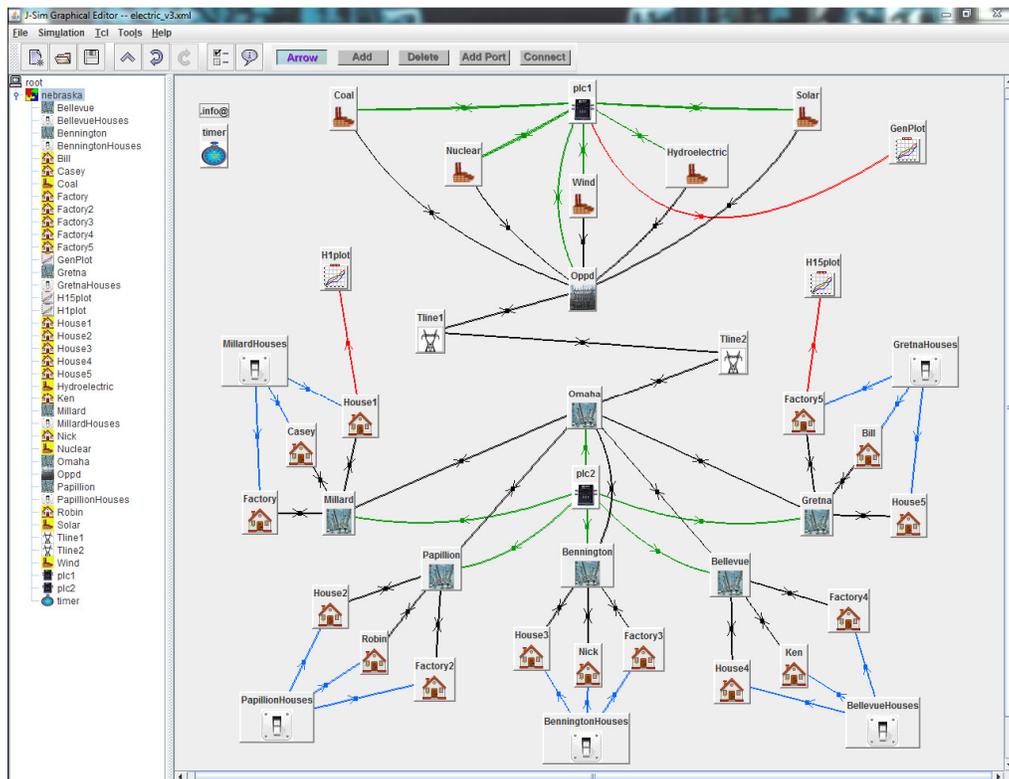


Figure 7: J-Sim power grid setup

specific component-based testing and debugging without having to worry about integration.

Communication between components is achieved through ports. Each component has its own ports through which it sends and receives data to or from the rest of the system. Since a component's ports belong only to itself, a component can be developed without the existence of any other components in the system. As long as the J-Sim specifications set up between the component and its ports are followed, the user does not have to worry about the other components or the mechanisms by which they communicate. For example, we have currently implemented a "mock" state-level electrical power grid simulation model, shown in Figure 7, and we have created a Transmission Line component which owns two ports, an input and an output. With ACA, the Transmission Line is only concerned with itself; that is, the component is essentially "asleep" until it receives data on its input port. At that point it performs some arbitrary computation on the data, sends the new data to its output port, and returns to an inactive state. This autonomous structure allows us to reuse a class of components to create multiple variations of a particular critical infrastructure model, such as an electrical power grid. This provides for many different penetration testing scenarios using the same basic set of components.

One of the components we have created for the electric grid simulation is the PLC. This component interfaces with the physical hardware PLC and reacts to messages sent by the HIL control server. In a typical attack scenario, a "malicious user" attempts to break into the control server through its web interface. After gaining access, the attacker uploads custom code to a connected real PLC. The changes that occur as a result of the new code are captured by the control server and forwarded to the simulation. The J-Sim PLC takes the changes and responds accordingly to the rest of the simulation model. Using both physical hardware and software simulation allows us to perform attacks on real PLCs and monitor the results in the simulation without the risk of damaging expensive physical electrical equipment.

## 7. An open-source programmable logic controller

The device designed and built to interface with PLCs is capable of reading and delivering standard voltage levels used by these systems. Furthermore, it is Ethernet controllable. In essence, this device has the capabilities of a low-end PLC at a very low price (\$50),

but with a more open design; this allows users to develop their own solutions that otherwise might have not been possible or were not cost effective.

The ability to use this platform for research on PLC design may lead to new applications for this technology. It may also lead to a more secure system design, as testing complicated systems has become far less costly. The low cost of these systems and lack of licensing overhead enables teachers to hand out a PLC that students can take home. The effect of this research does not need to stay specifically in HIL simulation.

## 8. Summary

HIL simulation is a powerful method for assessing the stability of critical control systems without the risk of testing a fully-live system. Mandatory testing on program upload for any PLC can help mitigate the risk of being infected with code designed to destroy critical infrastructure components.

The open-source HIL solution presented has the ability to integrate well with a basic lab setup and, due to its open design, is highly configurable. The system lowers the cost for HIL simulation by an order of magnitude and allows for a smoother and easier setup than one starting from scratch.

In order to be a functional HIL interface, the hardware we designed can also function as an actual PLC. This design is cost effective and useful in both academic and less critical environments. The possible effects of an open-source PLC are also much farther-reaching than what would be possible if only the HIL simulation aspect was focused on.

## 9. Future work

The current setup lends itself to numerous future improvements, including: supporting a more PLC-like functionality by supporting PLC programming languages and various communication protocols on the device, real-time responses for HIL simulation, and integration with various simulator back-ends.

### 9.1. Development as a PLC

Because this device holds much promise as a robust platform for experimentation with personal automation, support for easy program design and upload is a must. The current plan is to make the device web-programmable through the available Ethernet port. Through the use of a technique that allows scripts and data from outside sources to be used in a client browser, a very interactive, upgradeable UI can be

developed that also consumes little memory on the device.

Development of a secure communication protocol is also a desire. Research into real-time, low latency secure communication and validation solutions in the embedded space by using an elliptic curve key exchange combined with a symmetric encryption algorithm may better secure existing SCADA networks. It is a situation where the need to authenticate control messages comes at a head with the need to have fast response times. Due to the small packet size associated with these communications, we intend to first research the effectiveness of various linear feedback shift-register-based stream ciphers on protecting this information.

## 9.2. Improving SCADA penetration testing

Because this was designed for use in a security-centric setting, there could also be integration of an automatic fuzzer, which feeds erroneous data to a PLC in an effort to cause unexpected behavior. This would allow for the automation of attacks to which PLCs might be vulnerable.

In order to defend against less obvious vulnerabilities and detect subtle flaws, a more realistic simulator back-end is required. An interface into an open-source dynamic system modeler such as Scicos [19] or a similar software modeler could allow for more intricate environments to be set up.

## 10. Acknowledgements

This research was funded partially by the DoD-AFOSR, NSF Award #FA9550-10-1-0341.

## 11. References

- [1] R. Isermann, J. Schaffnita and S. Sinsela, "Hardware-in-the-loop simulation for the design and testing of engine-control systems", *Control Engineering Practice*, Volume 7, Issue 5, May 1999, Pages 643-653.
- [2] H. Hanselmann, "Hardware-in-the-Loop Simulation as a Standard Approach for Development, Customization, and Production Test", Paper Number: 930207, SAE International.
- [3] I. R. Kendall, and R. P. Jones, "An investigation into the use of hardware-in-the-loop simulation testing for automotive electronic control systems", *Control Engineering Practice*, Volume 7, Issue 11, November 1999, 1343-1356.
- [4] P. Terwiesch, T. Keller, E. Scheiben, "Rail vehicle control system integration testing using digital hardware-in-the-loop simulation", *IEEE Transactions on Control Systems Technology*, May 1999, Volume 7 Issue 3, 352 – 362.
- [5] D. Maclay, "Simulation gets into the loop", *IEE Review*, May 1997, Volume 43 Issue 3, 109 – 112.
- [6] National Instruments. "National Instruments – Test and Measurement". 2011. <http://www.ni.com>.
- [7] National Instruments. "NI Labview – Improving the Productivity of Engineers and Scientists". 2011. <http://www.ni.com/labview/>.
- [8] MathWorks. "Simulink – Simulation and Model-Based Design". 2011. <http://www.mathworks.com/products/simulink/>.
- [9] J. Wu, Y. Sheng, A. Srivastava, N. Schulz, H.L. Ginn. "Hardware in the Loop Test for Power System Modeling and Simulation". *Power Systems Conference and Exposition*, Atlanta, GA, 2006.
- [10] B. Lu, X. Wu, H. Figueroa, A. Monti. "A Low-Cost Real-Time Hardware-in-the-Loop Testing Approach of Power Electronics Controls". *IEEE Transactions on Industrial Electronics*, Vol. 54, No. 2, April 2007.
- [11] J. Hou. "J-Sim Home Page". 2005. <http://j-sim.cs.uiuc.edu/>.
- [12] "start [geda Wiki]". 2011. <http://geda.seul.org/wiki/>.
- [13] Microchip. "PIC18F4455". 2009. <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010293>.
- [14] Microchip. "ENC18J60". 2008. <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en022889>.
- [15] S. Dutta. "SDCC - Small Device C Compiler". 2010. <http://sdcc.sourceforge.net/>.
- [16] A. Dunkels. "Main Page – uIP". 2010. [http://www.sics.se/~adam/uip/index.php/Main\\_Page](http://www.sics.se/~adam/uip/index.php/Main_Page).
- [17] Dangerous Prototypes. "dangerous-prototypes-open-hardware" 2011. <http://code.google.com/p/dangerous-prototypes-open-hardware/source/browse/#svn%2Ftrunk%2FHonken%20USB%20stack%20CDC%20test>.
- [18] "Tutorial: Working With J-Sim." 2003. <http://sites.google.com/site/jsimofficial/j-sim-tutorial>.
- [19] T. Netter. "Scicos Homepage". 2009. <http://www-rocq.inria.fr/scicos>.

## An empirical study of a vulnerability metric aggregation method

Su Zhang, Xinming Ou  
Kansas State University  
Manhattan, KS, USA  
{zhangs84,xou}@ksu.edu

Anoop Singhal  
National Institute of Standards and Technology  
Gaithersburg, Maryland, USA  
psinghal@nist.gov

John Homer  
Abilene Christian University  
Abilene, Texas, USA  
jdh08a@acu.edu

**Abstract**—Quantifying security risk is an important and yet difficult task in enterprise network risk management, critical for proactive mission assurance. Even though metrics exist for individual vulnerabilities, there is currently no standard way of aggregating such metrics. We developed a quantitative model that can be used to aggregate vulnerability metrics in an enterprise network, with a sound computation model. Our model produces quantitative metrics that measure the likelihood that breaches can occur within a given network configuration, taking into consideration the effects of all possible interplays between vulnerabilities. In order to validate the effectiveness (scalability and accuracy) of this approach to realistic networks, we present the empirical study results of the approach on a number of system configurations. We use a real network as the test bed to demonstrate the utility of the approach, show that the sound computation model is crucial for interpreting the metric result.

**Keywords**-enterprise network security; attack graph; vulnerability metrics, quantitative risk assessment

### I. INTRODUCTION

Proactive security is an important part of mission assurance and critical-infrastructure protection. Metrics indicating inherent risk in a network system can help in prioritizing precious resources to improve security and reduce the possibility of mission interruption from successful cyber attacks. Quantifying a security level for large-scale networks has been a challenging work for a long time. System administrators currently have to act based on their experience rather than objective metrics and models. Much work has been done along the line of attack graph construction from network configuration in order to analyze network security [1], [2], [4], [5], [6], [7], [10], [11], [12], [13], [14], [15]. Attack graphs can show the cumulative effect of vulnerabilities throughout the network by visualizing the logical dependency between the attacker's initial position and the attacker's goal. However, attack graphs alone cannot tell how severe or dangerous these attack paths may be. CVSS metrics [8] have been developed to indicate certain security levels for each single vulnerability. Moreover, a number of works [2], [10], [16], [17], [18], [19] have attempted to compute the cumulative effect of vulnerabilities in a network quantitatively. In our prior work we proposed a sound approach [3] to aggregating vulnerability metrics in an enterprise network with a clear semantics on what

the calculated metrics mean: the likelihood a “dedicated attacker” can successfully penetrate the system and achieve certain privileges. Compared to previous works, this approach is based on a dependency attack graph which can be computed more efficiently [9], [11], and it correctly handles a number of important graphical structures such as shared dependencies and cycles to ensure the correctness of the result. In order to rigorously evaluate the effectiveness of this algorithm, we performed a series of empirical studies of this vulnerability metric aggregation method.

This empirical study is important because it can reveal both the effectiveness and limitations of a risk assessment method. Earlier work on security metrics has also performed substantial empirical evaluation [10] on production systems. Our method is based on a modern attack graph that has efficient computation, and it calculates the metric directly on a dependency attack graph without expanding it to a state attack graph which may incur an exponential blow up [6], [10]. To evaluate the benefit of such a method requires empirical evaluation on both its metric results and its running time on realistic systems. The empirical evaluation can also identify incorrect model assumptions or input parameters that make the result unrealistic and unusable, providing a feedback loop to calibrate the metric model.

There are a number of challenges of the empirical study. The configuration information of network is hard to obtain due to privacy or commercial reasons. It is also difficult to determine a set of proper parameters for the risk assessment approach. Implementation requires much work as well.

In order to address the aforementioned difficulties, we firstly talked with the system administrator and requested daily scanning results of certain machines in the Computing and Information Sciences Department at Kansas State University. We built a database to store related vulnerability information in NVD, where we retrieve the relevant information for each host's vulnerabilities. A parser is developed to construct input for MulVAL [11], [12], the attack graph tool used in our research, to generate attack graphs which were then used to calculate the security metrics based on the algorithms from our prior work [3].

## II. VULNERABILITY METRIC AGGREGATION METHOD

The MulVAL attack graph [11] is used as a structural basis for security metrics calculation, although our approach should be easily adapted to other attack graphs generators with similar semantics [4], [5].

### A. An example scenario

Figure 1 shows an example enterprise network, which will be used to illustrate a number of our security metrics algorithms.

**Reachability and Host:** There are three subnets and two firewalls (one internal and one external). The web server resides in DMZ which could be reached from Internet through the external firewall. The database server is in the internal subnet which contains sensitive information. It can only be reached through web server and the User subnet. The user workstations (used by normal employees) are all in the User subnet. All outbound connections from the User subnet are allowed by the external firewall.

**Vulnerabilities:** The web server has the vulnerability CVE-2006-3747 in the Apache HTTP service, which could be utilized by remote attackers to gain certain level of privilege to execute arbitrary code on the server. The database server has the vulnerability CVE-2009-2446 in the MySQL database service, by which attacker could gain administrator's privilege. The workstations contain the vulnerability CVE-2009-1918 in Internet Explorer; if an innocent user accessed malicious data through IE, the machine he is using will possibly be hacked. Usually, the system administrator would have limited time or energy to address all security issues. He would like to know which vulnerability is more dangerous or more urgent than others and deal with that first. Our approach could assist system administrators in this prioritization by offering quantitative metrics.

**Attack-graph semantics:** The lower part of Figure 1 is the MulVAL attack graph of the aforementioned network. Information about each node of the attack graph is found at the right side of the figure. A MulVAL attack graph has three types of nodes: (1) attack-step nodes, represented within the graph as circular-shaped AND-nodes. Each AND-node indicates a step of attack which could happen when all preconditions (either configuration nodes or privilege nodes) are held; (2) privilege nodes, represented within the graph as diamond-shaped OR-nodes. Each privilege node stands for a certain level of privilege which could be derived from any one of its predecessors (AND-nodes); (3) configuration nodes, which are not shown in this graph. Each configuration node represents a configuration condition of the network. For example, the network connections or vulnerability properties are all included in configuration nodes. As one example attack path, the attack graph shows that an attacker could first compromise the web server and then use it as an

intermediate stop for his next step of attack on the database server (0-28-8-7-6-4-3-2-1).

**Component metrics:** The input of our metric model are component metrics, used as an indicator of each attack-step's likelihood of success. A number of these metrics are constructed based on CVSS metrics like Access Complexity (AC). The metrics can be regarded as conditional probabilities while all the preconditions for exploiting the vulnerability are satisfied. For example, if the vulnerability is hard to access (with a high value on AC), then even if all the preconditions are met, it will still have a small chance of being successfully utilized by attackers.

**Assumptions in the metric model:** We assume that an attacker knows the complete picture of the network, including network reachability, services running, and vulnerability information in applications. In other words, the adversary possesses the complete information in the attack graph. Further, we assume that the attacker will try all possible ways in the attack graph to compromise the system. In other words, the prior probability that an attack will be attempted is assumed to be one.

The output of our metric model is the likelihood of being hacked for individual machines. The major challenge in calculating the metrics with the above semantics is shared dependency and cycles in attack graphs. We developed techniques [3] to overcome these difficulties in our past work.

## III. EXPERIMENT STRATEGY

To prepare the experiments, we performed a number of preliminary tasks.

- **Data Collection/Scanning**

We scan seven windows servers running Windows Server 2003 at the CIS department of Kansas State University, by using the reference interpreter<sup>1</sup> of OVAL<sup>2</sup>. We have implemented a cron job to perform the scan daily and send the reports (XML files generated by OVAL) to a central repository.

- **Database Setup**

In order to speed up the processing of data, we first extract all useful information from the NVD<sup>3</sup> data feeds (XML files including information of all vulnerabilities) into a separate mysql database, and build a tuple for each vulnerability in the database. The key of the tuple is the CVE ID, and other elements include CVSS metrics, attack range of the vulnerability (either remote service, local or remote client), consequences (compromising confidentiality, integrity or availability), and so on.

<sup>1</sup><http://oval.mitre.org/language/interpreter.html>

<sup>2</sup><http://oval.mitre.org/>

<sup>3</sup>National Vulnerability Database, <http://nvd.nist.gov/>

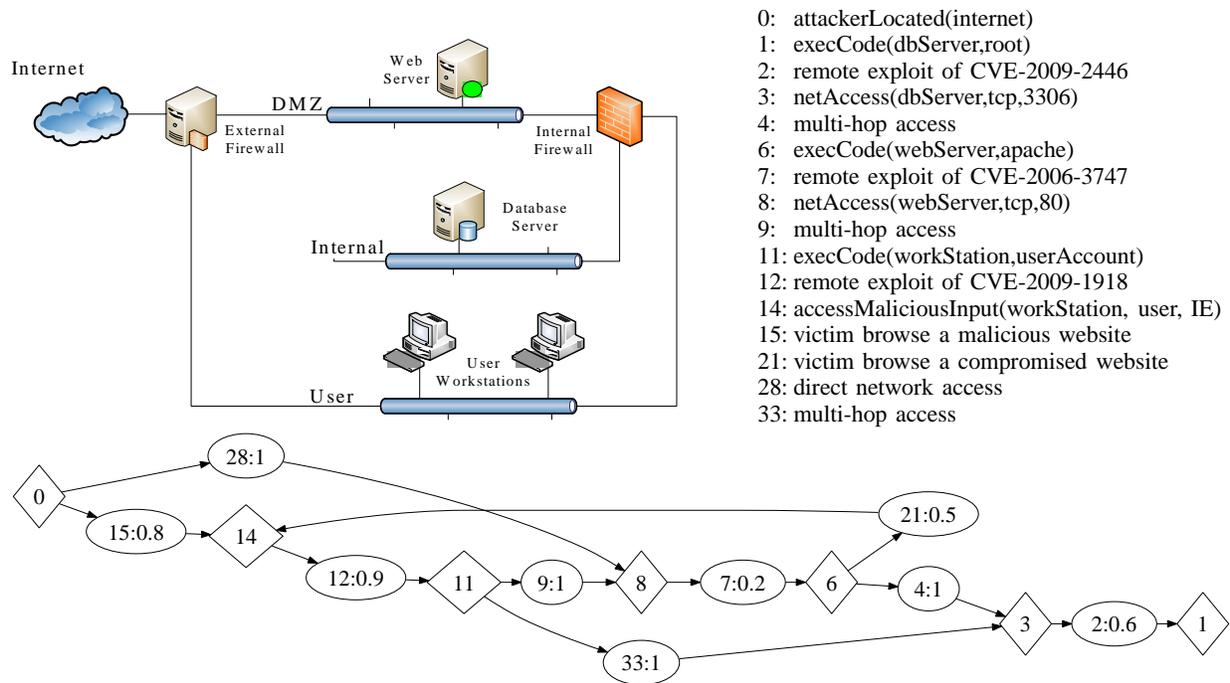


Figure 1. Example scenario and attack graph

• **Data Parsing/Input Construction**

We then construct input data for the MuVAL attack-graph generator. For each machine, we parse its scanning report and obtain the CVE IDs of the vulnerabilities on this machine. By using these CVE IDs, we extract other information about the vulnerabilities through the database we built in the previous step. Meanwhile, we construct input for attack-graph generator based on the extracted information.

After we finish constructing the input data for risk assessment, we run our attack-graph generator and risk-assessment algorithms. Figure 2 illustrates the data flow for the empirical study.

We conducted two lines of experiments:

• **Empirical study on each single host.**

We did experiments on each single host without considering the multi-host attack by assuming there is a direct connection between the attacker and each host. We then compare the security level of different hosts and present the result to the system administrator for verification.

• **The previous experiment repeated over time.**

To observe the security metrics change trend over time, we did a number of experiments for each host at different points of time. We then analyze the detailed information returned from the vulnerability scan to confirm whether the risk trend indicated by metrics makes sense.

MuVAL attack-graph tool-chain

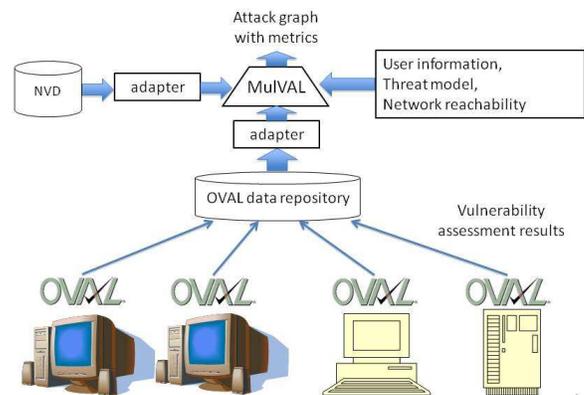


Figure 2. MuVAL attack-graph tool-chain

IV. EXPERIMENTATION RESULT

When we saw the first batch of risk assessment results from the production systems, it quickly became clear that the original graphical model is insufficient to capture some hidden correlations important in gauging the risk levels. Thus we introduced additional modeling artifacts to capture them for the subsequent experiments.

*Modeling artifacts for capturing hidden correlations:*

Figure 3 indicates two attack graphs with security metrics of two servers. From the two attack graphs we can tell that server (a) has many more vulnerabilities than server (b). This could be easily observed from the difference of the

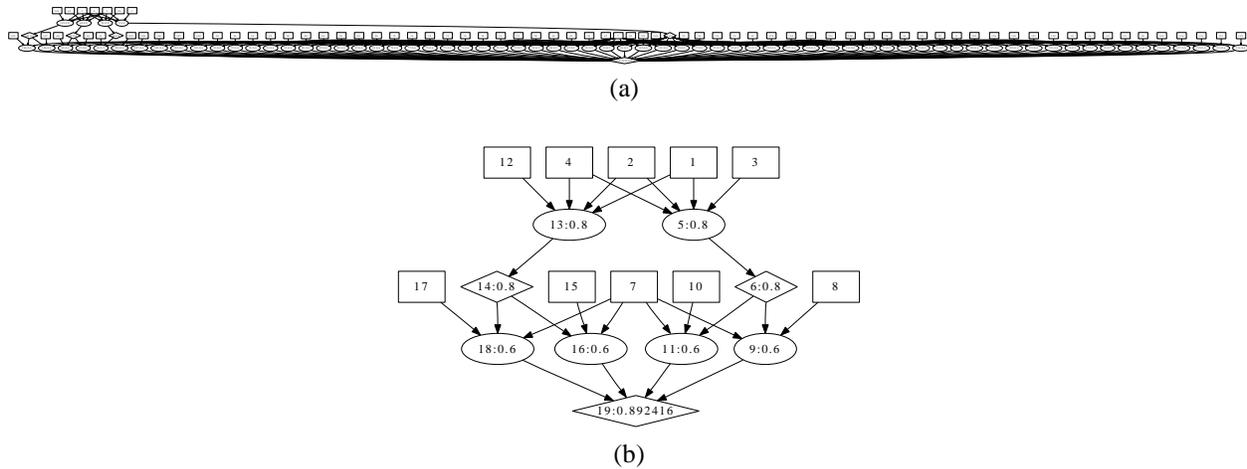


Figure 3. Attack graphs from two production servers

density of the two attack graphs (one of them is so dense that it appears almost like a line in the limited paper space). However, the difference of the cumulative security metrics of the two servers are not that obvious: server (a) is 0.99 while server (b) is 0.89. As we noticed, all the other servers in our department have more vulnerabilities than (b), meaning the security metrics are always close to 1. Intuitively, a question needs to be asked: is this the true reflection of these machines security situation? We consulted our system administrator and he thought that intrinsic differences among the vulnerabilities is an important factor while gauging the risk. If a host has one application with ten vulnerabilities, it will have lower security risk than one with ten different applications with one vulnerability in each. If an attacker is not familiar with a specific application, then even if this application has ten vulnerabilities, he still has a very low chance of utilizing these security holes. However, if there are ten applications each with one vulnerability, and if the attacker happens to know one of these applications well, he would have a much higher chance of compromising the machine successfully. The attack graph did not sufficiently capture the dependency between vulnerabilities and applications. Server (a) has 67 vulnerabilities in four applications while server (b) has only four vulnerabilities in two applications. Therefore, the metric calculated for server (a) should be much higher than server (b), which is not the case for the results from the original model.

Another hidden correlation arises in multi-stage attacks. Suppose an attacker just compromised one machine through a vulnerability from application A. If his subsequent targets have the same or similar vulnerability also from application A, he would have a very high chance of success since he should have known the underlying structure of application A very well. Suppose that exploiting a vulnerability has a 0.6 success likelihood. Based on our current attack-graph, a two-step attack utilizing the same vulnerability would give

the attacker 0.36 (multiplied by two 0.6) chance of success. However, the experience of hacking the first target would lead to a much higher success possibility of the second step (almost 1). Therefore, the metrics would be close to 0.6 rather than 0.36, if we account for such hidden correlations.

We created additional modeling artifacts in our graphical model used in calculating the metrics, in order to capture these hidden correlations. The vulnerabilities belonging to the same application are grouped into one node representing a successfully exploitation on any of them. The access complexity metrics of the grouped vulnerability is equal to the lowest value from the vulnerabilities in the group. This schema not only applies to single machine (Figure 4.a) but also to multiple hosts (Figure 4.b) to capture the hidden correlations among multiple hosts. The likelihood of successfully exploiting at least one of the vulnerabilities within the same group is associated with the virtual node  $A_V$ , which is the parent of the original exploit nodes. This way the hidden correlation is correctly captured. In (a), an attacker success (failure) in exploiting  $A_V$  is equivalent to success (failure) in hacking  $A_1, \dots, A_4$ . The schema rectified the previous distorted metrics (which is extremely close to 1) by grouping similar vulnerabilities (*i.e.*  $A_1, \dots, A_4$ ). In (b), if an attacker managed the expertise of hacking through  $A_2$ , it will succeed in  $A_4$  as well since the two involve the same vulnerability.

Returning to the servers modeled in Figure 3, when running the algorithms on the extended graphical model, the difference in the calculated metrics for the two hosts widened: 0.98 vs 0.73. This difference conforms to our intuitive assessment based on system administrator's feedback. Also, the number of vulnerable applications (same as number of grouped vulnerabilities) becomes the number of exploits instead of number of vulnerabilities, reducing the attack-graph size as well (see Figure 5).

The calculated metrics of the two hosts are still very high

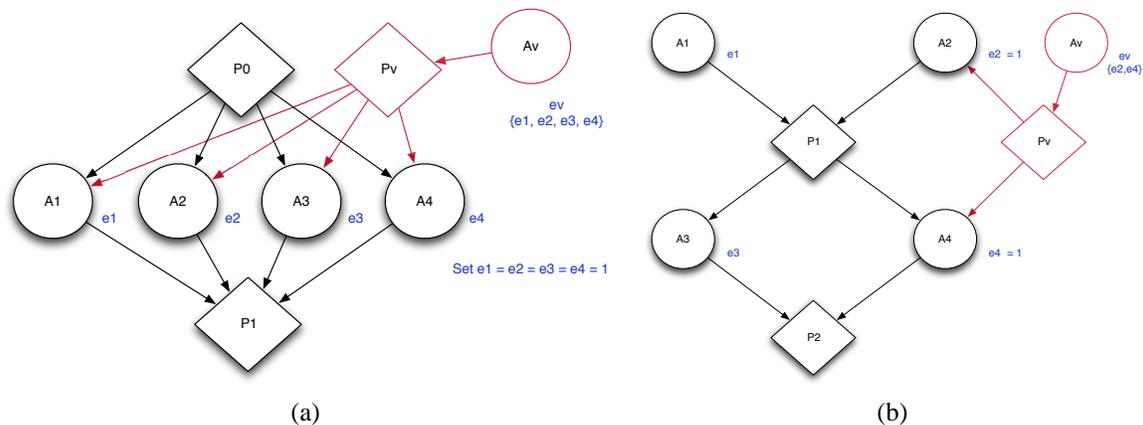


Figure 4. Modeling artifacts for capturing hidden correlations

which means there are vulnerabilities existing on the two hosts need to be patched. After we reviewed the attack graph, we realized that most of the vulnerabilities are client side and would need to be triggered through user actions. However, most of these applications have a low probability of being invoked by users, due to the functionalities of these servers, whereas we assigned a likelihood of 0.8 for all of them. This indicates that assigning such component metrics based on context is necessary in order to make the measured metrics reflect the true security situation of the network.

#### A. Experiment on individual machines

For this experiment, we run our risk-assessment algorithms against several different machines of CIS department at Kansas State University. The evaluating results indicate the security levels for these different machines. The departmental network has a fairly simple network topology. We assume all machines are directly connected to the Internet (where attacker located) and without considering multi-host attacks. For this configuration, the functions of servers and their cumulative metrics results are shown in Table I. In the table, the numbers indicate the likelihood various machines can be successfully compromised by an attacker. In order to justify the differences between the metrics, we reviewed their scanning reports. For example, on the report of November 4th 2010, machine1 is safer than machine4 in terms of the metrics (0.52 vs 0.816) with normal user privilege. After reviewing the scanning reports of the two machines, we are assured that our calculated metrics conform to the security level of the machines. For example, machine1 has two groups of service vulnerabilities (both under services with normal user privilege from the Windows system). Attackers could have two different major paths to exploit it. One representative is CVE-2010-3139 (exists in a number of Windows systems) and the other is CVE-2010-0820 (in Windows 7 only). Therefore the attacker could launch attacks either through certain libraries insecurely loaded by Windows Progman Group Converter (CVE-2010-

3139) if he knew the user is using a generic Windows operating system. Or the attacker could utilize malformed LDAP messages (CVE-2010-0820) if he knew the victim machine is running Windows 7. Both vulnerabilities are fairly easy (with low Access Complexity metrics). Therefore, for machine1, the attacker has two easily accessible and independent paths to compromise it. As for machine4, not only it has the aforementioned two vulnerabilities in machine1, but also has two other user-privilege service security holes that could be utilized by attackers. One is CVE-2009-3959, Acrobat 9.x (before 9.3) allowing attackers remotely execute arbitrary code via a malformed PDF document easily (AC is low). The other is CVE-2009-4764, where an attacker could execute the EXE files embedded at the pdf files through Adobe reader remotely with reasonable amount of cost/effort (AC is medium). Therefore by having two additional independent attack paths, machine4 has a higher risk metric than machine1 with normal user privilege. Besides, machine4 has one local vulnerability CVE-2010-3959 (while machine1 does not) which could be used by attackers to further escalate their privilege from normal user to root through a crafted CMAP table in an OpenType font. Thus an attacker could not compromise machine1 with a root privilege but can gain the administrative privilege on machine4 with a likelihood of 0.539.

#### B. Experiments over time series

In order to observe the trend of security levels over time through our approach, we did the same experiment on the individual machines at different points of varying, created MulVAL input files representing network of each time spot, and evaluated them with the current implementation of our algorithm. We carefully reviewed vulnerability scanning reports for all the hosts we used in our experiments. The trend of the machines' security levels conform to our metrics. The change could be either an increase or decrease of vulnerability number or change of CVSS vectors. For example, machine6 has three grouped service

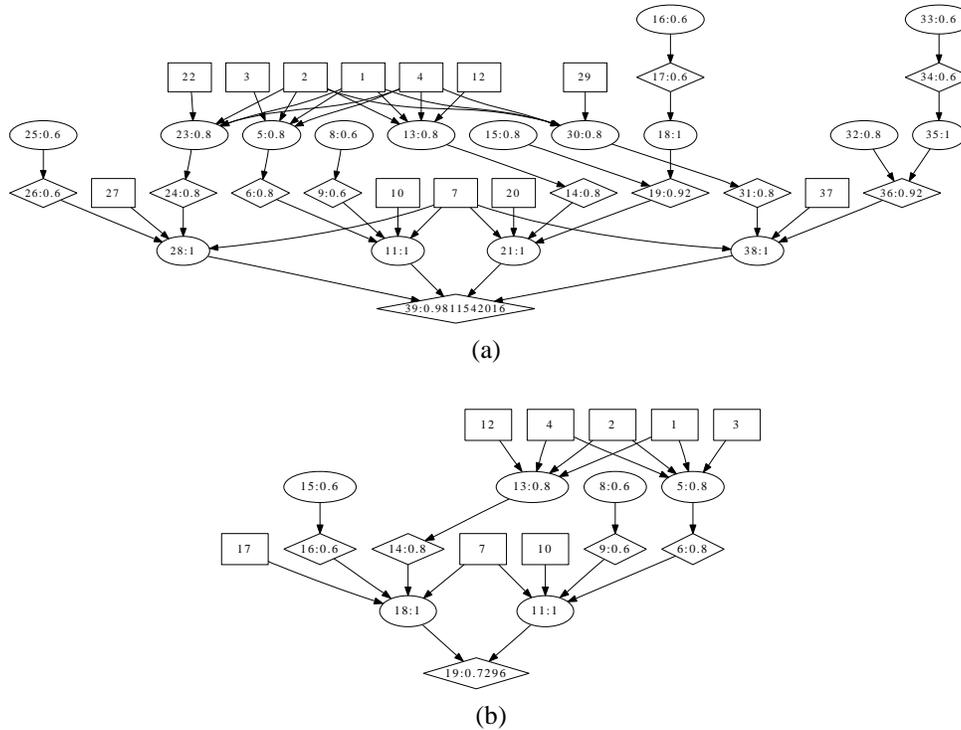


Figure 5. Attack graphs with modeling artifacts for capturing hidden correlations

Host	Function	Individual metric over time					
		11/04/2010		12/19/2010		02/17/2011	
		user	root	user	root	user	root
machine1	Printing	0.52	0.0	0.52	0.0	0.52	0.27
machine3	Scanning	0.853	0.054	0.853	0.054	0.853	0.054
machine2	Camera video collection	0.988	0.028	0.988	0.028	0.988	0.028
machine4	DeepFreeze	0.816	0.539	0.816	0.539	0.816	0.280
machine5	Active Directory Mirror	0.958	0.141	0.958	0.141	0.958	0.141
machine6	Camtasia Relay	0.616	0	0.616	0	0.616	0.32
machine7	DNS/Active Directory	0.992	0.028	0.994	0.028	0.994	0.028

Table I  
PROBABILITY OF COMPROMISE FOR INDIVIDUAL MACHINES OVER TIME

vulnerabilities. One is on the general Windows framework. Among the vulnerabilities grouped into this one, the lowest Access Complexity (AC) level is medium. Another group of vulnerabilities fall into Windows 7, the lowest AC of which is low. The rest of the service vulnerabilities belongs to IIS, the easiest accessible level of these vulnerabilities is medium. While taking CVSS Access Complexity metrics into consideration, and based on our attack graph rules, we can derive the facts that attacker could execute arbitrary code as a normal user with probability 0.616. We did our experiments over three time spots: November 4th 2010, December 19th 2010 and February 17th 2011. We found that most of the metrics for attacker executing arbitrary code as a normal user did not change because the number and the Access Complexity of the vulnerabilities (service vulnerabilities running with user privilege) did not change.

On the other hand, the metric for attacker running arbitrary code on machine6 as an administrator rose from 0 (in November and December) to 0.32 (in February). This change is attributable to a local exploitable vulnerability detected in February. Since the attacker has a certain chance (0.616) of executing arbitrary code as a normal user, along with the local exploitable vulnerability, he could escalate his privilege to root with probability 0.32. The Access Complexity of this group of vulnerabilities is low. Similarly, for machine1, there are two grouped vulnerabilities from November to February and all of the services are running under normal user privilege. Therefore the attacker has the same set of attack paths to compromise the host with normal user's privilege. There is only the one local exploitable vulnerability, first detected in February. The attacker could have one more attack path to compromise the machine with root privilege;

thus the risk metrics for machine1 root privilege is raised from 0 to 0.27. See table I for all the results.

## V. CONCLUSION

We have presented an empirical study of a vulnerability metrics aggregation approach. The approach is sound in that, given component metrics which characterize the likelihood that individual vulnerabilities can be successfully exploited, the model computes a numeric value representing the cumulative likelihood for an attacker to succeed in gaining a specific privilege or carrying out an attack in the network. We confirmed the metric model's effectiveness by evaluating it on a number of servers in a departmental network. By analyzing the security level trend over time, we conclude that the metrics computed by our approach conformed to the real security situation change (*i.e.* increase or decrease of vulnerabilities or a change of a vulnerability's severity) of the scanned machines.

## VI. ACKNOWLEDGMENT

This material is based upon work supported by U.S. National Science Foundation under grant no. 1038366 and 1018703, AFOSR under Award No. FA9550-09-1-0138, and HP Labs Innovation Research Program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or Hewlett-Packard Development Company, L.P.

## REFERENCES

- [1] Paul Ammann, Duminda Wijesekera, and Saket Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of 9th ACM Conference on Computer and Communications Security*, Washington, DC, November 2002.
- [2] Marc Dacier, Yves Deswarte, and Mohamed Kaâniche. Models and tools for quantitative assessment of operational security. In *IFIP SEC*, 1996.
- [3] John Homer, Xinming Ou, and David Schmidt. A sound and practical approach to quantifying security risk in enterprise networks. Technical report, Kansas State University, 2009.
- [4] Kyle Ingols, Richard Lippmann, and Keith Piwowarski. Practical attack graph generation for network defense. In *22nd Annual Computer Security Applications Conference (ACSAC)*, Miami Beach, Florida, December 2006.
- [5] Sushil Jajodia and Steven Noel. Advanced cyber attack modeling analysis and visualization. Technical Report AFRL-RI-RS-TR-2010-078, Air Force Research Laboratory, March 2010.
- [6] Richard Lippmann and Kyle W. Ingols. An annotated review of past papers on attack graphs. Technical report, MIT Lincoln Laboratory, March 2005.
- [7] Richard P. Lippmann, Kyle W. Ingols, Chris Scott, Keith Piwowarski, Kendra Kratkiewicz, Michael Artz, and Robert Cunningham. Evaluating and strengthening enterprise network security using attack graphs. Technical Report ESC-TR-2005-064, MIT Lincoln Laboratory, October 2005.
- [8] Peter Mell, Karen Scarfone, and Sasha Romanosky. *A Complete Guide to the Common Vulnerability Scoring System Version 2.0*. Forum of Incident Response and Security Teams (FIRST), June 2007.
- [9] Steven Noel and Sushil Jajodia. Managing attack graph complexity through visual hierarchical aggregation. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 109–118, New York, NY, USA, 2004. ACM Press.
- [10] Rodolphe Ortalo, Yves Deswarte, and Mohamed Kaâniche. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Transactions on Software Engineering*, 25(5), 1999.
- [11] Xinming Ou, Wayne F. Boyer, and Miles A. McQueen. A scalable approach to attack graph generation. In *13th ACM Conference on Computer and Communications Security (CCS)*, pages 336–345, 2006.
- [12] Xinming Ou, Sudhakar Govindavajhala, and Andrew W. Appel. MulVAL: A logic-based network security analyzer. In *14th USENIX Security Symposium*, 2005.
- [13] Cynthia Phillips and Laura Painton Swiler. A graph-based system for network-vulnerability analysis. In *NSPW '98: Proceedings of the 1998 workshop on New security paradigms*, pages 71–79. ACM Press, 1998.
- [14] Oleg Sheyner, Joshua Haines, Somesh Jha, Richard Lippmann, and Jeannette M. Wing. Automated generation and analysis of attack graphs. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 254–265, 2002.
- [15] Laura P. Swiler, Cynthia Phillips, David Ellis, and Stefan Chakerian. Computer-attack graph generation tool. In *DARPA Information Survivability Conference and Exposition (DISCEX II'01)*, volume 2, June 2001.
- [16] Lingyu Wang, Tania Islam, Tao Long, Anoop Singhal, and Sushil Jajodia. An attack graph-based probabilistic security metric. In *Proceedings of The 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSEC'08)*, 2008.
- [17] Lingyu Wang, Sushil Jajodia, Anoop Singhal, and Steven Noel. k-zero day safety: Measuring the security risk of networks against unknown attacks. In *Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS 10)*, Athens, Greece, September 2010. Springer Verlag.
- [18] Lingyu Wang, Anoop Singhal, and Sushil Jajodia. Measuring network security using attack graphs. In *Third Workshop on Quality of Protection (QoP)*, 2007.
- [19] Su Zhang, Doina Caragea, and Xinming Ou. An empirical study of using the national vulnerability database to predict software vulnerabilities (submitted). 2011.

# A Method to Determine Superior QoS Configurations for Mission Objectives: Aligning the Network with the Mission

Vinod Naga<sup>1</sup>, John Colombi<sup>1</sup>, Michael Grimaila<sup>1</sup> and Kenneth Hopkinson<sup>2</sup>

<sup>1</sup>Department of Systems and Engineering Management

<sup>2</sup>Department of Electrical and Computer Engineering

Air Force Institute of Technology

Wright-Patterson Air Force Base, OH, 45433, USA

{vinod.naga, john.colombi, michael.grimaila, kenneth.hopkinson}@afit.edu

**Abstract** - Across the system-of-systems, network components must be configured and aligned to organizational objectives, missions and tasks to maintain best performance. This alignment is especially critical when resources are inadequate to meet all the demand. A method is presented to find and promote the alignment by conducting an iterative heuristic search to identify a superior Quality of Service (QoS) configuration for a combination of mission and network requirements. Results using OPNET Discrete Event Simulation confirm the method which enables maintaining a consistent network service value under increasing traffic conditions.

**Index Terms** – Mission Assurance, Quality of Service (QoS), Network Management, System of Systems, Systems Engineering

## I. INTRODUCTION

Organizations often operate according to a strategy designed to achieve a set of goals. The leadership must communicate the strategy as well as the goals to the members of the organization through a series of objectives supported by guidelines and divided temporally into phases. The UML association diagram in Fig. 1 describes a generalized relation-

ship between these defined information elements. Each sub-organization accepts the assigned objectives and then details sets of missions which together are designed to achieve the objective through a series of tasks. Progress of missions towards the objectives is measured against predetermined performance expectations often stated as requirements. Network policy must follow organizational policy. Functional analysis enables alignment of missions and tasks to physical network resources. Translating goals and strategy into a network language also carries benefits. This paper presents a method to improve the network configuration by aligning it to mission objectives through related network performance goals. The value of this method is then demonstrated with an experiment.

Aligning these resources permits the managing of all goal-related operations to maintain an overall high level of performance. When necessary resources are lacking they may be shared, tasks and missions may be prioritized and tasks and activities smartly eliminated with the overall objectives in mind. Operations may be improved by aligning both network resources and the information handled by the resources with overall goals, objectives and missions.

The network managers may estimate the network state but increases in load, unexpected communication dropouts and attempts by outside parties to compromise network reliability are not known beforehand. Network state uncertainty makes it hard to design the network with adequate capacity and capa-

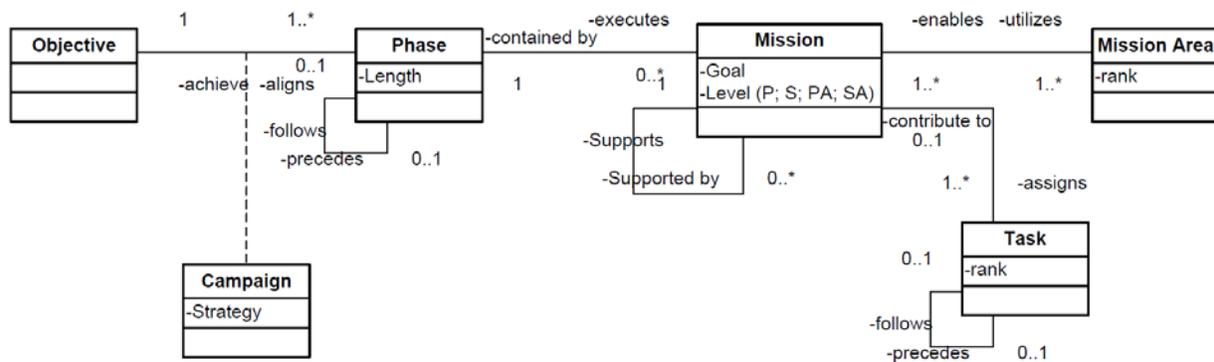


Fig. 1 UML Association Diagram Outlines Objective, Mission, Task Relationships.

bility for all these situations. However, the network designed to actively apply mission importance can better handle changes to network state. Quality of Service (QoS) routines are available both within and between network domains to manage the distribution of resources on those domains. An alignment of these QoS routines requires communication of objectives and mission but offers the benefit of improved network performance towards overall organizational goals.

For networks, service level agreements (SLA) and service level specifications (SLS) capture the requirements which are important to organizational function [1]. The functions of the organization and its components define the information exchange requirements (IER). These IERs, when levied on network resources, become the SLS [2].

Time-value functions and value analysis are methods by which network performance may be measured [3] [4]. Value across the system-of-systems is built and assessed using Analytical Hierarchy Process (AHP), a technique related to Objectives Hierarchy or Value Focused Thinking [5] [6]. The overall goal of realizing the best network service level for the given mission is decomposed into subgoals and criteria.

Five subgoals are a mix of mission and network aspects, such as:

- (1) Tailor network to current mission,
- (2) Tailor network to all missions,
- (3) Focus network for dynamic links,
- (4) Maintain some network reserve capacity, and
- (5) Deliver superior performance to all applications/flows.

An evaluation method composed of these subgoals is provided in this paper and visually depicted in Fig 3.

A series of experiments demonstrate the value of the organization policy to QoS linkage in designing the critical factors for network management. A network's service value to a mission is assessed via an AHP-derived objective function based on observations and measured performance. Network simulation tools approximate network activity to evaluate the network service value. By applying the service value in an iterative heuristic search, a network which serves the mission with a superior QoS program is realized.

The paper is organized as follows. Background and previous work is reviewed. The method, developed to search for configurations which address the five subgoals, is outlined with its supporting objective hierarchy. An experiment which demonstrates the utility of the method is outlined and recent network service value results are provided. The paper concludes with a listing of future opportunities.

## II. PREVIOUS WORK

### A. Graceful Recovery

Under graceful degradation, protocols manage traffic to maintain application performance under pressures of deteriorating network resources. Graceful recovery, in this context, applies to the network's ability to service applications in priority order as network resources are restored. Both concepts require a priori determination of organizational mission

and its reliance on certain applications as a prerequisite to realize high availability for IP networks [7] [8].

### B. Mission-Oriented Quality of Service

Previous research emphasized the benefit of both context and mission-awareness for application task and behavior [9] as well as for network security and protection requirements [10]. In [11] QoS parameters were captured in association diagrams starting with missions and [12] recommended further research to raise the QoS level of abstraction from network-centric (IP address, traffic class, etc.) to more abstract classes like high reliability or high priority. These are categories which can be organized to serve defined missions.

### C. Mission Prioritization and Characterization

The connections between mission, systems engineering and quality of service address many aspects outlined by Cebrowski in the Network Centric Operations (NCO) concept [13]. QoS implementations which are built bottom-up and optimized for operations can use as a basis the missions handed from top-down. This method of arranging QoS programming will deliver optimal performance for the infrastructure of the network system of systems.

### D. Quality of Service

Effective Quality of Service (QoS) acknowledges a requested level of performance, schedules or reserves such resources and then operates according to the agreement. Applications may then operate within the bounds of stated requirements with assurances that the network will support its operations. SLAs with SLSs or multiple individual contracts are used to manage this process. QoS in the network context is negotiated using one of various metrics. Bandwidth refers to the size of a data channel in terms of how much information can transit the channel per unit time and is often the critical managed resource. Other metrics such as end-to-end delay or latency, end-to-end jitter (variance of delay), bit error rate, packet loss rate, packet loss ratio, queuing delay and queue size are also available to infer performance.

### E. The Analytical Hierarchy Process (AHP)

AHP was designed to provide structure to multi-criteria decision problems designed to address an objective [14]. In AHP, independent subgoals are individually valued as contributors to the objective. The subgoals are valued according to a series of relevant criteria. The influence of a particular subgoal on the objective or the influence of a criteria measure on a subgoal are based on a relative weighting ultimately tied to a pairwise comparison process. Decisions made in one-to-one comparisons establish a comparison matrix and the principle eigenvector of this matrix leads to a priority vector. If the matrix consistency index is sufficiently low (demonstrating consistent prioritizations as direct comparisons are extrapolated to second and higher degrees), the priority vector pro-

vides a normalized, relative importance between and among subgoals or criteria all falling under the same heading [6] [14].

Competing telecommunications systems in traditional circuit-switched applications are assessed using customer responses to caller QoS standards then scored in an AHP-framework [15] [16]. In [17], an AHP methodology is followed to manage the resources of Grid Computing against an established SLA. In [18], QoS refers to the combination of service levels offered where AHP and Grey Relational Analysis permits selection among multiple 4th Generation (4G) networks. Quality of Service is applied to process control and web services which originate IP traffic [19] and web service QoS are evaluated using AHP [20] [21]. The Flexible Integrated System Capability (FISC) measure calculates a multi-dimensional measure to rate resource manager quality [22]. A utility model based on market pricing allows a mobile user to select the wireless local area network (WLAN) providing most value [23]. AHP has proven to be a valuable method for managing resources and effective decision-making in a number of network management challenges including cases which require a robust system-of-systems network and shared infrastructure.

### III. METHOD

The goal of this effort is to develop a repeatable method to realize network configuration improvements. The realization is driven by mission objectives which define network performance goals using Quality of Service (QoS) methods. Maximizing service value of network configuration subject to certain applicable constraints follows the format:

$$\text{Maximize } V(n_{hi}) = \sum s(Q_{hi}, c_j, v_k) \quad (1)$$

$$\text{subject to } S(r_a) \quad (2)$$

The service value  $V$  of a particular network configuration  $n_{hi}$  is calculated by summing the scores  $s$  for the set of subgoals detailed in the AHP-derived objective function. The subgoal scores are calculated during iteration  $h$  by observing network  $i$ 's satisfaction of each criteria  $c_j$ . Those criteria related to network performance are scored using Single Dimensional Value Function (SDVF)  $v_k$ . The  $n_{hi}$  network configuration utilizes QoS configuration  $Q_{hi}$  to manage network flows. The set of constraints  $S$  for the value function come directly from the IERs outlined in each SLS entry  $r_a$ .

A methodology is proposed to incorporate heuristic search pursuing improved network configurations as measured by  $V(n_{hi})$  from Eqn. 1. The method calculates the service value which a network configuration can deliver to the objective then finds alternative improved configurations and their scores by updating h-values in a form of heuristic search similar to Adaptive A\* [24] [25].

The network service value is calculated for a set of network configurations using models whose performance is simulated in OPNET Discrete Event Simulation and results then applied via the proposed methodology.

A feedback system permits simulation performance to influence subsequent network configurations.

$$N_{h+1} = C(\{V(n_{h1}), V(n_{h2}), \dots, V(n_{ht})\}) \quad (3)$$

The algorithm represented by  $C(V(n_{hi}))$  identifies the best-scoring of the configurations out of the set  $N_h = \{n_1, n_2, \dots, n_t\}$ . It then uses the best scoring configurations as a basis to search for the next set  $N_{h+1}$  to use in the next iteration. The process iterates by repeating these steps until improvement from new configurations reaches a pre-established limit.

#### A. Addressing the QoS Configuration Space

The network configuration alternatives are sampled from the set of all possible QoS configurations. The first set of alternatives are maximally separated and span the configuration space.

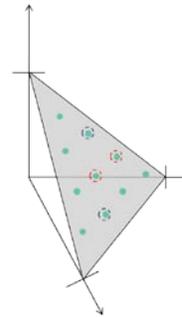


Fig. 2 3-Class Configuration Space

QoS protocols often use classes as ways to organize or bin network elements sharing common features. A class may contain a set of one to many information flows or applications which are all related by task, objective or some other defined relationship. The entire configuration space is defined by scales for each class with a QoS configuration occupying a point in the multi-scale space. The position of the point relates priority weight of each class amongst the other classes. Resources come from a finite pool so the

sum of all points' positions on the various scales must equal 1. If there are three classes, then the space of all possible configuration solutions is the face of a two-dimensional plane in the all-positive octant as we see in Fig. 2. For  $c$  classes, the possible configuration space is the  $c-1$  dimensional hyperplane in the all-positive space.

Developing this method requires building the components of the network service value objective function being maximized. The components may be realized by building an objectives hierarchy under the stated goal of "Obtain the best network service for the stated mission." The five subgoals which relate to this goal were provided in the introduction.

Relative weights for these subgoals result from pairwise comparisons under AHP. The objectives hierarchy supporting this analytical method is provided in Fig. 3.

A similar process is followed for the criteria which represent each sub-goal. Each criterion is assigned a relative weight based on its perceived importance in pairwise comparisons with the other criteria. If a criterion has sub-criteria, the AHP process is repeated with each set of sub-criteria for a criterion. The end result of this process is a set of tiered weights assigned to elements of the objective hierarchy at all levels. The weights have no units but each contributes to the overall value which is being maximized in the objective function. In the objective hierarchy outlined above, the weights are developed in the following fashion.

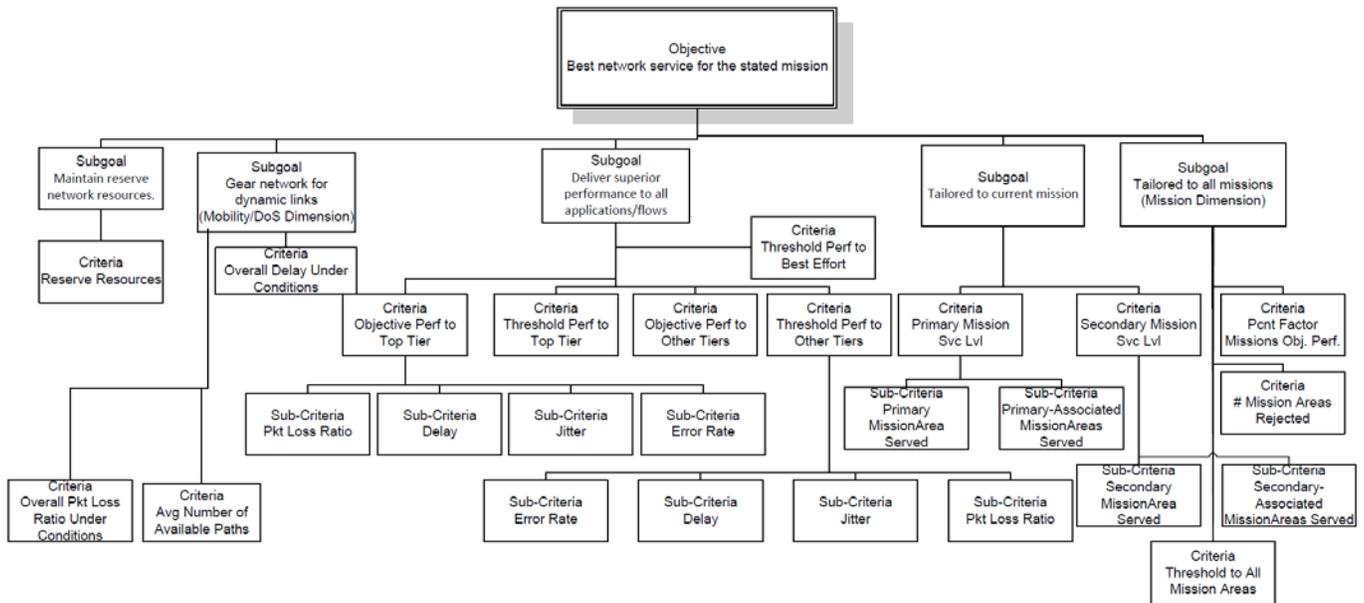


Fig. 3 Objective hierarchy used to find a network configuration to serve the stated mission.

### B. Objective: Best network service for stated mission.

The overall objective has five stated subgoals listed in the introduction. Referencing the labels in this list, a pairwise comparison of subgoals is made with the scale 1-9 (1 means subgoals are equally important and 9 meaning a subgoal is maximally more important). The relative scores would often be provided by a mission manager or mission management team (as opposed to a network engineering team) and allow completion of the upper half of the AHP matrix as in Table 1. Some guiding principles used by the mission manager in the comparison in the particular experiment described in this paper are:

- serve the current mission with superior (objective or better) performance
- favor the primary mission area but also provide superior (objective or better) service to the secondary mission area
- provide adequate (threshold) network capability to all defined missions
- design for dynamic links is a low priority
- reserve network capacity offers little value

Tailoring to circumstances at hand may result in derivative forms of the Fig. 3 hierarchy. This tailoring may, for exam-

ple, include an alternative organization of subgoals and criteria such as if missions are valued together as a related set and if they must compete with other sets on a shared infrastructure

Table 1 results by performing pairwise comparisons among subgoals while adhering to the mission manager guidance. The last column is the priority vector found by normalizing the primary eigenvector so the resulting weights sum to one. The consistency index (CI) for this arrangement is 0.072 and the consistency ratio (CR) is 6.4% giving us confidence in the consistency of the elements making up the decision matrix [14]. The priority vector in Table 1 provides the weights for the subgoals in the objective hierarchy in order to configure the network which meets the objective. Therefore, just over half the emphasis should go to subgoal 1. Subgoal 2 receives a quarter of the emphasis with an eighth dedicated to subgoal 4. A similar analysis establishes the weights for criteria and sub-criteria under each of the subgoals.

### C. Details for this method.

1) *Developing values for network performance:* The single dimensional value function (SDVF) permits one of a variety of measures to be translated into a common scale of value [4] [26]. A unique SDVF is required for each measure. Objective and threshold values as well as the shape of their interrelated SDVF depend heavily on the relationship between a flow and the mission(s) to which it contributes.

2) *Solving the complex linear program objective function:* The objective function for value detailed in Eqn. 1 is a linear program (LP) composed of subgoal and criteria LPs. Although some SDVFs may be non-linear, the objective function value  $V(n_{hi})$  will be continuous and solvable if the SDVFs are continuous. If SDVFs are represented as linear or piecewise linear functions, maximizing  $V(n_{hi})$  may be possible by solving an LP. In cases where clear divisions cannot be drawn between subgoals or criteria such as highly interdependent systems,

Table I

Matrix of relative importance for subgoals listed in introduction which result from pairwise comparisons

subgoals	1	2	3	4	5	priority vector
1	1	3	7	5	7	0.507
2	1/3	1	5	3	5	0.257
3	1/7	1/5	1	1/3	1/3	0.0438
4	1/5	1/3	3	1	3	0.123
5	1/7	1/5	3	1/3	1	0.0691

this AHP-based method may not be appropriate. These are design considerations for establishing and applying the sub-goals and criteria for the objective function.

3) *Finding improved network configurations*: The goal of finding an improved network configuration may be realized by observing the scores arising from the hierarchical objective function detailed in Eqn. 1. Equation 3 demonstrates how the scoring  $C(V(n_{hi}))$  leads to improvements which deliver the next network and its configuration  $N_{h+1}$ . The initial series of  $t$  possible QoS configurations, set  $Q_1 = \{Q_{11}, Q_{12}, \dots, Q_{1t}\}$ , evenly spans the  $c$  class dimensions. If it is known apriori which mission areas should receive preference, this vector of preferred starting areas is used to assign classes to dimensions and provide  $Q_1$  as a starting point for the heuristic search.

The top-scoring configurations form a basis set  $A$  for the next solution. The bottom-scoring configurations form a set  $B$  and are binned together with top-scoring configurations from  $A$  based on having similar features defined in  $C$  given by

$$(4)$$

Sets  $(AB)_1, (AB)_2, \dots$  each have top and bottom-scoring configurations which have similar features. This is a necessary step as each poor-performing configuration may be poor for different reasons. Only bottom-scoring configurations which have similar features to a top-scoring configuration should be used to adjust that top-scoring configuration. The set from iteration  $h$  includes pairings  $(A_h, B_h)$ . Combining the top and bottom scoring configurations yields improved configurations.

$$Q_{21} = Q_{1A_1} - kQ_{1B_1} + q \quad (5)$$

$$Q_{22} = Q_{1A_2} - kQ_{1B_2} + q \quad (6)$$

$$Q_{2y} = Q_{1A_y} - kQ_{1B_y} + q \quad (7)$$

The scaling factor  $k$  controls the influence of the bottom-scoring configuration and exponentially decays with each iteration. Limited noise  $q$  is added to permit searching of the space surrounding the new configuration point. Applying the resulting set of configurations in an iterative fashion,  $\{Q_1, Q_2, \dots, Q_z\}$ , yields subsequently improved configurations.

#### IV. EXPERIMENT AND ANALYSIS

In order to examine the performance of the AHP-based configuration improvement process, a simple network model was assembled using the OPNET's Discrete Event Simulation. The method outlined in Sec. 3 was implemented, incorporating OPNET to simulate and measure network performance. Utilizing a simulation engine permits the finding of improved configurations and performance without implicitly and analytically solving an LP as described in Sec. 3 Paragraph C-2. This is especially useful when an LP does not

Table II  
Key settings for network model and experiment.

Parameter	Value
simulation duration	20 minute snapshots
Number of mission area loads	8 x 50 kbps UDP
Number of classes	8
Number of general traffic loads	8 x 100 kbps UDP
Traffic scaling factor	[3x...8x]
Channel capacity	2 Mbps

properly capture the optimization problem. The basic network layout is provided in Fig. 4. Server local area networks (LANs) on the left provide information at the request of users in the Center LANs on the right. A total of 8 classes exist. The critical link has 2 Mbps capacity as shown in Fig. 4. The constrained link can be managed effectively using QoS mechanisms which will be exercised by the applications at the edge and the routers connected by the link. Weighted-Fair Queuing (WFQ) is the QoS differentiated services protocol used in this experiment. Traffic intensity is the aggravating factor in this experiment. As operations continue, all loads increase linearly with the traffic intensity factor. These increases aggravate the model since capacity remains at 2 Mbps. Table 2 lists traffic loads and other relevant factors. The figures presented here convey traffic intensity in terms of total mission area load divided by available capacity (Load/Capacity, L/C).

Through multiple iterations, the system arrives at a superior QoS configuration. The overall network value score, ranging [0...1], demonstrates the value of the various network configurations at six different traffic intensities. Figure 5 depicts the progression of value which various QoS configurations deliver as the configurations are exercised, examined and iteratively improved using the Sec. 3 AHP method. In Fig. 5, dotted plots represent early configurations, dashed plots are intermediate and the solid plots are the final set of QoS configurations. The final, superior configuration chosen is represented by the thicker, dark plot with highest network

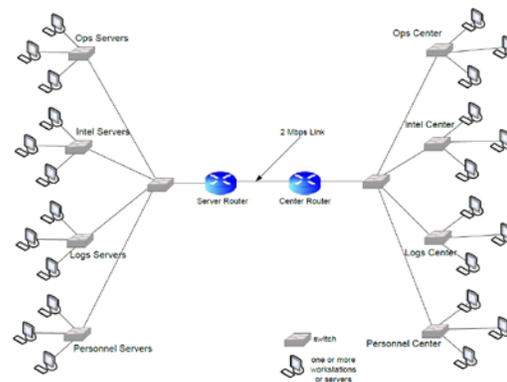


Fig. 4 Basic network model depicting various missions, users and corresponding servers.

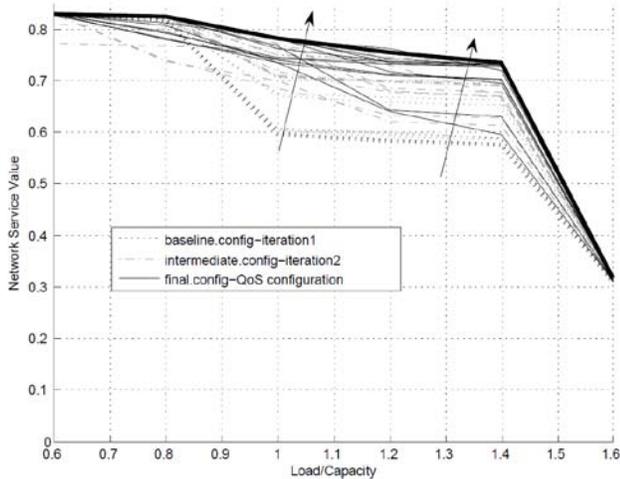


Fig. 5 Network Service Value for various configurations during increasing traffic intensity. Arrows demonstrate improving service value with subsequent iterations.

service value.

Within certain Load/Capacity ranges, the process provides a 30% improvement in network service value score over the most inappropriate configuration examined. Assessing the best-suited initial configuration as the starting point yields a 4% improvement in network service value. The top configurations maintain high network service value in the range 0.73-1.4 L/C..

Packet loss ratios are near 100% for best-effort traffic but for the primary mission area are zero up to 1.2 L/C and exceed 10% only above 1.5 L/C. Mission areas which are neither primary or secondary have zero losses until 1 L/C and those losses are limited to 25% at 1.6 L/C.

## V. CONCLUSIONS

Networks which are used by organizations to conduct operations function best when they are aligned to the goals and missions of the organizations. This alignment is especially critical when availability of resources is a major constraint. The Analytical Hierarchy Process (AHP) provides the means to design a formal and objective-based method to meet an objective function. Starting with the objective of managing a network to serve an organization's key interests, this paper described an AHP-based method driving an iterative search to find a superior Quality of Service (QoS) configuration for the network. The superior configuration enables efficient mission-oriented distribution of inadequate resources among competing applications which are associated with different mission areas.

The superior configuration meets the objective made up of a unique subgoal combination. A simple example experiment demonstrates the improved network service value delivered by following the prescribed method. Extending the AHP-based method to cases with different missions and different objectives would yield different weights for criteria and ultimately subgoals. These variations require multiple Fig. 1 and Fig. 3 representations yielding different configurations and future research will find methods to further tailor the method for varying missions. Interdependence and feedback often occur

among software applications and mission areas. While this method prescribes a first step to configure a mission-oriented network, more complex mission arrangements may not be well-served. The Analytic Network Process (ANP) generalization of AHP may offer future improvements when interdependencies increase the level of complexity. The effort described in this paper focused on connectionless, stateless communications as these carry critical data and consume considerable bandwidth. Future research will develop this method for connection-oriented traffic and will also examine additional aggravating factors such as link failure and network attack. Finally, this work relies on simulation results to approximate network performance but the method may also be applied to an operational network by measuring its network service value then feeding back updated QoS configurations.

## REFERENCES

- [1] Doshi, B., Kim, P., Liebowitz, B., Park, K. I., & Wang, S. "Service level agreements and QoS delivery in mission oriented networks," (White Paper No. 06-0084). Boston, MA: MITRE. 2006.
- [2] Naga, V., Colombi, J., Grimaila, M., Hopkinson, K., "Mission-related execution and planning through Quality of Service Methods," 15<sup>th</sup> Int. Command and Control Symposium, (ICCRTS) June 2010.
- [3] Jensen, E., "Utility Functions: A General Scalable Technology for Software Execution Timeliness as a Quality of Service," *Proc. Software Technology Conf.*, Utah State Univ., April 2000.
- [4] Dawley, L., Marentette, L., Long, A., "Developing a decision model for joint improvised explosive device defeat organization (JIEDDO) proposal selection," Thesis, Air Force Inst. of Technology 2008.
- [5] Keeney, R. *Value-Focused Thinking : A Path to Creative Decisionmaking*. Cambridge, Mass. : Harvard University Press, 1992
- [6] Saaty, T., *Decision Making with Dependencies and Feedback: The Analytic Network Process*, 1<sup>st</sup> Ed. Pittsburgh, PA : RWS Publications, 1996.
- [7] Sheth, P., "Build high availability into your IP Network: Part 1," EE Times Design Article, 3 Jan 2003.
- [8] Sheth, P., "Build high availability into your IP Network: Part 2," EE Times Design Article, 3 Jan 2003.
- [9] Loyall, J., Carvalho, M., Martignoni, A., Schmidt, D., Sinclair, A., Gillen, M., Edmondson, J., Bunch, L., Corman, D. "QoS enabled dissemination of managed information objects in a publish-subscribe-query information broker," in *Proc. of SPIE Conf. on Defense Transformation and Net-Centric Systems*. Orlando, FL. 13-17 Apr 2009.
- [10] Mitchell, G., Loyall, J., Webb, J., Gillen, M., Gronosky, A., Atighetchi, M., et al. "A software architecture for federating information spaces for coalition operations," in *Proceedings MILCOM 2008*, San Diego, CA. 2008.

- [11] Mujumdar, S., Mahadevan, N., Neema, S., and Abdelwahed, S. "A model-based design framework to achieve end-to-end QoS management," in *Proceedings of the 43rd Annual Southeast Regional Conference - Volume 1* (ACM-SE 43), Vol. 1. ACM, New York, NY, USA, 176-181. 2005.
- [12] Dasarathy, B., Gadgil, S., Vaidyanathan, R., Parmeswaran, K., Coan, B., Conarty, M., et al. "Network QoS assurance in a multi-layer adaptive resource management scheme for mission-critical applications using the CORBA middleware framework," Paper presented at the 11<sup>th</sup> IEEE Real Time and Embedded Technology and Applications Symposium, 2005. RTAS 2005. pp. 246-255.
- [13] Cebrowski, Arthur K. and John J. Garstka, "Network-Centric Warfare: Its Origins and Future," *U.S. Naval Institute Proceedings*, Annapolis, Maryland, January 1998.
- [14] Saaty, T., *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, 2<sup>nd</sup>Ed. Pittsburgh, PA : RWS Publications, 1990.
- [15] Douligeris, C. and Pereira, I., "An Analytical Hierarchy Process Approach to the Analysis of Quality in Telecommunication Systems," IEEE GLOBECOM 1992, Orlando, FL, Dec. 6-9 1992, pp. 1684-1688.
- [16] Douligeris, C. and Pereira, I., "A Telecommunications Quality Study Using the Analytic Hierarchy Process," *IEEE Journal on Selected Areas on Communications Special Issue on Quality of Telecommunications Services, Networks and Products*, Vol. 12, No 2, pp. 241-250, February 1994.
- [17] Bandini, M., Mury, A.R., and Schulze, R., "A Grid QoS Decision Support System Using Service Level Agreements," LNCC, Tech. Rep., 2009.
- [18] Charilas, D.E., Markaki, O.I., Nikitopoulos, D., and Theologou, M.E., "Packet-switched network selection with the highest QoS in 4G networks," presented at *Computer Networks*, 2008, pp.248-258.
- [19] Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K., "Quality of service for workflows and web service processes," *J. Web Sem.* 1(3): pp. 281-308. 2004.
- [20] Sun, Y., He, S.,Leu,J., "Syndicating web services: A QoS and user-driven approach," *Journal Decision Support Systems*, V43 No.1 Feb 2007.
- [21] Godse, M., Sonar, R., Mulik, S.: "The Analytical Hierarchy Process Approach for Prioritizing Features in the Selection of Web Service," *ECOWS 2008*: pp. 41-50.
- [22] Kim, Jong-K., Hensgen, D.A., Kidd, T.; Siegel, H.J.; St. John, D.; Irvine, C.; Levin, T.; Porter, N.W.; Prasanna, V.K.; Freund, R.F.; "A QoS performance measure framework for distributed heterogeneous networks," *Parallel and Distributed Processing, 2000. Proceedings. 8th Euromicro Workshop on* , vol., no., pp.18-27, 2000.
- [23] Chan, H., Fan,P., Cao, Z.; "A utility-based network selection scheme for multiple services in heterogeneous networks," *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on* , vol.2, no., pp. 1175- 1180 vol.2, 13-16 June 2005.
- [24] Koenig, S., Likhachev, M. and Furcy, D. "Lifelong Planning A\*" *Artificial Intelligence Journal*, 155, (1-2), 93-146, 2004.
- [25] Sun, X., Koenig, S., and Yeoh, W. "Generalized Adaptive A\*" In *Proceedings of the Int. Conf. on Autonomous Agents & Multiagent Systems (AAMAS)*, 2008.
- [26] Dalby, T., "Services flights as a result of shortages in manpower," Thesis, Air Force Inst. of Tech. 2007.

## BIOGRAPHIES

**Vinod D. Naga** is a Systems Engineering PhD candidate at the Air Force Institute of Technology. His research focuses on improving quality of service for DoD networks. He previously served as a technical liaison officer to a non-DoD agency, a field operations officer, a C2ISR technology transition manager and a program manager for E-8C JSTARS moving target indicator radar advancements at the Air Force Research Laboratory.

**Dr. John Colombi** is an Assistant Professor of Systems Engineering at the Air Force Institute of Technology. He leads sponsored research in architectural analysis, quality of service and Human Systems Integration. Before joining the faculty, Dr. Colombi led command and control systems integration efforts, systems engineering for the E-3 AWACS aircraft, served at the National Security Agency developing biometrics and information security and ran communications networking research at Air Force Research Laboratory.

**Dr. Michael R. Grimaila** is an Associate Professor of Systems and Engineering Management at the Air Force Institute of Technology. His research interests include mission assurance, cyber damage assessment, embedded system security, and risk mitigation. Dr. Grimaila serves as an Editorial Board member of the Information System Security Association (ISSA) Journal and consults for a number of DoD organizations.

**Dr. Kenneth Hopkinson** is an Associate Professor of Computer Science at the Air Force Institute of Technology. His research interests focus on simulation, networking, and distributed systems. More specifically, he looks at fault-tolerant and reliable approaches to mobile wireless networks as well as security and reliability in critical infrastructures.

# Measuring the Utility of a Cyber Incident Mission Impact Assessment (CIMIA) Notification Process

Christy Peterson, Michael R. Grimaila, Robert Mills

Center for Cyberspace Research  
Air Force Institute of Technology  
Wright-Patterson Air Force Base, OH 45433 USA  
{Christy.Peterson,Michael.Grimaila,Robert.Mills}@afit.edu

Michael W. Haas, Gina Thomas, Doug Kelly

711<sup>th</sup> Human Performance Wing  
Air Force Research Laboratory  
Wright-Patterson AFB, OH 45433 USA  
{Michael.Haas,Gina.Thomas}@wpafb.af.mil

**Abstract** — Information is a critical asset on which all modern organizations depend to meet their mission objectives. Military organizations, in particular, have embedded Information and Communications Technologies (ICT) into their core mission processes as a means to increase their operational efficiency, exploit automation, improve decision quality, and enable real-time situational awareness. However, an extreme dependence upon ICT results in an environment where a cyber incident can result in severe mission degradation, or possibly failure, with catastrophic consequences to life, limb, and property. In this paper, we present the initial results of an experiment designed to measure the utility of a Cyber Incident Mission Impact Assessment (CIMIA) notification process. CIMIA is focused upon minimizing the consequences following an information incident by maintaining real-time situational awareness of mission critical resources so appropriate contingency actions can be taken in a timely manner to assure mission success. The results of the experiment show that implementing a CIMIA notification process significantly reduced the response time required for subjects to recognize and take proper contingency measures. The research confirms that timely and relevant notification following a cyber incident is an essential element of mission assurance.

**Keywords-** CIMIA, Cyber incident notification, mission assurance, human subjects experiment, contingency planning

## I. INTRODUCTION

In a military context, information is continuously being collected, processed and analyzed, aggregated, stored, and distributed for multiple purposes, including support of situational awareness, operations planning, and command decision making (Fortson, 2007; Fortson et al., 2007; Grimaila et al., 2008a). Military organizations exhibit unique attributes such as high levels of sustained information interaction among multiple entities, distributed time sensitive decision making, and the criticality of consequences that may result from ill-informed decision-making (Grimaila et al., 2008a). In some cases, operations have critical time interdependencies which require significant planning and coordination to ensure the success of the mission objectives (Grimaila and Badiru, 2011). The timeliness of the information used in the decision making process dramatically impacts the quality of command decisions. Hence, the documentation of information

dependencies is essential for the organization to gain a full appreciation of its operational risks (Grimaila et al., 2009b; Grimaila et al., 2010). Information dependencies encompass not only the information itself, but also all of the ICT systems and devices used to store, process, transmit, or disseminate the information (NIST 800-30, 2002). One must understand how the information resources support the organizational objectives and how their value changes as a function of time in relation to other mission activities. This insight is needed to proactively design robust missions, develop and maintain situational awareness following an incident, take appropriate contingency measures to assure mission success, and to retain and exploit the “lessons learned” gained from experience (Grimaila et al., 2010; Hale et al., 2010). These facts led to the establishment of the Cyber Incident Mission Impact Assessment (CIMIA) project focused on the development of a Decision Support System (DSS) that provides automated, timely, accurate, secure, and relevant notification from the instant an information incident is declared, until the cyber incident is fully remediated (Grimaila et al., 2009a).

In this paper, we present the initial results of an experiment designed to measure the utility of timely and relevant notification following a cyber incident in a model operational setting. The underlying premise of the research is that timely and relevant notification will enable appropriate contingency actions to be taken sooner, improving operational outcomes and mission assurance. The research objectively evaluates the effectiveness of cyber incident notifications both in the status quo “as is” case and in the presence of the CIMIA incident notification process. The focus is upon understanding how information dependency knowledge can be used following a cyber incident to improve incident response and decision making in order to assure mission operations.

The remainder of this paper is organized as follows: In Section II, we present a brief review of the relevant literature surrounding existing incident notification. In Section III, we present the research hypothesis. In Section IV we present the research methodology and experimental design. In Section V, we present a statistical analysis of the results. Finally, in section VI, we discuss the conclusions, identify limitations, and identify future research.

## II. BACKGROUND

This section acquaints the reader with several key concepts and issues pertaining to the research.

### A. Situational Awareness

The concept of Situational Awareness (SA) has its roots in the fields of air traffic control, airplane cockpit control, military commands and control, and information warfare. SA can be traced back to World War I, where it was recognized as a crucial component for crews in military aircraft (Endsley 1996; Endsley & Jones, 1997, 2001; Endsley & Garland 2000). While SA can be achieved by people alone, the human factors approach often addresses the integration of technology, such as automation. While several limitations have been identified and discussed regarding problems with automation and automation failures, (Ephrath & Young, 1981; Kessel & Wickens, 1982; Wickens & Kessel, 1979; Young, 1969; Endsley 1996), Endsley (1996) points out that the use of automation can also be beneficial to achieving a higher level of SA with several new approaches to automation. One daunting challenge is to keep the “human in the loop” (Endsley, 1995). Endsley suggests one approach would be “to optimize the assignment of control between the human and the automated system by keeping both involved in the system operation” (1996). Furthermore, to reduce negative impact on the operator’s SA (lower levels), a level of automation should be determined while keeping the human actively involved in the decision making loop (Endsley, 1996).

According to Endsley, decision makers’ SA is a major factor driving the quality of the decision process (1997). SA influences the decision making process as it is “represented as the main precursor to decision making” (Endsley & Garland, 2000, p. 8). Endsley’s definition is widely recognized and defined as “the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future” (Endsley, 1988, p.97). The “elements” of SA “vary widely between domains, the nature of SA and the mechanism used for achieving SA can be described generically” (Endsley, 2000, p.5). SA is described as being dynamic, hard to maintain, and easy to lose. Although SA is a challenging to maintain, it is central to good decision making and performance (Endsley, 2000).

In cyberspace, decision makers face the challenge of maintaining a high level of situational awareness to function in a timely and effective manner following a cyber incident. SA in cyberspace is crucial to mission success to allow decision makers to understand what matters. They must be able to continuously depend on critical ICT and avoid working with tampered, corrupt, or missing information. Therefore, SA in cyberspace must be maintained in order to ensure information dominance in cyberspace. Thus, maintaining real-time SA of mission critical ICT resources before, during, and after cyber incidents is an essential element of mission assurance.

### B. The Existing USAF Incident Notification Process

USAF Instruction 33-138 defines the process used by Network Operations to generate, disseminate, acknowledge, implement, track, and report network compliance end status information (AFI 33-138, 2008). This document details the use of Time Compliance Network Orders for communicating

downward-directed operations, security and configuration management-related orders issued by the Air Force 624th Operations Center . Notification following a cyber incident occurs using a Command, Control, Communications, and Computers Notice to Airmen (C4 NOTAMs). C4 NOTAMs are informative in nature and are the primary means for notifying organizations that a network incident has occurred which may impact their mission operations. C4 NOTAMs are disseminated via email to organizations required to be notified in accordance with AFI 33-138. An example fictionalized C4 NOTAM is shown in Figure 1.

```
UNCLASSIFIED//FOR OFFICIAL USE ONLY
ACKNOWLEDGEMENT DATE: 13 JAN 2011
INITIAL RELEASE TIME: 13 0455Z JAN 11
TCNO TRACKING NUMBER: NOTAM C4-N AFNOC 2010-100-001
ORIGINATING AGENCY: 663 OC/CYCC
TYPE: INFORMATIVE
CATEGORY: NOTAM
PRIORITY: SERIOUS
SUBJECT: VULNERABILITY IN MICROSOFT RPC PROCESS COULD
ALLOW REMOTE CODE EXECUTION (321374)
MISSION IMPACT: LOSS OF SYSTEM AVAILABILITY/INTEGRITY
EXECUTIVE SUMMARY:
A VULNERABILITY EXISTS IN MICROSOFT RPC THAT COULD
ALLOW A REMOTE ATTACKER TO RUN CODE OF THE
ATTACKER'S CHOICE.
SYSTEM(S) AFFECTED:
WINDOWS XP SP 2 AND WINDOWS XP SP 3
WINDOWS XP PROFESSIONAL X64 EDITION SP 2
ACTION:
PATCHES ARE NOT AVAILABLE AT THIS TIME. ORGANIZATIONS
MAY APPLY THE MS ADVISORY WORK AROUNDS TO
TEMPORARILY ALLEVIATE THIS PROBLEM. A TCNO WILL BE
RELEASED ONCE MS ISSUES PATCHES FOR THIS VULNERABILITY.
NOTE: ALL WORK AROUNDS WILL IMPACT OPERATIONS IN SOME
WAY. PLEASE READ THE WORK AROUNDS CAREFULLY.
RISKS ASSOCIATED WITH UNPATCHED SYSTEMS:
UNAUTHORIZED ACCESS TO COMPROMISED SYSTEMS.
REPORTING REQUIREMENTS:
NONE
REMARKS:
PLEASE CONTACT 663 CS HELP DESK - IF YOU HAVE QUESTIONS
AND/OR CONCERNS AT 6503
```

Figure 1. Example Fictional C4 NOTAM

The C4 NOTAM is broadcast via email to communications personnel within organizations identified as potentially affected by the incident. It is up to communications personnel to decide if the information contained within the C4 NOTAM is further disseminated to all personnel within their organization. As a consequence, some organizations may be notified who are not dependent upon the affected ICT systems, the information may not be passed to personnel who are critically dependent upon the affected resource, and/or notification may be ignored to the sheer volume of notifications that do not affect the organizations (Grimaila et al., 2009b). Worse, some organizations may not be identified as dependent on the affected resource even though they are directly or indirectly critically dependent upon the affected ICT systems. This situation prevents a decision maker from consistently acquiring a meaningful level of SA on the status of critical ICT.

### C. The Cyber Incident Mission Impact Assessment (CIMIA) Incident Notification Process

The goal of the CIMIA incident notification process is to overcome limitations in the existing incident notification process by providing timely, accurate, secure, and relevant notification from the instant an information incident is declared, until the cyber incident is fully remediated (Grimaila et al., 2007b; Grimaila et al., 2009a). The basis for CIMIA is rooted in the need to identify and document critical resources prior to mission execution (Fortson, 2007). This introspection is not only required for mission risk management, but also enables the development and execution of informed contingency plans (Hale, 2010).

Hellesen (2009) examined the importance of proactive information valuation and Sorrells (2009) proposed a secure information architecture for the CIMIA incident notification process. Hale (2010) highlighted the importance of implementing a CIMIA incident notification process for mission assurance. Woskov (2011; Woskov et al. 2011) identified case-based reasoning as the ideal DSS technology for incident notification. Miller proposed a scalable incident notification architecture that links together decentralized mission dependent entities (Miller, 2011; Miller et al., 2011). A key benefit of the CIMIA approach is to replace the existing manual effort required to coordinate with the affected system owners and custodians with an semi-automated system that links entities together that are involved with mission fulfillment (Grimaila et al., 2008b). CIMIA will enable organizations to gain real-time notification following an information incident by subscribing to the status of mission critical resources.

CIMIA also differs in the way that incident notification occurs. Instead of utilizing email, the CIMIA incident notification process is disseminated via a pop-up to downstream consumers who subscribe and pull the status of the critical ICT they depend on. Pop-ups are known to visually capture that attention of an operator while using a computer. Some research suggests that interruptions (e.g. warnings, alerts, reminders, notifications) slows an operators' performance on interrupted tasks (Bailey et al., 2000); however, some evidence exist that an interruption may actually speed up the completion of a task (Zijlstra et al., 1999). Hence operators are affected by interruptions in different ways. It is important to identify key features that may impact the effectiveness of the interruption. Cohen (1988) suggests design of messages should focus on the "critical gaps between what consumers know and what they need to know" (p. 664). Because interruptions are typically viewed as communications whose purpose is to inform and influence behavior, the mockup of the pop-up was based on Laughery and Wogalter's eight criteria for design and assessment of warnings (1997, p. 1195):

1. Attention – should be designed to attract attention;
2. Hazard Information – should contain information about the nature of the hazard

3. Consequence information – should contain information about the potential outcomes
4. Instructions – should instruct about appropriate and inappropriate behavior
5. Comprehension – should be understood by the target audience
6. Motivation – should motivate people to comply
7. Brevity – should be brief as possible
8. Durability – should be available as long as needed

In addition to this criterion, the challenge was to make the CIMIA pop-up salient, attract the operator's attention, and to make the information seem relevant. Therefore, special consideration was given to the size, color, signal words, and content of the pop-up. An example CIMIA popup that would result from a root level breach of an internal web server containing mission critical information is shown in Figure 2.

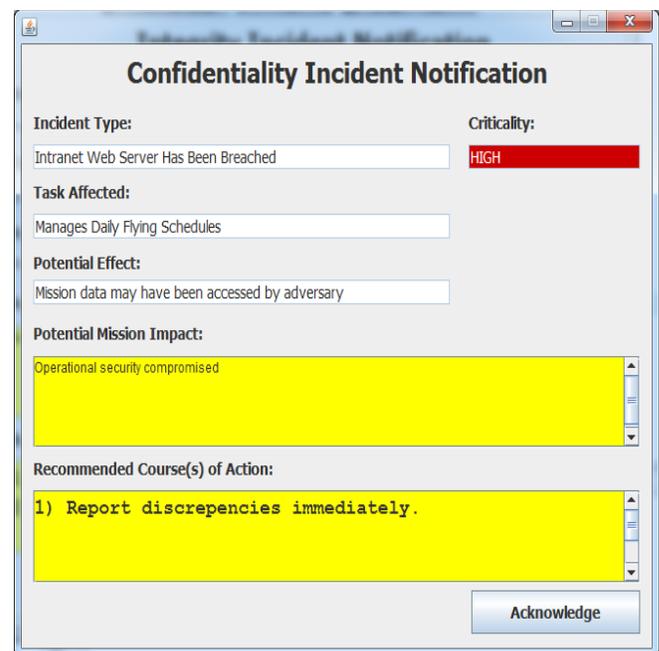


Figure 2. Example CIMIA Incident Notification

While it is envisioned that a CIMIA DSS will improve contingency decision making and provide knowledge continuity for mission owners, there has been no quantitative research to measure the effect of timely and relevant cyber incident notification with respect to contingency operations. The purpose of this experiment is to remedy this deficiency by designing an experiment in a realistic mission environment that will provide the empirical evidence necessary to test the utility of timely and relevant incident notification in terms of the time required to take appropriate contingency measures following a cyber incident.

### III. RESEARCH HYPOTHESIS

The main hypothesis for this research was developed based on the notion that it is important to promptly notify decision makers within an organization about cyber incidents

in a timely manner so they can take appropriate contingency measures to assure their mission. The key metric in this experiment is the difference between the time when an information incident occurs and when the participant takes the proper contingency action. The null and alternate hypotheses are stated as follows:

**Ho:** *There is no statistical difference between the existing and CIMIA incident notification processes in the length of time required for mission personnel to recognize and take proper contingency actions in response to cyber incidents.*

**Ha:** *There is a statistical difference between the existing and CIMIA incident notification processes in the length of time required for mission personnel to recognize and take proper contingency actions in response to cyber incidents.*

The research model used in this research is shown in Figure 3.

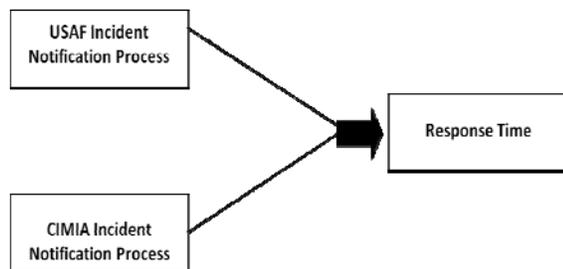


Figure 3. Research Model

#### IV. EXPERIMENT DESIGN

Military and civilian research personnel met regularly over a six month period and identified an aircraft Maintenance Operations Center (MOC) as the experimental environment in which to conduct the research. The MOC operates under the aircraft maintenance group and is considered the eyes and ears of the maintenance group commander. The MOC operates around-the-clock, and is continuously dependent on ICT throughout day-to-day operations. MOC personnel work day, swing, and mid shifts where they plan, schedule, and manage actions for assigned aircraft. This information-rich environment must maintain awareness of competing resources based on daily flying schedules and maintenance priorities (AFI 21-101, 2010; AFI 21-102, 2010; AFI 21-103, 2010). MOC personnel are responsible for maintaining aircraft readiness per AFI 21-101; they “monitor and coordinate sortie production, maintenance production, and execution of the flying and maintenance schedules” (p.114). Aircraft maintenance data collection and documentation are tracked in the Maintenance Management Information System known as “GO81” (AFI 21-101, 2010). This system is highly integrated with a global system called Global Decision Support System (GDSS). Both systems share information and operate independently. GO81 is used by MOC personnel to update the status of their aircraft. Information is transferred between systems by a middleware transfer agent on a periodic basis. For instance, GO81 may push aircraft discrepancies and

aircraft status to GDSS while GDSS pushes missions, launch, and landing times to GO81. GDSS is the authoritative source for military commanders to check aircraft availability, discrepancies, and monitor the status of the United States Air Force's fleet of aircraft.

The MOC ensures that the information is accurately entered into the GO81 in a timely manner so higher levels of command can determine aircraft availability for mission tasking (AFI 21-101, 2010). If information is not accurately updated in a timely manner, it could impair the military mission. Real-time data updates help reduce ground times and improve management of base support functions. It is apparent that the MOC's mission depends on information that is accurate to conduct operations. To obtain a better understanding, research personnel had the opportunity to visit a MOC. With the support of the MOC's superintendent, military personnel were interviewed and provided substantial input on the most critical aspects of their operations. This input was used to develop a case study providing a framework for evaluating the CIMIA incident notification process. Based on these findings, two mission objectives were used to develop an experiment: 1) ensure all GO81 information is entered accurately and in a timely manner, and 2) ensure aircraft status is reported accurately in both GO81 and GDSS.

##### A. Equipment and Facilities

The experiment was conducted in a room at the Air Force Institute of Technology on Wright-Patterson AFB, Ohio. The room was configured to resemble the operational environment of a MOC, as shown below in Figure 4.

A local area network (LAN) was constructed containing two workstations. The subject's workstation contained an email client and a graphical interface (GUI) interface to the GO81 database. In addition, two 42" monitors were used to display the current state of MOC aircraft in the GO81 and GDSS systems. A second workstation was used by the facilitator and contained an Oracle database, a Domain Name System (DNS) server, an email server, and host system for two databases, similar to GDSS and GO81.

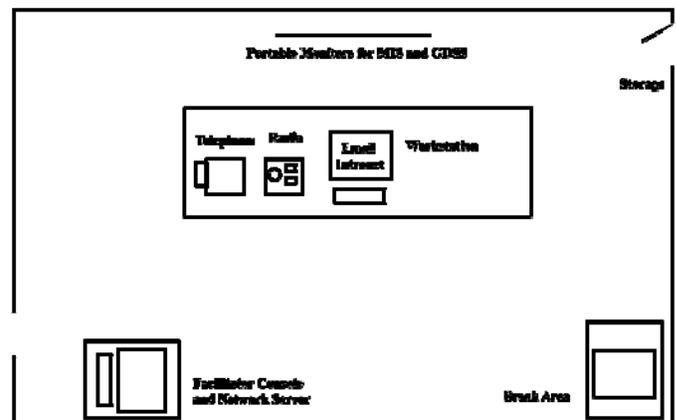


Figure 4. Experiment Environment

### B. Software

Oracle Database 10.2g was used to create two databases that mimic GO81 and GDSS functionality supporting the research experiment. A Graphical User Interface (GUI) was developed using NetBeans Integrated Development Environment (IDE) 6.9.1 to interact with the GO81. The MOC utilizes GO81 to document aircraft information and exercise related missions to ensure 100 percent reporting accuracy on aircraft mission capability (AFI 21-101, 2010). As shown in Figure 5, sixteen fields were selected for the case study scenario to provide relevant information about the aircraft's current mission capability. During the experiment, the subjects

interacted with the GO81 interface to change the status of a series of aircraft "owned" by the MOC.

The second database, GDSS, was designed with the same look and feel as the GO81 interface. In the experiment, data is transferred into GDSS two minutes after the information is entered into GO81. Each database has a total of 16 fields, of which only 12 are the same between the two systems. The aircraft summary displays provide visibility to monitor aircraft resources. The displays for GO81 and GDSS reflect aircraft owned by the hypothetical airlift wing regardless of the aircraft's global location.

Serial #	Tail #	MDS	Owner	Mission #	Call Sign	Mx Stat	Priority	GEOLOC	Eff Stat Time	JCH	Shop	Member Name	Spot	WJC	Remark
990001125	1125	C17A	663MKS	ZJGH85421364	SONIC 20	MC	1B1	PQWY	1600/1100		NA		NA		N
990001126	1126	C17A	663MKS	7GH487451258	HUSKY 10	FMC	1B1	PQWY	1500/1000		NA		NA		N
990001127	1127	C17A	663MKS			PMCS	1B1	TYFR	1900/1400	36269856	HYD	DANIELS	NA	3466AS001	Y
990001128	1128	C17A	DEPOT			NMCHU	1A4	PQWY	2015/1515	24474586	JETS	HOCKS	NA	5447AA001	Y
990001129	1129	C17A	663MKS			NMCH	1B1	PQWY	1918/1418	44587542	APG	SCOTT	NA	3623AA001	Y
990001130	1130	C17A	663MKS			MC	1A2	PQWY	0745/0245		NA		NA		Y
990001131	1131	C17A	663MKS	PUN474585693	RAWLEY 51	MC	1B2	XLWU	1235/0735		NA		NA		Y
990001132	1132	C17A	663MKS			NMCB	NA	PQWY	0910/0410	32545688	HYD	MARTIN	NA	6586B2002	Y
990001133	1133	C17A	663MKS	SAM698411256	CROME 12	PMCS	1A1	PRQE	1702/1202	96584525	ECM	TORRES	NA		Y
990001134	1134	C17A	663MKS	PUN444587988	HERC 85	FMC	1A3	PQWY	1523/1023		NA		NA		N
990001135	1135	C17A	663MKS			NMCB	1C3	PQWY	1009/0509	54855236	APG	MATTEWS	NA	6566AS002	Y
990001136	1136	C17A	663MKS			FMC	NA	PQWY	0236/0936		NA		NA		N
990001137	1137	C17A	663MKS	MKT59871129	LARRY 45	MC	1A2	ASHE	1922/1422		NA		NA		Y
990001138	1138	C17A	663MKS			NMCU	1B3	PQWY	0815/0315	25969535	MCS	GRANT	NA	8985AS001	Y
990001139	1139	C5	663MKS			NMCB	1A4	PQWY	1749/1249	24547436	APG	MOORE	NA	3454A2001	Y
990001140	1140	C5	663MKS			NMCH	1A1	ALDA	1203/0703	24547869	APG	TERRY	NA	4156AL002	Y
990001141	1141	C5	663MKS			NMCH	1A1	XDAT	0655/0155	74845966	APG	JONES	NA	9984BG001	Y
990001142	1142	C5	DEPOT			NMCB	1B1	PQWY	0916/0416	35471410	APG	HAYES	NA	6474AA001	Y

Figure 5. Example GO81 Information Display

### C. Experiment Design Approach

This experiment was tailored from factorial experimentation which "permits the manipulation of more than one independent treatment in the same experiment" (Keppel, 2009, p.20). Specifically, a 2 x 2 mixed factorial design with a combination of within-subjects and between-subjects variables was used. The term "mixed" refers to the elements of both within-subject and between-subject designs (Keppel, 2009). This design uses the same subjects with every condition of the research, including the control. In the 2x2 Mixed Factorial Design shown in Table I, the design consists of one within-subject variable (type of incident notification), with two levels (NOTAM and Pop-up), and one between-subjects variable (incident notification order), with two levels (NOTAM/Pop-up and Pop-up/NOTAM).

In this case, the USAF incident notification process (NOTAM) was one of two independent variables, compared to the proposed CIMIA incident notification process (Pop-up) being the second independent variable. Subjects who participated in the experiment received random assignment to the between-subject variable. This procedure guaranteed that the treatment condition had an equal opportunity of being assigned to a given subject and whatever other uncontrolled factors might be present during any testing (Keppel, 2009). "The critical features of random assignment, then, are that each subject-session combination is equally likely to be assigned to any one of the treatment and that the assignment of each subject is independent of that of the others" (Keppel, 2009, p.16).

TABLE I. MIXED FACTORIAL EXPERIMENT DESIGN

		Factor C Type of Incident Notification	
Factor A Initial Notification	Factor B Subjects	Session 1	Session 2
NOTAM	S <sub>1</sub> S <sub>2</sub> S <sub>3</sub> S <sub>8</sub> S <sub>9</sub> S <sub>10</sub> S <sub>11</sub> S <sub>16</sub> S <sub>17</sub> S <sub>18</sub> S <sub>19</sub> S <sub>24</sub> S <sub>25</sub>	NOTAM	Pop-up
Pop-up	S <sub>4</sub> S <sub>5</sub> S <sub>6</sub> S <sub>7</sub> S <sub>12</sub> S <sub>13</sub> S <sub>14</sub> S <sub>15</sub> S <sub>20</sub> S <sub>21</sub> S <sub>22</sub> S <sub>23</sub>	Pop-up	NOTAM

#### D. Experimental Scenario

The participants in the experiment were asked to complete a set of tasks that are representative of those found in the MOC. The participants were placed in this environment and trained on a series of tasks, such as updating a backlog of information into GO81, monitoring email, listening for radio updates, and answering any phone calls that come into the simulated MOC environment. Participants were instructed to contact the GDSS help desk immediately if they detected a discrepancy in the MOC aircraft information between the GDSS and GO81 systems.

During the experiment, participants experienced a series of cyber attacks that compromised Confidentiality, Integrity, and Availability (CIA) of their ICT resources. For example, information entered into GO81 may be corrupted while being transferred into GDSS. The time required for the participant to both notice that the information was incorrect in GDSS and notify the GDSS help desk of the error was the primary metric collected in this research.

#### E. Procedure

The participants in the experiment attended a 30 minute introduction and practice session, followed by two 30 minute experimental sessions. The two experimental sessions were separated by a 15 break period. During the session, participants were presented with cyber attacks that result in the loss of CIA at random times occurring at the same datasheets for each session. Additionally, each participant experienced distractions in the form of emails, radio updates, and calls that must be acknowledged, and in some cases they were required to take a course of action.

## V. RESULTS AND ANALYSIS

#### A. Subject Demographics

Participants in the experiment included 13 graduate students at the Air Force Institute of Technology and 12 undergraduate students from Wright State University. Subject demographic data are shown in Table II. The majority of the subjects were in the age groups 18-30 (64%). There were more male subjects (76%) than female (24%). Thirty-

six percent of the subjects did not have a degree, while 40% of the subjects had a Bachelors degree.

TABLE II. SUBJECT DEMOGRAPHICS

Demographic Factor (N=25)	Frequency	% of total	
Age	18-30	16	64
	30-45	8	32
	45-60	1	4
Gender	Male	19	76
	Female	6	24
Academic Level	No Degree	9	36
	Associate	4	16
	Bachelor	10	40
	Master	2	8

#### B. Experiment Results

The research objective was to compare the USAF incident notification process to the CIMIA incident notification process with respect to the response time to recognize and take proper contingency actions following a cyber incident. A paired sample t-test was used to compare the means of the two incident notification processes. Response time was measured in terms of the length of time (in seconds) it took a subject to report that a cyber incident had occurred. The main hypothesis for this research was developed based on the notion that it is important to promptly notify decision makers within an organization about cyber incidents in a timely manner so they can take appropriate contingency measures and assure their mission. The null and alternate hypotheses were:

**H<sub>0</sub>:** There is no statistical difference between the existing and CIMIA incident notification processes in the length of time required for mission personnel to recognize and take proper contingency actions in response to cyber incidents.

**H<sub>a</sub>:** There is a statistical difference between the existing and CIMIA incident notification processes in the length of time required for mission personnel to recognize and take proper contingency actions in response to cyber incidents.

Table III shows the results of the t-test of the paired differences between the sessions with CIMIA and without CIMIA. The null hypothesis was rejected (mean 311.4 and standard deviation 523.8), and it was concluded that there was a significant effect for the CIMIA incident notification process in the length of time required to recognize and take proper contingency actions in response to a cyber incident ( $t(40) = 3.807$ ,  $p < .000$ ,  $r = .50$ ). In the eight blocks where the treatment was absent, participants responded in approximately 8.85 minutes after either a loss of availability or loss of integrity occurred. On the other hand, when the CIMIA incident notification process, treatment, was present, the response decreased to 3.66 minutes. Hence, subjects in the eight blocks where the treatment was present responded

approximately 5.19 minutes sooner. Table IV summarizes these results. The effect size was calculated using the Rosenthal and Rosnow (1991) formula. The effect size of  $r = 0.52$  is a large effect which accounts for 25% of the variance (Cohen, 1988).

TABLE III. RESULTS OF T-TEST: PAIRED DIFFERENCE FOR SESSIONS WITH CIMIA AND WITHOUT CIMIA (RESPONSE TIME IN SECONDS)

Statistic	Paired Differences
Mean	311.4
Standard Deviation	523.8
Standard Error Mean	81.8
95% Confidence Level Lower	146.1
95% Confidence Level Upper	476.7

TABLE IV. DESCRIPTIVE STATISTICS (RESPONSE TIME IN SECONDS)

N=41	Minimum	Maximum	Mean	Standard Deviation
Without CIMIA	14.0	1740.0	531.5	409.5
With CIMIA	11.0	1222.0	220.1	261.7

### C. Discussion

As predicted in Ha, the CIMIA incident notification process had a large significant effect in the response time for subjects to recognize and take proper contingency actions in response to cyber incidents. These findings are consistent with Endsley's performance-based measures of situational awareness (SA). Performance-based measurements evaluate the real-life actions of a subject and only make inferences to SA. However, using direct testable response gives a more concise measurement of SA, which "requires a discernible, identifiable action from the operator" (Endsley, 2000, p.203). The fact that subjects had to observe what was going on in their environment (information available), make an assessment about the current state (information processing), and understand that an action was required (an alert) is an indication of levels 1 and 2 of SA. According to Endsley, different measures of SA can be defined by the points in the decision making process. Once subjects understood that something was wrong in their environment, they made a decision about the projected future state of the system and perceived a need to take a course of action. The actions taken by the subjects in response to the CIMIA incident notification process are testable responses that reinforce inferred higher levels of SA.

Clearly, subjects had higher levels of SA and performed more successfully in the second session of the experiment which consequentially increased the number of discrepancies reported from the induced manipulations of cyber attacks. In session 1 absent the treatment, subjects responded only 50 percent of the time to discrepancies, while 92 percent responded after being exposed to the treatment. Conversely, subjects that were exposed to the treatment in session 1 and absent the treatment in session 2

responded 92 percent and 100 percent of the time respectively. Having the CIMIA incident notification process in session 1 alerted the subject to look more closely for discrepancies in session 2 when the treatment was absent.

## VI. CONCLUSION

This paper presented the initial results of an experiment to measure the utility of a Cyber Incident Mission Impact Assessment (CIMIA) incident notification process in terms of the time required to take a contingency action. In the absence of the CIMIA notification process, participants required 8.85 minutes on average after either a loss of availability or loss of integrity occurred. In contrast, in the presence of the CIMIA incident notification process, the average response decreased to 3.66 minutes. The subjects that received the NOTAM as the initial notification performed worse than the subjects that received the pop-up first. The subject's performance in response to the NOTAM in the second session of the experiment was better after being exposed to the pop-up. In each instance, the pop-up had a positive effect on performance and resulted in reduction of response times. Further analysis of the results is underway and is will be present in a journal article submission.

## VII. DISCLAIMER

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

## VIII. REFERENCES

- Adelman, L. (1991). "Experiments, Quasi-Experiments, and Case Studies: A Review of Empirical Methods for Evaluating Decision Support Systems." IEEE Transaction on Systems, Man, and Cybernetics, Vol. 21 No. 2, March/April 1991.
- Bailey, B. P., Konstan, J. A., & Carlis, J. V. (2000). "Measuring the Effects of interruptions on Task Performance in the User Interface." Paper presented at the IEEE Conference on Systems, Man and Cybernetics, Nashville, TN.
- Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers, p. 19.
- DeCoster, J., (2001). "Transforming and Restructuring Data." Department of Psychology. University of Alabama. Retrieved from <http://www.stat-help.com/struct.pdf>.
- Department of the Air Force. (2010). Aircraft and Equipment Maintenance Management. AFI 21-101. Washington: HQ USAF. 26 July 2010. Retrieved from <http://www.e-publishing.af.mil/shared/media/epubs/AFI21-101.pdf>
- Department of the Air Force. (2010). Depot Maintenance Management. AFI 21-102. Washington: HQ USAF. 19 July 1994. Retrieved from <http://www.e-publishing.af.mil/shared/media/epubs/AFI21-102.pdf>
- Department of the Air Force. (2010). Equipment Inventory, Status and Utilization Reporting. AFI 21-103. Washington: HQ USAF. 9 Apr 2010. Retrieved from <http://www.e-publishing.af.mil/shared/media/epubs/AFI21-103.pdf>
- Department of the Air Force. (2005). Enterprise Network Operations Notification and Tracking. AFI 33-138. Washington: HQ USAF. 28 November 2005. Retrieved from <http://www.e-publishing.af.mil/shared/media/epubs/AFI33-138.pdf>.
- Endsley, M.R., (1995). "Measurement of situation awareness in dynamic systems." Human Factors, 37 (1), p. 65-84.

- Endsley, M. R. (1996). "Automation and situation awareness." In R. Parasuraman & M. Mouloua (Eds.), *Automation and human performance: Theory and applications* (p. 163-181). Mahwah, NJ: Erlbaum.
- Endsley, M.R. and Jones, W.M. (1997). "Situation Awareness and Information Dominance and Information Warfare." United States Air Force Armstrong Laboratory. February 1997.  
<http://www.satechnologies.com/Papers/pdf/IW%26SAreport%20.pdf>
- Endsley, M.R. (1988). "Design and evaluation for situation awareness enhancement." In Proc. of Human Factors Society 32nd Annual Meeting (Vol. 1 p. 97-100). Santa Monica, CA Human Factors Society.
- Endsley, M.R. and Garland, D.J. (2000). "Situation Awareness and Analysis and Measurement." CRC Press, Boca Raton, FL.
- Endsley, M. R., & Jones, D. G. (2001). "Disruptions, Interruptions, and Information Attack: Impact on Situation Awareness and Decision Making". Paper presented at the Human Factors and Ergonomics Society 45th Annual Meeting. Santa Monica, CA.
- Ephrath, A. R., and Young, L.R. (1981). "Monitoring vs. man-in-the-loop detection of aircraft control failures." In J. Rasmussen and W.B Rouse (Eds.), *Human decision failures*. New York: Plenum Press.
- Fields, A., (2005). *Discovering Statistics Using SPSS*. Sage Publications Ltd, London.
- Fortson, L.W. and Grimaila, M.R. (2007) "Development of a Defensive Cyber Damage Assessment Framework," Proc. of the 2007 International Conference on Information Warfare and Security (ICIW 2007). Naval Postgraduate School, Monterey, CA.
- Fortson, L.W. (2007). "Towards the Development of a Defensive Cyber Damage and Mission Impact Methodology," Master's Thesis, AFIT/GIR/ENV/07-M9, Department of Systems and Engineering Management, Air Force Institute of Technology, Wright-Patterson AFB, March 2007.
- Grimaila, M.R. and Fortson, L.W. (2007) "Towards an Information Asset-Based Defensive Cyber Damage Assessment Process," Proc. of the 2007 IEEE Computational Intelligence for Security and Defense Applications (CISDA 2007). Honolulu, HI, pp. 206-212.
- Grimaila, M.R., Mills R.F., Fortson, L.W., and Mills, R.F. (2008a) "An Architecture for Cyber Incident Mission Impact Assessment (CIMIA)," Proc. of the 2008 International Conference on Information Warfare and Security (ICIW 2008). Omaha, NE 2008
- Grimaila, M.R., Mills, R.F., and Fortson, L.W., (2008b) "An Automated Information Asset Tracking Methodology to Enable Timely Cyber Incident Mission Impact Assessment," Proc. of the 2008 International Command and Control Research and Technology Symposium (ICCRTS 2008). Bellevue, WA
- Grimaila, M.R., Fortson, L.W., and Sutton, J.L. (2009a). "Design Considerations for a Cyber Incident Mission Impact Assessment (CIMIA) Process," Proc. of the 2009 International Conference on Security and Management (SAM09). Las Vegas, NV 2009
- Grimaila, M.R., Schechtman, G., and Mills, R.F. (2009b) "Improving Cyber Incident Notification in Military Operations," Proc. of the 2009 Institute of Industrial Engineers Annual Conference (IERC 2009). Miami, FL.
- Grimaila, M.R., Mills, R.F., Haas, M., and Kelly, D. (2010), "Mission Assurance: Issues and Challenges," Proc. of the 2010 International Conference on Security and Management (SAM10), Las Vegas, Nevada, July 12-15, 2010.
- Grimaila, M.R. and Badiru, A. (2011), "A hybrid dynamic decision making methodology for defensive information technology contingency measure selection in the presence of cyber threats," *Operational Research*, pp. 1-22.
- Hale, B. Grimaila, M.R., Mills, R.F., Haas, M., and Maynard, P., "Communicating Potential Mission Impact using Shared Mission Representations," Proc. of the 2010 International Conference on Information Warfare and Security (ICIW 2010), WPAFB, OH, April 8-9, 2010.
- Hale, B. (2010), "Mission Assurance: A Review of Continuity of Operations Guidance for Application to Cyber Incident Mission Impact Assessment," Master's Thesis, AFIT/GIR/ENV/10-J01, Department of Systems and Engineering Management, Air Force Institute of Technology, Wright-Patterson AFB, June 2010.
- Keppel, G., (1982). *Design and Analysis: A Researcher's Handbook*, Second Edition. Prentice-Hall Inc., Englewood Cliffs, N.J.
- Keppel, G. and Saufley, W.H. (1980). *Introduction to Design and Analysis: A Student's Handbook*. W.H. Freeman and Company: New York.
- Kessel, C.J. and Wickens C.D. (1982). "The transfer of failure-detection skills between monitoring and controlling dynamic systems." *Human Factors*, 24, (1), p. 46-60.
- National Institute of Standards and Technology. (2002). *Risk Management Guide for Information Technology Systems*. NIST Special Publication 800-30. Gaithersburg, MD: Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, US Department of Commerce, March 2008.
- Miller, J.L. (2011), "An Architecture for Improving Timeliness and Relevance of Cyber Incident Notification," Master's Thesis, AFIT/GCO/ENG/11-09, Department of Computer and Electrical Engineering, Air Force Institute of Technology, Wright-Patterson AFB, March 2011.
- Miller, J.L., Mills, R.F., Grimaila, M.R., and Haas, M.W. (2011), "A Scalable Architecture for Improving the Timeliness and Relevance of Cyber Incident Notifications," Proc. of 2011 IEEE Symposium on Computational Intelligence in Cyber Security, Paris, France, April 12-15, 2011.
- Pipkin, D.L. (2000). *Information Security Protecting the Global Enterprise*. Hewlett-Packard Company.
- Rosenthal, R., and Rosnow, R. L. (1991). *Essentials of Behavioral Research, Methods and Data Analysis*. San Francisco: McGraw-Hill, p. 276-300.
- Sorrels, D.M., Grimaila, M.R., Fortson, L.W., and Mills, R.F. (2008). "An Architecture for Cyber Incident Mission Impact Assessment (CIMIA)," Proc. of the 2008 International Conference on Information Warfare and Security (ICIW 2008), Peter Kiewit Institute, University of Nebraska Omaha.
- Wickens, C.D. and Kessel C. J. (1979). "The effect of participatory mode and task workload on the detection of dynamic system failures." *IEEE Transactions on Systems, Man and Cybernetics*. SMC-9(1), p. 24-34.
- Woskov, S. M. (2011), "Improving the Relevance of Cyber Incident Notification for Mission Assurance," Master's Thesis, AFIT/GIR/ENV/11-M06, Department of Systems and Engineering Management, Air Force Institute of Technology, Wright-Patterson AFB, March 2011.
- Woskov, S., Grimaila, M.R., Mills, R.F., and Haas, M.W. (2011), "Design Considerations for a Case-Based Reasoning Engine for Scenario-Based Cyber Incident Notification," Proc. of 2011 IEEE Symposium on Computational Intelligence in Cyber Security, Paris, France, April 12-15, 2011.
- Zijlstra, F. R. H., Row, R. A., Leonora, A. B., and Krediet, I. (1999). "Temporal factors in mental work: Effects of interrupted activities." *Journal of Occupational and Organizational Psychology*, Vol. 72, p. 163-185

# A Spatial Risk Analysis of Oil Refineries within the United States

Zachary L. Schiff<sup>1</sup> and William E. Sitzabee, Ph.D., P.E.<sup>2</sup>

<sup>1&2</sup>Systems and Engineering Management, Air Force Institute of Technology, WPAFB, OH, USA

**Abstract** - *A risk analysis methodology is necessary to manage potential effects of oil refinery outages to the increasingly connected, interdependent critical infrastructure of the United States. This paper outlines an approach to develop a risk analysis methodology that incorporates spatial and coupling elements in order to develop a better understanding of risk. The methodology proposed in this paper utilizes a three phase approach to look at both natural disaster and terrorist risk. Understanding the uncertainty involved with the events that could shut down the petroleum energy sector enables decision-makers to make better decisions in order to manage risk to the government, people, and economy.*

**Keywords:** Geographic Information Systems (GIS), Critical Infrastructure, Risk Analysis

## 1 Introduction

In the past decade, the United States has experienced first-hand the devastating impacts of disasters, both natural and terrorist, to critical infrastructure. The events of the September 11, 2001 attacks (9/11); Hurricanes Ike, Katrina, and Rita; and British Petroleum's Deep Horizon oil accident illustrate the effects of a major disaster to the United States. The monetary costs of 9/11, Hurricane Katrina, and Deep Horizon oil accident are estimated at \$110 billion, \$81 billion, and \$40 billion, respectively [1]-[3]. The nation's security, economy, and health are dependent on critical infrastructure to provide key services in order for the government, people, and businesses to function properly.

During Hurricanes Katrina and Rita, refinery capability was reduced 13 percent and 14 percent, correspondingly. Due to reduced capacity, the hurricanes influenced gas prices to rise from \$1.10 to \$2.55 after the disasters [4]. The cost is an increase that has not been recovered from and has contributed to the economic recession. In addition, increased petroleum demand in the past 20 years has increased at a faster rate than refining capability to provide gas, diesel, and other petroleum products. According to GAO-09-87, refineries are producing at a level very near their maximum capacity across the United States [5]. As a result, a disaster, either natural or terrorist, could potentially result in large shortages for a given time period.

The Department of Defense (DoD) fuel costs represented nearly 1.2 percent of total DoD spending during Fiscal Year 2000 and increased to nearly 3.0 percent by Fiscal Year 2008 [6]. Andrews [6] stated that over the same period, total defense spending doubled and fuel costs increased 500 percent from \$3.6 billion to \$17.9 billion. Nearly 97.7 billion barrels of jet fuel were consumed in FY2008 and represents nearly 71 percent of all fuel purchased by the DoD. According to the Air Force Infrastructure Energy Plan, the fuel bill for the Air Force exceeds \$10 million dollars per day and every \$10 per barrel fuel price increase drives costs up \$600 million dollars per year [7]. In 2007, the Air Force spent \$67.7 million on ground fuel energy and consumed 31.2 million gallons of petroleum. The ground fuel energy only accounts for four percent of all fuel costs [7]. The military is a large customer of oil refinery products and is dependent on petroleum to complete military operations.

In the past decade, the petroleum industry has experienced several examples of cascading failures, including Hurricanes Katrina and Rita. These experiences can provide useful data with regards to outages and consequences of the events. Integrating spatial analysis into the research provides two opportunities to advance risk management: 1) utilize spatial tools to analyze relationships that provide insight into how the system functions and 2) visually identify trends that are not obvious within data analysis. This paper outlines an approach to develop a modified risk equation incorporating interdependency and spatial relationships utilizing critical infrastructure analysis and geographical information systems and sciences.

## 2 Background

### 2.1 Critical Infrastructure

The USA Patriot Act of 2001 (P.L. 107-56 Section 1016e) contains the federal government's definition of critical infrastructure. It stated that critical infrastructure is the "set of systems and assets, whether physical or virtual, so vital to the United States that the incapacity or destruction of such systems and assets would have a debilitating impact on security, national economic security, national public health or safety, or the combination of those matters."

The National Strategy for Homeland Security categorized critical infrastructure into 13 different sectors and they are as follows: Agriculture, Food, Water, Public Health, Emergency Services, Government, Defense Industrial Base, Information and Telecommunications, Energy, Transportation, Banking and Finance, Chemical Industry and Hazardous Materials, and Postal and Shipping [8].

Approximately 85 percent of the national infrastructure is owned by private industry [9]. The relationship between government and private industry is complicated with the government acting as both regulator and consumer. This is especially true within the energy sector which is composed of electrical power, oil, and gas infrastructure [10]. The energy sector is connected physically and virtually to all other sectors and has been shown to cause cascading failures to other sectors.

The petroleum industry was split into five Petroleum Administration for Defense Districts (PADDDs) based on geographic location during WWII [11]. Parformak [12] discussed geographic concentration of critical infrastructure across numerous sectors and policy methods for encouraging dispersion. Specifically, Texas and Louisiana (PADD 2) refineries account for over 43 percent of the total United States refining capacity [12]. Rinaldi, Peerenboom, and Kelly [13] discussed interdependencies, coupling and response behavior, and types of failures with respect to critical infrastructure across the United States.

## 2.2 Risk Analysis Methods

The Department of Homeland Security (DHS) introduced the risk function as a combination of threat, vulnerability, and consequence, displayed below as Equation (1) [14]. Lowrance [15] introduced risk as a measure of the probability and severity of adverse effects. Chertoff [14] defined threat as a natural or manmade occurrence that has the potential to harm life, operations, or property; vulnerability as the physical feature that renders an entity open to exploitation; and consequence as the effect and loss resulting from event.

$$Risk = f(Threat, Vulnerability, Consequence) \quad (1)$$

Solano [16] investigated vulnerability assessment methods for determining risk of critical infrastructure and spatial distribution appeared to be an area where research can be expanded. Rinaldi, Peerenboom, and Kelly [13] discussed the challenges of modeling multiple interdependent infrastructures due to volume of data required and that isolation of infrastructure does not adequately analyze behavior of the system. Ahearne [17] discussed the appropriateness of the multiplicative use of the risk function and found that it is generally accepted for natural disasters. Chai, Liu, et al. [18] utilized a social network analysis to evaluate the relationship between infrastructure risk and

interdependencies. The study utilized a node and arc approach to determine the number of in and out degrees to show dependencies and coupling. Expanding this approach could potentially result in better quantification of coupling effects on critical infrastructure.

Mohtadi [19] presented extreme value analysis as a method to predict large-scale terrorism events. In the study, methods for measuring terrorism as a probabilistic risk were developed for terrorism risk which is extreme and occurs infrequently. Paté-Cornell and Guikema [20] presented a model that utilized risk analysis, decision analysis and elements of game theory to account for both the probabilities of scenario and objectives between the terrorists and United States. In their research, the importance of utilizing a multi-source method for collecting data on terrorism risk which includes expert opinion, output of other system analysis, and statistics from past events. Leung, Lambert and Mosenthal [21] utilized the risk filtering, ranking, and management (RFRM) and Hierarchical Holographic Modeling (HHM) to conduct a multi-level analysis of protecting bridges against terrorist attacks.

## 2.3 Geographic Information Systems Tools

Nearly 40 years ago, Tobler [22] stated that “nearly everything is related to everything else, but near things are more related than distant things.” This became Tobler’s First Law of Geography and is acknowledged as the foundation of geographic information systems and science. Longley, Goodchild, Maguire, and Rind [23] discussed spatial autocorrelation as a tool that allows us to describe the interrelatedness of events and relationships that exist across space. Griffith [24] discussed spatial autocorrelation as “a dependency exists between values of a variable...or a systematic pattern in values of a variable across the locations on a map due to underlying common factors.”

## 3 Methodology

The goal of this study is to establish a process and develop techniques that can be expanded to look at the risk to both the critical infrastructure system and critical components of the system. This is a three-phase study and is organized in the following manner: 1) assess and compile inventory of assets, risk components, and characteristics; 2) validate the natural disaster quantitative risk model with spatial and coupling effects, and 3) qualitatively assess terrorism risk utilizing coefficients from the quantitative model. Fig. 1 shows the research process and provides an outline of the phase progression.

The first phase analyzed the factors that contribute to risk, the data available to characterize infrastructure, and the methodologies that are currently used to quantify risk. This paper presents the first phase of the study, which resulted in

the identification of two additional variables: 1) spatial relationship and 2) coupling effect. Equation (2) shows the modified risk equation which is the focus of the next phases.

$$\text{Risk} = f(\text{Threat}, \text{Vulnerability}, \text{Consequence}, \text{SpatialRelationship}, \text{CouplingEffect}) \quad (2)$$

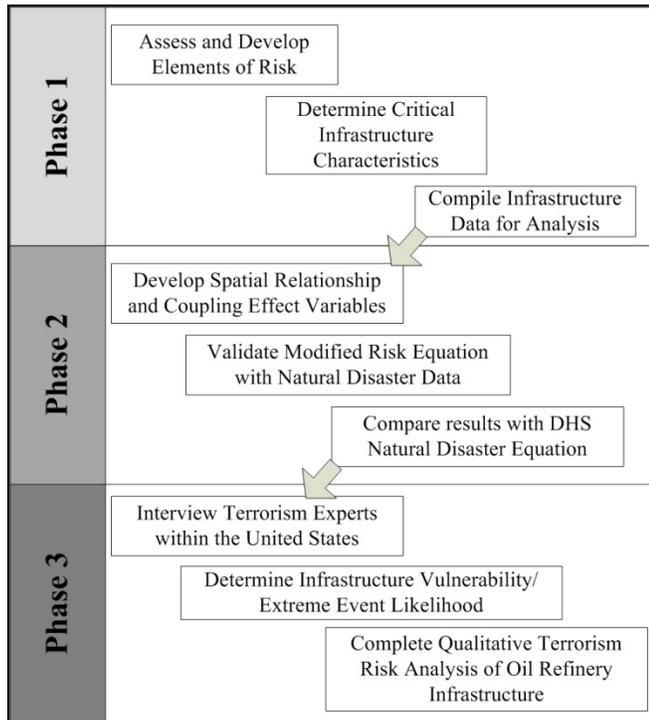


Fig. 1 Overview of Research Phases and Risk Analysis Methodologies

The goal of the second phase is to better quantify the cumulative risk of cascading failures by including the spatial relationship and coupling effects. To determine the spatial relationship, spatial auto-correlation will be utilized to develop a quantitative relationship of distance between critical infrastructures. The coupling effect will utilize a node-arc analysis to determine the number of connections to other infrastructures and expand on the research effort by Chai, Lui, et al. [18]. Natural disaster data will be utilized to develop a case study to compare the results of the second phase to already established and validated methods of quantifying risk.

The third and final phase of this research is to utilize the spatial and coupling effect information to qualitatively assess terrorism risk. This phase will include phenomenological methods which will be utilized to interview experts in the terrorism field in order to develop threat and vulnerability data for petroleum infrastructure. The combination of results from the second and third phases will provide the foundation to complete a qualitative terrorism risk assessment. The goal of the third phase is to determine the highest terrorism risk to oil refinery infrastructure that could potentially result in cascading failures and large impacts to the United States.

## 4 Preliminary Findings

In order to study the spatial relationships of refineries within the United States, ESRI ArcMap 9.3.1 was utilized to complete an initial analysis. Data were collected from public sources such as the U.S. Department of Energy (DoE), Energy Information Administration (DoE), National Oceanic and Atmospheric Administration (NOAA), and the United States Census Bureau. The data pieces include the following refining capacity and location by refinery, hurricane paths (both lines and points), and baseline state and world boundaries.

The initial analysis consisted of creating a 50-mile buffer zone around each refinery and overlaying hurricane tracks for both Category 4 and 5 with the intent of visually inspecting the relationship between storms and clusters of refineries. Fig. 2 below, shows the number of Category 4 and 5 storms historically that have made landfall within a 50-mile radius of a refinery.

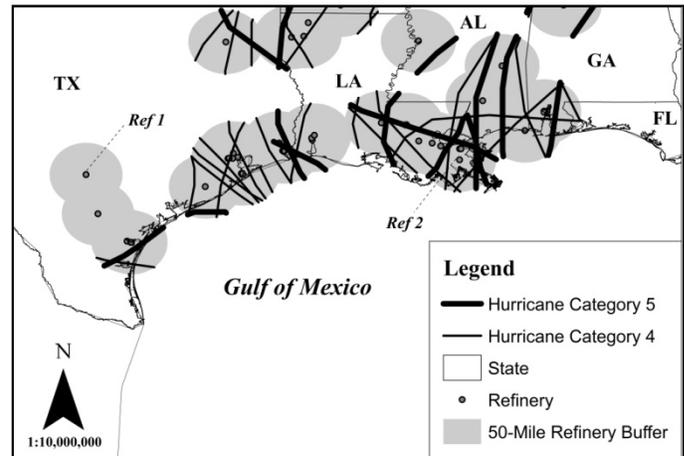


Fig. 2 50-mile Refinery Buffer and Category 4 and 5 Hurricane Tracks

The results of the initial analysis confirm that there is a spatial relationship in the infrastructure with respect to natural disasters. For example, refinery 1 shown above has not been impacted by a hurricane of a Category 4 or 5 strength. Refinery 2 on the other hand, has experienced three Category 5 hurricanes and two Category 4 hurricanes. This shows that the location of a refinery can increase the risk to the refinery. Also of note, refineries with the highest production capacity were often located in clusters of two to three within a mile of each other. Further analysis will be completed in Phase II to determine how the spatial relationship and coupling effects can be incorporated into the overall risk equation.

## 5 Conclusion

The relationships between critical infrastructures are complicated and interdependencies that exist between infrastructures are not well-defined. Incorporating spatial relationships and coupling effects into the risk equation

proposes a better way to predict the effect of interdependencies which have been shown to cause cascading failures during disaster events. Understanding and analyzing risk provides the decision and policy-making process better information in order to protect critical infrastructure across the United States.

This paper presents a new risk equation and a methodology to analyze and validate risk based on the modified risk equation. While spatial relationships and coupling have been identified as key factors to quantifying infrastructure risk, it appears that this is an area of study that requires further investigation. This research intends to further define the interdependencies of the infrastructure system in order to better quantify the overall risk to both the infrastructure system and individual parts of the system.

## 6 References

- [1] Robbie Berg, "Tropical Cyclone Report Hurricane Ike (TCR-AL092008), National Hurricane Center, National Oceanic and Atmospheric Administration (NOAA); 2009.
- [2] R. Knabb, J. Rhome, and D. Brown, "Tropical Cyclone Report Hurricane Katrina (AL122005), National Hurricane Center, National Oceanic and Atmospheric Administration (NOAA); 2005.
- [3] W.C. Thompson, "One Year Later: Fiscal Impact of 9/11 on New York City", City of New York: 2002.
- [4] J. Seesel, "Investigation of Gas Price Manipulation and Post-Katrina Gas Price Increases, Federal Trade Commission, Washington, D.C., 2006.
- [5] F. Rusco, "Refinery Outages Can Impact Petroleum Product Prices, but No Federal Regulations to Report Outages Exist," Federal Trade Commission, Washington, D.C.; 2008.
- [6] Anthony Andrews, "Department of Defense Fuel Spending, Supply, Acquisition, and Policy," Congressional Research Service Report for Congress, Washington, D.C.; 2009.
- [7] "Air Force Energy Infrastructure Plan," 2010.
- [8] George W. Bush, "The National Strategy for the Physical Protection of Critical Infrastructure," White House, Washington, D.C.; 2003.
- [9] C. Robinson, J. Woodard, and S. Varnado, "Critical Infrastructure: Interlinked and Vulnerable," *Issues in Science and Technology*, pp. 61-67, September 1999.
- [10] J. Simonoff, C. Restrepo, R. Zimmerman, and Z. Naphtali, "Analysis of Electrical Power and Oil and Gas Pipeline Failures," in *International Federation for Information Processing*, E. Goetz and S. Sheno, Eds. Boston: Springer, 2008, vol. 253, pp. 381-294.
- [11] Cheryl Trench. (2001, March) Oil Market Basics. [Online]. [http://www.eia.doe.gov/pub/oil\\_gas/petroleum/analysis\\_publications/oil\\_market\\_basics/full\\_contents.htm](http://www.eia.doe.gov/pub/oil_gas/petroleum/analysis_publications/oil_market_basics/full_contents.htm)
- [12] P. Parformak, "Vulnerability of Concentrated Critical Infrastructure: Background and Policy Options," Congressional Research Service Report for Congress, Washington, D.C.; 2007.
- [13] S. Rinaldi, J. Peerenboom, and T. and Kelly, "Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies," *IEEE Control Systems Magazine*, pp. 11-25, 2001.
- [14] Michael Chertoff, "National Infrastructure Protection Plan," Department of Homeland Security (DHS), Washington, D.C., 2009.
- [15] William Lowrance, "Of Acceptable Risk," William Kaufmann, Los Altos, 1976.
- [16] Eric Solano, "Methods for Assessing Vulnerability of Critical Infrastructure," Research Triangle Park, NC, 2010.
- [17] John F. Ahearne, "Review of the Department of Homeland Security's Approach to Risk Analysis," Washington, D.C., 2010.
- [18] L. Chai et al., "Social Network Analysis of the Vulnerabilities of Interdependent Critical Infrastructures," *International Journal of Critical Infrastructures*, Vol. 3, No. 4, 2008, pp. 256-273, 2008.
- [19] H. Mohtadi and A. P. Murshid, "Risk of catastrophic terrorism: an extreme value approach.," *Journal of Applied Econometrics*, pp. 24: 537-559, 2009.
- [20] Elisabeth Pate-Cornell and Seth Guikema, "Probabilistic Modeling of Terrorist Threats: A Systems Analysis Approach to Setting Priorities Among Countermeasures," *Military Operations Research*, vol. 7, no. 4, pp. 5-20, December 2002.
- [21] Maria Leung, James Lambert, and Alexander Mosenthal, "A Risk-Based Approach to Setting Priorities in Protecting Bridges Against Terrorist Attacks," *Risk Analysis*, vol. 24, no. 4, pp. 963-984, September 2004.
- [22] W.R. Tobler, "A computer movie simulating urban growth in the Detroit region.," *Economic Geography*, pp. 234-240, 1970.
- [23] P. Longley, M. Goodchild, D. Maguire, and D. Rhind, *Geographic Information Systems and Science*. Hoboken, NJ: John Wiley & Sons, Inc., 2011.
- [24] D.A. Griffith, "Spatial Autocorrelation," Fort Worth, TX, 2009.

# Holistic Network Defense: Fusing Host and Network Features for Attack Classification

J. Ji, G. Peterson, M. Grimaila and R. Mills

Dept. of Electrical and Computer Engineering, Air Force Institute of Technology, Wright Patterson AFB, OH, USA

**Abstract** - *Current defensive systems focus primarily on network data, and are plagued by a high false positive rate and/or duplicate alerts with no ranking of importance. This work presents a hybrid network-host monitoring strategy, fusing data from both the network and the host to recognize malware infections. This research seeks to categorize systems into one of three classes: Normal, Scanning, and Infected. The objective is accomplished by fusing data from multiple network/host sensors, extracting features from network traffic using the Fullstats Network Feature generator and from the host using text mining, using the frequency of the 500 most common strings and analyzing them as word vectors. Hybrid method results outperformed both host only and network only classification. This approach reduces the number of alerts while remaining accurate compared with the commercial IDS SNORT. These results improve the relevance of alerts so that most typical users could understand alert classification messages.*

**Keywords:** Intrusion, Machine Learning, Host, Network, Fusion

## 1 Introduction

The Department of Defense (DoD) officials observed that the number of attempted intrusions into military networks has increased, from 40,076 incidents in 2001, to 43,086 in 2002, to 54,488 in 2003, and to 24,745 as of June 2004 [1]. A newer report, the 2007 E-Crime Watch Survey from CSO (Chief Security Officer) Magazine found the number of security incidents increased for the majority of companies polled from the period between 2005 and 2007 [2]. Network Intrusion Detection Systems (NIDS) have the capacity to detect initial incoming intrusion attempts, but at the sacrifice of a very high false positive rate and an overabundance of relevant true positives as seen by the prolific frequency by which they produce alarms in operational networks [3]. Furthermore, limited throughput often requires sampling of traffic; as it would overwhelm the NIDS to inspect every packet. On host systems, antivirus (AV) software, as an example of a Host IDS (HIDS), is relied upon to prevent malicious downloaded code from being installed or alert an operator when a successful infiltration occurs. Unfortunately, AV scanning of executables for malware detection faces a number of significant problems, one being that current malware programs typically implement run-time packing and

self-modifying code [4]. Detecting, distinguishing and preventing a successful local host infection from the myriad scans, intrusion attempts and AV evasions is ultimately the real goal of effective and intelligent network security applications.

Although there are several NIDS and HIDS implementations, there currently is no commercial IDS and few if any research in this field which fuses data from both of these sources and applies Machine Learning algorithms for classification of the stages and the progression of a cyber-attack. Statistical based methods of intrusion detection should defend better against a zero day attack, which takes advantage of a bug that neither the software's creators nor users are aware of. The work presented in this paper will demonstrate that having information available from both host and network increases classification of attack accuracy much more than each working alone.

## 2 Related Work

Given that signature-based sensors are not feasible for detecting all threats, researchers must consider alternative solutions [5][6]. But the bulk of the research efforts for threat detection thus far focus on developing methods relying solely on network traffic [7], on event logs [8], or on system calls [9]. Unsurprisingly, due to the pressures of finite resources and the push to select the fewest features to process, little or no research efforts thus far have attempted to combine data from various system sensor categories, such as file I/O, network traffic and process meta data, in order to form a holistic picture, giving meaningful knowledge on the status from the system level down to the individual host level.

One example of the creative use of NIDS research is alert correlation. Nearly all of the proposed alert correlation methods are based on syntax-oriented approaches. For example, Wei [10] exploits the semantics of attack behaviors, and presents the semantic vector space model to extract and classify the attack scenarios automatically. Wei uses first order predicate logic (FOPL) and linguistics to classify DDOS computer attacks based on features derived from NIDS alert streams.

Indeed, there is also significant research in the area of consolidating security alarms generated by HIDS into

coherent incident pictures [11]. One major vein of research in intrusion report correlation is that of alert fusion, clustering similar events under a single label [12]. The primary goal of fusion is log reduction, and in most systems similarity is based upon either attributing multiple events to a single threat agent or providing a consolidated view of a common set of events that target a single victim.

The only published paper available from the literature search on a hybrid methodology is by Depren [13], the hybrid methodology was tested on the flawed MIT Lincoln Labs KDD (Knowledge Discovery and Data Mining) 99 dataset which entirely consists of TCP dumps and no host data. Depren does not explain how the host misuse detection module contributes to the test results, since host information is not part of the KDD set.

The KDD 99 dataset is the most commonly used standard for researchers to benchmark their performance but has been under fire for not being a true representation of how a system performs in a real operating network. McHugh [14] even provides a critique of the Lincoln Lab's procedures in the creation of the 1998 (and some of the 1999) IDEVAL dataset. His main criticisms included: that their assumptions were made without corroborating evidence or descriptions, that the traffic density and uniformity of the dataset, their odd ROC curves used for analysis, and their procedures for scoring IDSs make little or no sense with respect to the intrusion detection field. In conclusion, McHugh finds the IDEVAL dataset a step in the right direction in providing intrusion detection testing data, but finds many errors in the process that make the dataset only of limited utility for testing intrusion detection systems.

To shed light on how data collection affects algorithm performance, Maxion and Tan [15] proposed a system for benchmarking anomaly-detection systems, investigating whether the regularity of a data set influences the effectiveness of the IDS. Artificial datasets were generated with a given entropy value between 0 and 1. Results confirm that regularity of the data set affects the effectiveness of the IDS.

Testing of methodologies at least requires a dataset containing two types of packets: a clean dataset for training to establish a baseline and a dataset including labeled attacks for testing performance. Such a dataset is difficult to obtain due to privacy laws. Packet traces cannot contain personally identifiable information in its payload and must be scrubbed of user names and passwords as a minimum measure. This is currently is a tedious process which could easily overwhelm any research effort. This work tries to address the lack of standardized and publically available data sets by creating its own host and network datasets in simulated environments, collecting from both network and host sources with clients set up to simulate normal and infected activity while surfing the internet. Another class of scanning activity data was obtained

from the 2010 Cyber Defense eXercise at the Air Force Institute of Technology (AFIT).

### 3 Methodology

The host features were extracted and put into a CSV Comma Separated Values (CSV) file using text mining tools available in Weka [16] from text files derived of the following SysInternals tools [17] polling in 15 second intervals: DLLs, Handles, LogonSessions, Netstat, Processes, PsInfo and Services. The CSV file containing the features that define the raw network data are derived from packet captured by Wireshark as PCAP (short for Packet Capture) files, using an in-house updated version of Andrew Moore's [18] Fullstats packet feature generator with the input being those PCAPs. The two files were then manually integrated by MS Excel, based on closeness in time space, discarding any extras to preserve a one to one relation between a network feature set and host feature set. The final CSV file was tested and trained using a randomly proportioned 20% for training and 80% for testing purposes. The algorithm used was Weka SMO, which is a version of Support Vector Machine, and the reason it was chosen is discussed in detail in section 4, results and analysis. The data collection process involved two environments, The Cyber Defense eXercise and a Vista Machine surfing the internet. Both host and network attributes were formatted into CSV format, using 248 numerical packet metrics for the network data and 500 most frequent words as word vector attributes in the case of the SysInternals host data text files.

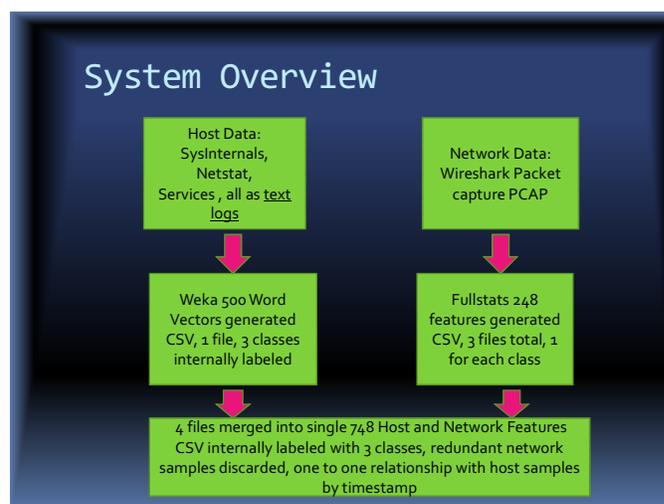


Fig 1. System Methodology Overview

The reason for mixing the two environments of data collection was that the AFIT1 team succeeded in preventing red team infiltration, but as a result, data was only of one category: scanning, defined as the attacker having mapped part of or the entirety of the network for the purpose of directing later exploit attempts at specific IP addresses. The CDX data for the scanning category used in the final analysis was from the network DNS server running Windows 2003

server. To attempt to remedy the lack of other categories of infection stages, additional data for the normal and infected classes were obtained on a test network consisting of a Windows Vista host connected to the Internet using the same Wireshark and SysInternals collection tools. The host was infected via visits to known malicious websites as listed in the malware domain list and infection was verified by antivirus software. The normal class is defined by the random exploits a host faces while surfing the world wide net and though it may encounter bad packets, it is not being specifically targeted and the antivirus is operational and running. Infected is defined as the state of the host when a Trojan has been downloaded and confirmed by the antivirus program to have replicated itself into the file system. This bulleted list shows how the Vista client was operated to simulate user activity and how data was collected:

- Host Operating System: Windows Vista Service Pack 2
- Applications: Windows Media Player in continuous MP3 play loop, Microsoft Internet Explorer 8.0 running automatic page refresh in 2 minute intervals using [www.lazywebtools.co.uk/cyclor.html](http://www.lazywebtools.co.uk/cyclor.html), AVG anti-virus Resident Shield OFF in infected case, ON in normal case
- Data Collection Tools (sensors): Always on: Wireshark, Snort IDS with standard rule set; 15 minute intervals: Win32dd memory capture; 1 minute intervals: C:\WINDOWS\System32\drivers\etc\services; SysInternals: Listdlls, LogonSessions, Handle, Pslist, NetStat; Once per logon: PsInfo

A separate feature CSV file was generated from the packet capture by Wireshark using a perl script originated by [41] Andrew Moore of Cambridge University, [fullstat.v1.0.tgz](http://fullstat.v1.0.tgz) (<http://www.cl.cam.ac.uk/research/srg/netos/brasil/downloads/index.html>). Moore summarizes the extracted features and its been noted that some of the features are correlated, in other words, not all are independent.

Part of this research also identified the best machine learning algorithm to analyze the collected data, comparing the performance of Self Organizing Maps, Learning Vector Quantization, and Support Vector Machines on a much smaller set of data from Andrew Moore's research [41], an overview of how these algorithms work is generalized here.

### 3.1 Self Organizing Maps

SOMs are an unsupervised method for visualizing data of high dimensionality [19]. The output of the algorithm form clusters of similarity. Thus, it can be a way to help analyze data when knowledge of how many classifications there should be is not available beforehand. One important component of SOM is the weight vectors or "neurons", these vectors contain the input data as well as its location in the

lattice space. Then, via a very simple algorithm, the neurons compete, the winners being the ones that best represent the data.

Calculating SOMs can be computationally expensive depending on the size of the lattice and the dimensionality of data, so there are some methods of initializing the weights such that samples which are known to be different start off far away. This can save a significant number of iterations in order to produce a good map.

### 3.2 Learning Vector Quantization

The basic LVQ approach is based on a standard trained SOM with input vectors and weight vectors [20]. The new factor is that the input data points have associated class information. This allows us to use the known classification labels of the inputs to find the best classification label for each weight vector. For example, by simply counting up the total number of instances of each class for the inputs within each classification cell, a new input without a class label can be assigned to the class of the cell it falls within.

### 3.3 Support Vector Machine

SVMs are one of the more advanced and accurate methods of data classification, however like many computational challenges, it is a trade-off of accuracy for speed [21]. SVMs are not yet primed for real time applications, especially for the high volume task of network and host data analysis. A general explanation of the theory is presented in this section.

The general idea behind SVMs is that the original feature input space which is difficult to separate can be mapped to a higher-dimensional feature space where the training set becomes more easily separable. With this mapping, the discriminant function is now:

$$g(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b = \sum_{i \in SV} \alpha_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b \quad (1)$$

In the above,  $\mathbf{x}$  is the feature vector,  $\mathbf{w}$  is the class. There is really no need to know this mapping explicitly, because only the dot product of feature vectors in both the training and test is used. Thus, a kernel function is defined that corresponds to a dot product of two feature vectors which maps the samples into an expanded feature space. For example, a linear kernel function is:

$$K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \quad (2)$$

Some other commonly used kernels are polynomial, Gaussian and sigmoidal [18]. Unfortunately the selection of the best kernel is a trial and error process [18].

To solve for the optimal hyperplane in the linearly separable case, Lagrangian multipliers are introduced: (Lagrangian Dual Problem)

$$\text{maximize } \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (3)$$

Such that  $0 \leq \alpha_i \leq C$  and  $\sum_{i=1}^n \alpha_i y_i = 0$

The solution of the discriminant function is

$$g(\mathbf{x}) = \sum_{i \in SV} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \quad (4)$$

get  $b$  from  $y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 = 0$ , where  $\mathbf{x}_i$  is support vector

The optimization technique then is the same as for the large margin classifier. The solution has the form:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i = \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i \quad (5)$$

The basic SVM algorithm is as follows:

1. Choose a kernel function
2. Choose a value for  $C$
3. Solve the quadratic programming problem (many software packages available)
4. Construct the discriminant function from the support vectors

## 4 Results and Discussion

The Andrew Moore dataset labeled entry 02, entry 04, entry 08 and entry 09 were used to test the Machine Learning algorithms in Weka, entry 04 and 08 were used as the training/model building sets and entry 02 and 09 were used as their respective testing sets. Since the results showed similar trends, only the entry 02/04 experimental pair results are presented. The Moore data set contains 377,526 samples of network flows, 248 features, and 12 classes, whose features include nominal, discrete, continuous, missing and noisy values. The samples of the data are restricted to bidirectional Transmission Control Protocol (TCP) flows. A portion of the data set is employed for this work since the original Moore data set consists of too many network flows to handle in a reasonable amount of time and the researcher encountered heap space issues with the full Moore data set. The reduced data set consists of 40,858 flows out of the 377,526 flows. The reduction was due to the fact that the majority of the flows consist of email and World Wide Web traffic and so had to be reduced to preserve a more equivalent ratio with respect to the other classes. The games class was removed because there were only 8 instances, and due to restrictions should not appear in enterprise network traffic.

The Moore dataset is used to perform a comparison test of the Self Organizing Map (SOM), Learning Vector Quantization (LVQ), and Support Vector Machine (SMO) classifiers which resulted in the SVM SMO as the most accurate classifier and ergo became the chosen classifier for final testing on the experimental CDX/Vista data.

Afterwards, testing of the Weka SMO classifier on the CDX/Vista data provided a comparison between pure HIDS (Host data set prior to merging features), with NIDS (Network data set prior to merge) and finally, with that of one form of the hybrid IDS.

### 4.1 ML Algorithm Comparison

In the Self Organizing Map experiment, the false positive rate for the largest class WWW (web browsing traffic) was very high, ranging from 73-93% testing the model on the various Moore data sets. Also, the size of the training data set used to create the model that corresponded in size with the test set generally led to better classification accuracy and this was also true of the other algorithms.

In the LVQ experimental results, LVQ 3 performed much better than LVQ 2.1, 90% versus 78% respectively. LVQ 3 had a higher true positive rate and lower false positive rate. For LVQ 2.1, two best match units are selected and only updated if one belongs to the desired class and one does not, and the distance ratio is within a defined window [20]. The difference in LVQ 3 is that even if both best match units are of the correct class, they are updated but adjusted using an epsilon value (adjusted learning rate instead of the global learning rate). Another note in using Weka is to turn voting off, or else it would basically put everything into the class WWW because of its overwhelming sample proportion relative to the other classes.

Overall, LVQ 2.1 turned out to be the worst performer, with a low true positive rate and a high false positive rate. The next would be the SOM which in Weka does not provide a means of doing magnification; magnification control in SOMs refers to the modification of the relationship between the probability density functions of the input samples and their prototypes (SOM weights). This was a problematic issue due to the nature of the data as not all the classes were even close to being equally represented. Forbidden magnification would've given more representation to smaller classes. LVQ 3 did a bit better than SOM but still had a very high false positive rate for the largest class. The best but most time consuming of all methods investigated was the SVM. Weka has a binary implementation called SMO, which means additional coupling and pair-wise classification and comparison steps, on the order of  $(n \text{ choose } 2)$  were required. SVM had the highest accuracy (96-99% range on the data set used for test), and lowest false positive rate of all the methods investigated. WWW class still had the highest FP rate, but it was only

3.3% for the training set 4 generated model tested on data set 2 and 8.6% for the 8 model on data set 9.

This investigation concluded that it is best to use SVMs whenever the application doesn't require real time results. SVMs may still be feasible in the application of network security if one could reduce the number of classes which would result in a speed boost. But as things are, SVMs lag the rest in terms of computational and time resources. However, since real time is not the point of this investigation, it was decided that SVM aka Weka SMO would be used to classify processed data from both the CDX and Vista collections.

## 4.2 Host only vs Network only vs Hybrid

The SMO parameters were the default settings in Weka using the combined data set containing all 500 and 248 host and network features, Table 1 lists the available options and the specific settings selected for the SMO algorithm. SMO was trained on 20% of the data set and the remainder 80% was used for testing. The parameter settings displayed are from the 3.6.2 version of Weka which is a more recent version than the one used to determine the best machine learning algorithm used in the preceding section.

Table 1. SMO Parameters in Weka

Parameter	Value
Build Logistic Model	False
C	1.0
Checks Turned Off	False
Epsilon	1.0E-12
Filter Type	Normalize Training Data
Kernel	PolyKernel -C 250007 -E 1.0
NumFolds	-1
Random Seed	1
Tolerance Parameter	0.001

It can be immediately noted that the scanning results are highly distinguishable from the normal and infected classes. Their confusion matrices are listed in Table 2 thru 4. A confusion matrix is a visualization tool typically used in supervised learning. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The rows in the confusion matrix are the labels of the samples, and the columns are the classification results. Host only achieved an accuracy of 76%, Network only achieved 87% and the Hybrid achieved an accuracy of 99%. Distinguishability between the Scanning class and the other two classes was high throughout these three scenarios; but, this is not a testament to the quality and effectiveness of the classification algorithm or of the feature extractor but rather of the fact that this data was collected in a separate environment, namely the CDX network. The other two remaining classes were gathered later from the Vista machine and thus share many more similar features that cause

the Weka SMO classifier confusion as to what to differentiate on and increases misclassifications.

Table 2. Host Only Classification Results Confusion Matrix

	Infected	Normal	Scanning
Infected	83	78	0
Normal	26	117	0
Scanning	0	1	138

It is safe to assume from Table 2 and also in the results that follow that the percentage of correctly classified instances is actually inflated due to the artificially high accuracy of the scanning class detection. Averaging just the Infected and Normal detection rate would yield a more representative accuracy metric of 66.7%.

Table 3. Network Only Classification Results

	Infected	Normal	Scanning
Infected	402	1593	7
Normal	876	2672	4
Scanning	0	1	10373

Averaging the Infected and Normal detection rate for Table 3 would yield a more representative accuracy metric of 73.4%. On the surface, this is clearly better than the results of the host data; however, one should consider the possibility that the type of infection, by trojan malware in this case, could leave a larger footprint or effect more statistically relevant change in network activity when compared to the host's.

Table 4. Hybrid Host and Network Classification Results

	Infected	Normal	Scanning
Infected	159	0	2
Normal	0	140	3
Scanning	0	1	138

Averaging the Infected and Normal detection rate in Table 4 would yield a more representative accuracy metric of 98.4%. This result outperformed host only classification by 31.7% and network only classification by 25%. Text mining is typically used to look at frequencies of word strings and is often used to try to identify natural language features like authors' writing style or language. Because effective HIDS depends heavily on event correlation, the text mining approach did not factor in cause and effect and looked only at the string structure. But this result is a positive indicator that something that seems as un-intuitive as textual frequencies of host data from snapshot sensors contributes to greater accuracy in malicious activity detection. It also lends credence to the hypothesis that if numerical metrics of behavioral information rather than text frequencies could be garnered, it may significantly improve detection while vastly

decreasing the number of attributes; and, thus save on processing times. Also, since attacks tend to originate more on the network side, greater accuracy possibly would've been achieved by placing more bias towards the network features. The final set of features trained and tested in Weka contained 500 attributes from the host data and only 248 attributes from the network data. It's conceivable that the number of attributes can be lowered on the host side and still preserve this level of performance. This is an avenue that can be considered in future work.

### 4.3 Comparison of Hybrid to Snort

The analysis of the host data covered approximately 15 minutes of operational time of data gathered from seven of the SysInternals tools. The size of the data was 17.2MB. Taking a look at the SNORT alerts, there was 119 SNORT alerts that contained a reference to 10.1.30.5, which was the IP address of the DNS (Domain Name) server contributing the CDX scanning data when the DNS was mostly likely discovered by NSA and being actively scanned. This is something that would likely get lost in the sea of alerts of all the other nodes of the network. Most of the alerts are purely repetitious and thus redundant.

The alert log contains over a hundred of the exact same alert, yet the alert reveals little information as to the true nature of what's going on between the client(s) and the DNS host.

There were 5 SNORT alerts for each of the data sets Normal and Infected and these were regarding SHELLCODE EXECUTION, attributable to the .bat scripts used to start the SysInternals sensors to collect snapshot data. Effectively, SNORT got 0% true positives and false positives, and fails to capture any relevant information in the VISTA experiment surfing the real World Wide Web.

Comparing SNORT performance to the performance as tested in the preceding section is not completely fair. SNORT is fine grained and intentionally designed to analyze each packet or sequence of packets it sniffs; it is not meant to interpret all the alerts together as a whole to give a classification decision. However, just based on this rough description of its output on the network data, it is clearly performing a dismal job, either missing alerts or overwhelming the user. The goal is to move towards a system where a novice administrator should be able to identify a security breach as it unfolds, yet SNORT is still a system that requires high level training and experience to use effectively.

## 5 Conclusions

The results of this investigation favors the idea that statistical analysis using text based data mining in combination with network traffic flows is a more effective method for intrusion detection than either host or network detection alone. At present, integrating host data to network

data may achieve the highest effectiveness by augmenting existing event based NIDS systems like SNORT or BRO; for example, adding an interface that allows it access to relevant host data to reduce alerts from the age old rules set checking method. A few already are trying this approach, namely McAfee Enterecept Host IPS/IntruShield Network IPS and ISS Proventia, but the trend has been slow to be adopted mainstream. Trying to integrate these two vantages in a completely new platform using machine learning techniques is still a ways off from everyday practicality. Machine learning based applications continue to be resource intensive ones.

The highest priority task for future work is to produce a labeled data set that contains a broad continuum of attack stages from both host and network data gathered together in the same environment. A methodology for this collection should be developed so that data that must be collected at different times and in different environments can still be compared.

Also of importance, is to identify an auxiliary method for associating host and network data. Timestamps are not the most accurate one-to-one associations and better "triggers" are needed.

An application that can deal with the high volume of data in "real time" and generate features "on the fly" that could compress the amount of analytical data would be highly sought after. At the start of this research, the initial time hog was thought to be running the machine learning algorithm. This proved true but additionally, both feature generation and large file transfers took many more hours than anticipated. Research that can bring tools for post mortem forensics into live action would greatly complement the existing means that System Administrators have to identify network breaches.

Further, a periodic maintenance update to the Fullstats attribute generator is needed. Since there really is no other tool that can pull as many features from a PCAP file, such an application is valuable in the search for the most important features or combinations of features which could significantly lessen the processing time for classification algorithms. Developing this tool to add a GUI interface or to integrate into Weka or MATLAB could open this field up to both seasoned researchers and novice investigators.

Lastly, due to the explosion in bandwidth of cell phone networks and large area WI-MAX, focus should be shifting to making intrusion detection tools more ubiquitous and able to function on a variety of mobile devices, more utility often comes with more vulnerability. Thus, it needs to be determined what features are most important for the host if it is a wireless media device that may contain other channels of communication such as 3G/4G, GPS or satellite radio. As communication starts pushing the barriers beyond IP packets, normal baselines shift, new optimized feature sets need to be

found and security becomes an even greater challenge to keep up with.

## 6 References

- [1] Clay, Wilson. "Botnets, Cybercrime, and Cyberterrorism: Vulnerabilities and Policy Issues for Congress," Washington, D.C.: Congressional Research Service, 25, (January 29, 2008).
- [2] W. Yan "Network Attack Scenarios Extraction and Categorization by Mining IDS Alert Streams" *Journal of Universal Computer Science*, vol. 11, no. 8, 1367-1382, 2005.
- [3] CSO Magazine. "OVER CONFIDENCE IS PERVERSIVE AMONGST SECURITY PROFESSIONALS," 2007 E-Crime Watch Survey, 26 Feb 2011. [www.cert.org/archive/pdf/ecrimesummary07.pdf](http://www.cert.org/archive/pdf/ecrimesummary07.pdf)
- [4] Gu, Guofei, Phillip Porras, Vinod Yegneswaran, Martin Fong, and Wenke Lee. "BotHunter: Detecting malware infection through ids-driven dialog correlation," *Proceedings of the 16th USENIX Security Symposium*, (2007).
- [5] Kolbitsch, Clemens, Paolo Milani Comparetti, Christopher Kruegel, Engin Kirda, Xiaoyong Zhou, and Xiaofeng Wang. "Effective and efficient malware detection at the end host," *USENIX Security Symposium*, (August 2009).
- [6] Biles, Simon. "Detecting the unknown with snort and statistical packet anomaly detection engine (SPADE)," *Computer Security Online Ltd.* 5 January 2011. <http://www.computersecurityonline.com/spade/SPADE.pdf>
- [7] Paxson, Vern. "BRO: A System for Detecting Network Intruders in Real Time," *Computer Networks*, 31 (23): 2435-2463 (1999).
- [8] Haag, Charles R., Gary B. Lamont, Paul D. Williams, and Gilbert L. Peterson. "An artificial immune system-inspired multiobjective evolutionary algorithm with application to the detection of distributed computer network intrusions", *GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation.* (2007).
- [9] Makanju, A.A.O., A.N. Zincir-Heywood, and E.E. Milios. "Clustering event logs using iterative partitioning," *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1255-1264. ACM New York, NY, USA, (2009).
- [10] Ning, P., Y. Cui, and D. Reeves. "Constructing attack scenarios through correlation of intrusion alerts." *Proceedings of Computer and Communications Security*, (2002).
- [11] Wei, Yan. "Network Attack Scenarios Extraction and Categorization by Mining IDS Alert Streams," *Journal of Universal Computer Science*, 11(8): 1367-1382 (2005).
- [12] Internet Security Systems White Paper, Network- vs. Host-based Intrusion Detection, A Guide to Intrusion Detection Technology. 5 January 2011. [http://www.documents.iss.net/whitepapers/nvh\\_ids.pdf](http://www.documents.iss.net/whitepapers/nvh_ids.pdf)
- [13] Valdes, A., and K. Skinner. "Probabilistic alert correlation," *Proceedings of Recent Advances in Intrusion Detection (RAID)*, 54-68, (2001).
- [14] Depren, O., M. Topallar, E. Anarim, and M.K. Ciliz. "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks," *Expert Systems with Applications*, 29(4): 713-722, (2005).
- [15] McHugh, J. "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA IDS evaluations as performed by Lincoln Laboratory," *ACM Transactions on Information and System Security*, 3(4) (November 2000).
- [16] Maxion, R.A., and K.M.C. Tan. "Benchmarking Anomaly-Based Detection Systems," *1st International Conference on Dependable Systems & Networks*, 25(28): 623-630 (June 2000).
- [17] Lee, J.B. "WEKA Classification Algorithms," 07 Sept 2010. <http://weka.classalgos.sourceforge.net/>
- [18] Russinovich, Mark. "Microsoft SysInternals Suite" 2 Feb 2011, <http://technet.microsoft.com/enus/sysinternals/bb842062.aspx>
- [19] Moore, Andrew W. and Denis Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques," *Proceedings of the ACM SIGMETRICS*, Banff, Canada, (June 2005) 3 Mar 2011. <http://www.cl.cam.ac.uk/research/srg/netos/brasil/data/index.htm>
- [20] Germano, T. "Self-Organizing Maps," 07 Sept 2010. <http://davis.wpi.edu/~matt/courses/soms/>
- [21] Bullinaria, J.A. "Learning Vector Quantization (LVQ): Introduction to Neural Computation: Guest Lecture 2," 07 Sept 2010. [http://www.cs.bham.ac.uk/~pxt/NC/lvq\\_jb.pdf](http://www.cs.bham.ac.uk/~pxt/NC/lvq_jb.pdf)
- [22] Gu J. "An Introduction of Support Vector Machine." (16 Oct 2008), 5 Nov 2010. <http://www1.cs.columbia.edu/~belhumeur/courses/biometrics/2009/svm.ppt>

## **SESSION**

# **NOVEL APPLICATIONS AND ALGORITHMS + METHODS RELATED TO: CYBER SECURITY, SECURITY POLICY, ATTACK DETECTION, RISK MANAGEMENT, AUTHENTICATION, AND ENCRYPTION**

**Chair(s)**

**Prof. Hamid R. Arabnia**



## KEYNOTE LECTURE: The Nature of Cyber Security

Prof. Eugene H. Spafford  
Purdue University CERIAS, USA

### Abstract:

There is an on-going discussion about establishing a scientific basis for cyber security. Efforts to date have often been ad hoc and conducted without any apparent insight into deeper formalisms. The result has been repeated system failures, and a steady progression of new attacks and compromises. A solution, then, would seem to be to identify underlying scientific principles of cyber security, articulate them, and then employ them in the design and construction of future systems. This is at the core of several recent government programs and initiatives. But the question that has not been asked is if "cyber security": is really the correct abstraction for analysis. There are some hints that perhaps it is not, and that some other approach is really more appropriate for systematic study - perhaps one we have yet to define.

In this talk I will provide some overview of the challenges in cyber security, the arguments being made for exploration and definition of a science of cyber security, and also some of the counterarguments. The goal of the presentation is not to convince the audience that either viewpoint is necessarily correct, but to suggest that perhaps there is sufficient doubt that we should carefully examine some of our assumptions about the field.

### Biography:

Eugene Howard Spafford is a Professor in the Purdue University. He is historically a significant Internet figure. He is renowned for first analyzing the Morris Worm, one of the earliest computer worms, and his prominent role in the Usenet backbone cabal. Spafford was a member of the President's Information Technology Advisory Committee 2003-2005, has been an advisor to the National Science Foundation (NSF), and serves as an advisor to over a dozen other government agencies and major corporations. Spafford attended State University of New York at Brockport for three years and completed his B.A. with a double major in mathematics and computer science in that time. He then attended the School of Information and Computer Sciences (now the College of Computing) at the Georgia Institute of Technology. He received his M.S. in 1981, and Ph.D. in 1986 for his design and implementation of the original Clouds distributed operating system kernel. During the early formative years of the Internet, Spafford made significant contributions to establishing semi-formal processes to organize and manage Usenet, then the primary channel of communication between users, as well as being influential in defining the standards of behavior governing its use.

# An IT Security Investigation into the Online Payment Systems of Selected Local Government Councils in WA

Sunsern Limwiriyakul<sup>1</sup> and Craig Valli<sup>2</sup>

SECAU – Security Research Centre, Edith Cowan University, Perth, Western Australia  
slimwiri@our.ecu.edu.au<sup>1</sup> and c.valli@ecu.edu.au<sup>2</sup>

**Abstract** - *The paper examined information technology (IT) security of the online payment systems at the three selected local government councils in Western Australia (WA). The scope of the study included the architecture, the infrastructure devices, port scanning and vulnerability testing of the online payment system server, as well as online database application auditing. Several industry and national benchmarking standards were utilized in this study. The investigative work was also carried out with the intention of establishing a security framework which could be easily implemented or adapted to suit any other councils or organizations with a similar online payment system.*

**Keywords:** framework, IT security, online payment system, vulnerability testing

## 1 Introduction

Online payment services have become an available payment option to WA's residential community. There are currently 81 WA councils who provide online payment service to residents as an alternative payment option. However, only nine of these 81 WA councils have their own in-house online payment systems which allow their residents to pay their rates, infringements and registrations online via secure sockets layer (SSL) encryption connections. The other 72 WA councils do not have their own online payment systems and use third-party online payment services such as BPOINT [3] and Postbillpay [1] instead.

One of the major concerns of providing online payment services is Information and Communications Technology (ICT) security which covers both the confidentiality and privacy aspects when dealing with the councils' residents sensitive information. The concern for a council in providing these services relate to the transmission of the users'

information over the Internet to the councils' online payment system network, keeping the user information secure as well as providing 24x7 system availability (personal communications, 2008, 2009, 2010).

ICT security of the current online payment systems at the three selected WA councils were investigated in order to identify whether the systems were implemented securely based on industry and national security standards. Furthermore, all the test data were gathered on a real-time basis at all of the three selected councils.

In addition, the outcomes of this study was the provision of a security recommendations report of the online payment system based on the suggested implementation framework, analyses, result finding as well as security consultations made to each individual council for which the study was carried out.

## 2 Implementation Framework

The implementation framework was constructed using various testing techniques which included Section C of Open Source Security Testing Methodology (OSSTMM) 2.2 [5], the Centre for Internet Security (CIS) – Security Configuration Benchmark for MS SQL Server 2005 version 1.2.0 [2], Information Systems Security Assessment Framework (ISSAF) version 0.2.1 [9], National Institute of Standards and Technology (NIST) and other relevant security information obtained from a variety of sources such as books, journals, personal interviews as well as World Wide Web (WWW).

There were five stages to the online payment system implementation framework including (1) network surveying; (2) online payment system infrastructure review; (3) services and system identification, port scanning and vulnerability

detection of the online payment system servers; (4) vendor security benchmarking on backend database server; and (5) online payment system security policy review. See Figure 1 for more details.

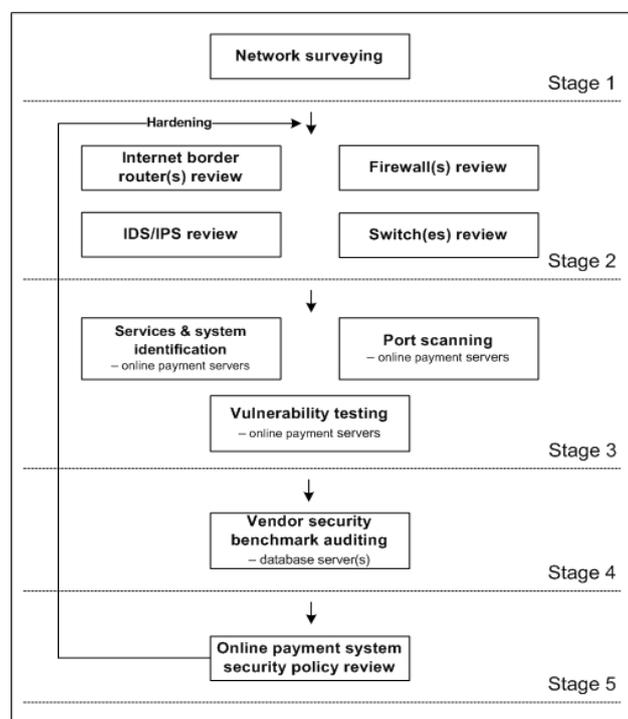


Figure 1. An implementation framework

Stage 1: This network surveying stage was used to collect information on the selected council's online payment system such as overall internetwork architecture diagram including the Demilitarized Zone (DMZ) infrastructure connectivity of the online payment system as well as a review of the overall online payment system architecture.

Stage 2: The online payment system infrastructure review was conducted to identify the specification as well as configuration codes of the online payment system related internetwork infrastructure devices at all the three selected councils. The internetwork infrastructure devices included the Internet border router (Councils A and B), the IDS/IPS, the firewall(s), the DMZ switch(s) and the reverse proxy server (Council C only).

Stage 3: In this stage, two network scanning tools, NMAP with GUI standard (open source Zenmap version 5.0) [7] and GFI LANguard version 9.0 [4] were used to perform system and services identification, port scanning and vulnerability testing. Both the network scanning tools were scanned with the comprehensive or full scan setting option in order to collect as much as possible data.

Stage 4: This testing stage was intended to test the security configuration of each selected council's online backend database application software. At all three selected councils, the backend database application used in online payment system was MS SQL Server 2005. Nevertheless, each of the selected councils used slightly different configuration settings. Consequently, the CIS – Security Configuration Benchmark for MS SQL Server 2005 version 1.2.0 which was used for the tests, was modified to suit each of the selected council's online backend database environments.

Stage 5: This stage was related to a review of the IT security policy in relation to the online payment system. The IT security policy involved the authorization, the authentication and the accounting of the online payment system in each of the three selected councils. In addition, the firewall, IDS/IPS servers, switch and other related server policies were also reviewed.

### 3 Current online payment systems' architectures

#### *Council A*

The online payment system of Council A (CoA) could be considered a three-tier client-server architecture. It consists of a frontend web, and application and backend database servers [8]. There are three components in a three-tiered client-server architecture which include the presentation logic, application logic and data logic [8]. Both the CoA-DMZ-Epathweb and the CoA-application servers perform all of the application logic which include application synchronization, command processing and calculations. In addition, the CoA-database server performs all the data logic which handles the storage and retrieval of information from the council's online payment database. The council uses a well known third-party online payment application called Pathway (with online payment features) [6] for its online payment system.

The CoA-DMZ-Epathweb acts as a frontend web server which is located in the council's DMZ area whereas the CoA-application and CoA-database servers are located within the council's internal network. The CoA-DMZ-Epathweb server is synchronized with the CoA-application server through the council's internetwork system via the HTTP port. Furthermore, the CoA-application server interacts with the backend online database server, the CoA-Database via the assigned SQL database port (TCP port 2134).

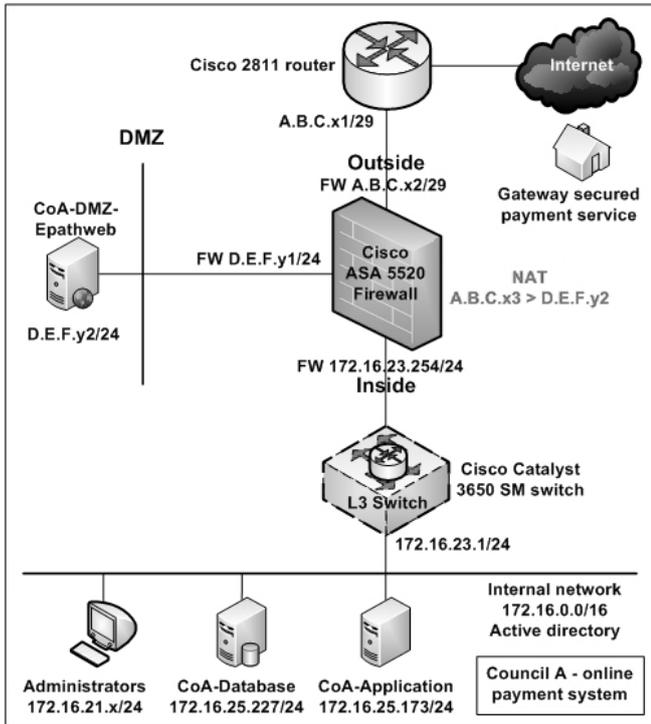


Figure 2. Council A's current online payment network diagram

In terms of infrastructure, the council's internetwork consists of a Cisco 2811 internet border router, a Cisco ASA (5520) firewall and three Cisco Catalyst 3650SM switches as denoted in Figure 2.

**Council B**

The infrastructure of Council B (CoB)'s online payment system can be also considered a three-tiered client-server architecture. It consists of two servers which are the CoB-DMZ-web and the CoB-database servers. The CoB-DMZ-web server performs the application logic function by serving both the frontend web and application servers. The CoB-database server performs the data logic function in dealing with all the database transactions. The CoB-DMZ-web server is located in the council's DMZ network whereas the CoB-database server is located in the council' internal network. Furthermore, Council B uses its own in-house built application software for its online payment system.

In addition, both HTTP and HTTPS are currently allowed internally and externally directly to the CoB-DMZ-web server. The CoB-DMZ-web server also communicates with the CoB-database server via a SQL port.

In terms of internetwork and DMZ, the council's infrastructure which consists of one internet border router (Cisco 2811), two firewalls (CheckPoint Firewall 1 – UTM-1 272) and one switch (Cisco Catalyst 3750G). The switch provides network connectivity for the council's internetwork, DMZ as well as the internal network as depicted in Figure 3.

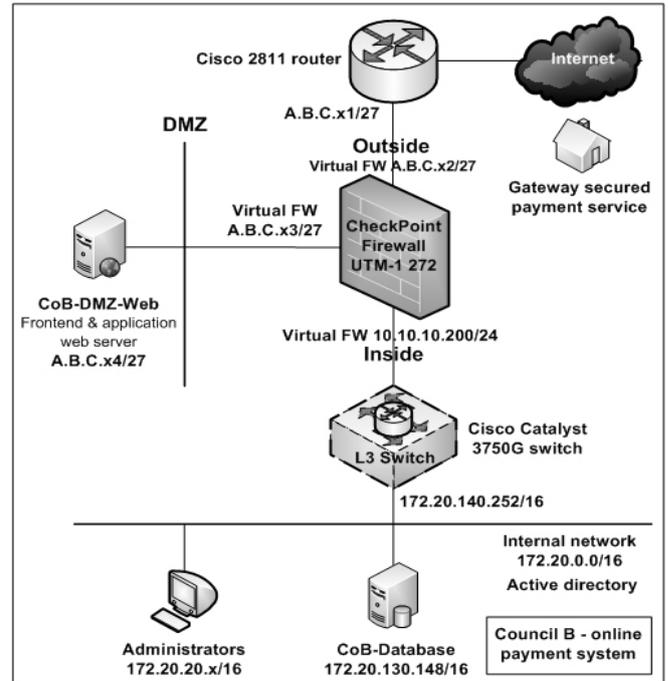


Figure 3. Council B's current online payment network diagram

**Council C**

Council C (CoC)'s online payment system architecture can be also considered as a three-tiered client-server similar to both Councils A and B. There network is constructed of three servers the CoC-DMZ-Epathweb, the CoC-application and the CoC-database servers. The CoC-DMZ-Epathweb acts as a frontend web server whereas the CoC-application server performs the application logic function and the CoC-Database server performs the data logic function.

The CoC-DMZ-Epathweb server communicates with the CoC-Application server through the council's reverse proxy server (MS ISA 2006) via the IP port whereas the CoC-application server interacts with the CoC-Database server via a standard SQL port as depicted in Figure 4.

In terms of infrastructure, the council's network consists of two firewalls (Juniper: SSG-350M) and one switch (HP: E5412zl). The switch serves the internetwork, the DMZ as well as the internal networks. Furthermore, the switch also functions as an internal central core switch which provides connectivity with all the council's desktop switches.

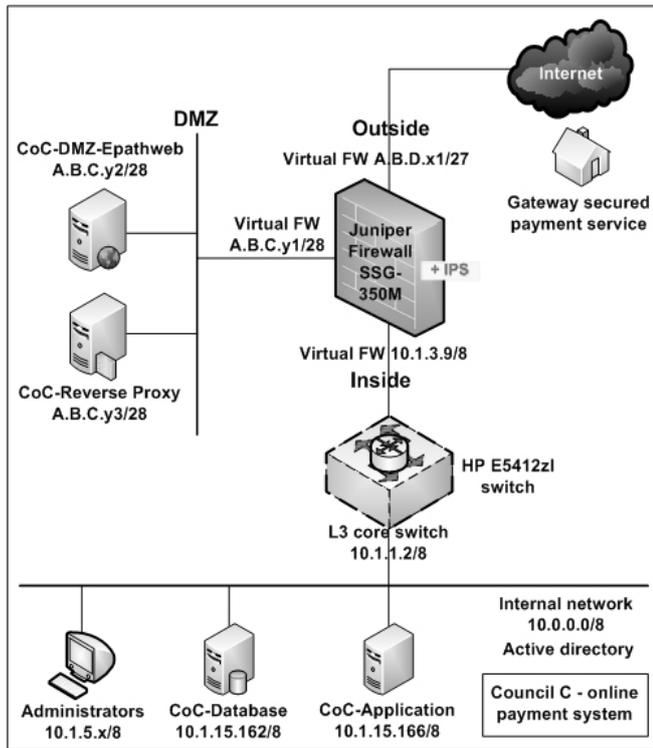


Figure 4. Council C's current online payment network diagram

**Encryption and digital certificates**

All three selected councils use a third party security gateway to provide secure payment for their residents. The corresponding encryption and digital certificate information for each council is presented in Table 1.

Table 1. Overall encryption and digital certificates details of the three

Council name	SSL encryption	Certificate Authority (CA)	Certificate validation type
A	128 bit	Thawte	Standard
B	128 bit	RapidSSL	Standard
C	128 bit	VeriSign	Extended

All the three selected councils use 128 bit SSL encryption. Whereas, Councils A and B use standard validation SSL certificates, Council C has deployed an extended validation

SSL certificate which "gives high-security Web browsers information to clearly identify" [10, p. 1] its website organizational identity. For example, by using Microsoft Internet Explorer (IE) version 7 or higher to go to the Council C online payment website, the URL address bar of the IE7 will display in green color. In addition, by using an extended validation may increase consumer confidence in online business activities [10].

**4 Data analysis and results finding**

The implementation framework as explained in methodology section was used for analyzing and testing the collected data. For simplicity, the results and findings for each of the five stages in each online payment system, are presented in a table format where appropriate.

**Stage1:** Network surveying – The related technical network document of online payment systems of all the three selected councils were found partly up-to-date. For example, the firewall and internetwork technical documents at all three selected councils were not up-to-date.

**Stage 2:** Internetwork infrastructure reviews  
NA represents not applicable

Table 2. Internet border router review

Online payment system infrastructure review: Internet border router			
Facility descriptions	CoA	CoB	CoC
Internet border router deployed	Yes	Yes	No
Internet border router redundancy/alternative internet link deployed	No	No	No
Router configuration against IP spoofing attacks	No	No	NA
Access Control List (ACL) rule to allow ONLY permitted related email protocols for in/out	No	No	NA
Use of best practice user name and/or password and strong password encryption (MD5)	Yes	No	NA
Administration of the router via unsecured communications (HTTP and Telnet) permitted	Yes	Yes	NA

Table 3. IDS/IPS review

Online payment system infrastructure review: IDS/IPS			
Facility descriptions	CoA	CoB	CoC
IDS feature/device in existence	Yes, on the internet border router	Yes, on the internet border router	No
IDS enabled	No	No	NA
IPS feature/device in existence	Yes, on the firewall	No, but can be added on to the firewall	Yes, on the firewall
IPS enabled	No	NA	Yes
IPS depth inspection on HTTP and HTTPS enabled	NA	NA	Yes
IDS feature/device in existence	Yes, on the internet border router	Yes, on the internet border router	No

Table 4. Firewall review

Online payment system infrastructure review: Firewall			
Facility descriptions	CoA	CoB	CoC
Firewall deployed	Yes	Yes	Yes
Firewall redundancy deployed	Yes	Yes	Yes
Stateful firewall	Yes	Yes	Yes
Firewall configuration rule to allow ONLY permitted related email protocols for in/out	No	No	No
Use of best practice user name and/or password and strong password encryption (MD5)	Yes	No	No
Administration of the firewall via unsecured communications (HTTP and Telnet) permitted	Yes	Yes	Yes

Table 5. Switches review

Online payment system infrastructure review: Switches			
Facility descriptions	CoA	CoB	CoC
Standalone external gateway switch deployed	Yes	No	No
Standalone DMZ switch deployed	Yes	No	No
Appropriate VLAN used	Yes	Yes	Partly
ACL applied to block unwanted devices	Yes	No	No
Anti ARP spoofing and poison attacks enabled	No	No	No
Port broadcast-storm control enabled	No	No	No
Port security limits MAC address to a port enabled	No	No	No
Use of best practice user name and/or password and strong password encryption (MD5)	Yes	No	No
Administration of the firewall via unsecured communications (HTTP and Telnet) permitted	Yes, HTTP only	Yes	Yes

**Stage 3:** Auditing review of the online payment servers of the three selected councils (services and system identification, port scanning and vulnerability detection).

Table 6. Audit review on the council's online payment frontend web servers

Auditing review: Frontend web servers			
Facility descriptions	CoA	CoB	CoC
Best practice user name and/or password used on the operating system (OS)	No	No	No
Best practice used on OS password policy	No	No	Yes
Unnecessary TCP and UDP services ports opened	Yes	Yes	Yes
Missing patches	No	Yes	Yes
Missing service packs	No	No	Yes
Overall vulnerabilities	High	High	High

Table 7. Audit review on the council's online payment application servers

Auditing review: Application servers			
Facility descriptions	CoA	CoB	CoC
Best practice user name and/or password used on the operating system (OS)	No	No	No
Best practice used on OS password policy	No	No	Yes
Unnecessary TCP and UDP services ports opened	Yes	Yes	Yes
Missing patches	Yes	Yes	Yes
Missing service packs	Yes	Yes	Yes
Overall vulnerabilities	High	High	High

Table 8. Audit review on the council's online payment database servers

Auditing review: Database servers			
Facility descriptions	CoA	CoB	CoC
Best practice user name and/or password used on the operating system (OS)	No	No	No
Best practice used on OS password policy	No	No	No
Unnecessary TCP and UDP services ports opened	Yes	Yes	Yes
Missing patches	No	No	Yes
Missing service packs	No	Yes	Yes
Overall vulnerabilities	High	High	High

**Stage 4:** Vendor security benchmarking (MS SQL Server 2005 version 1.2.0) on online payment database application

Table 9. Vendor security benchmarking review

Vendor security benchmarking review on the councils' online payment database servers			
Facility descriptions	CoA	CoB	CoC
Operating system and network specification configuration	Partly	Partly	Partly
SQL server installation and patches	Partly	Partly	Partly
SQL server settings	Partly	Partly	Partly
Access controls	Partly	Partly	Partly
Auditing and logging	Partly	Partly	Partly
Backup and disaster recovery procedures	Partly	Partly	Partly
Replication	NA	NA	NA
Application development best practices	NA	Partly	NA
Surface area configuration tool	NA	Partly	Partly

Stage 5: Online payment system security policy review

Table 10. Online payment systems security policy review

Online payment systems security policy review on the selected councils			
Facility descriptions	CoA	CoB	CoC
Use of general online payment related (internet) usage policy	Yes	Yes	Yes
Use of information security policy (online payment related) to the councils' staff	No	No	No
Use of information security policy – technical to the councils' IT staff	No	No	No
Regular update or review the online payment related policy	No	No	No
Advise general online payment usages including security awareness to new starter	No	No	No
Frequent advice to staff for IT security information including online payment related	No	No	No

## 5 Discussion

As per recommendations to best practice, based on the results provided earlier, there were insufficient settings, configurations and implementations of the related online payment system devices in all five testing stages. These deficiencies may be a source of potential risks to all the online payment systems at the three selected WA councils.

There were six main factors uncovered in this study which caused these deficiencies in the operation of the related online payment system devices. These factors are described as follows:

- 1) Lack of IT security standards awareness or industrial best practices by the IT staff

For example, there was no standalone DMZ switch deployed at both Councils B and C. There was no alternative or redundancy internet link at all the three selected councils. There was no internet border router deployed at Council C.

- 2) Inadequate specific knowledge

The infrastructure devices of the online payment systems of all the three selected councils such as the internet border router, the DMZ switch and the firewalls had missing and inadequate configuration. One of the reasons for this shortcoming is that the IT staff at all three selected councils were not well trained in these specific technical areas. Refer to Tables 2 to 5 for more details.

- 3) Inefficient communication between the IT staff

There was evidence that there were incorrect configurations on the firewall' ACL codes related to the online payment systems of both Councils A and B. In addition, there were no change management processes at all three selected councils. This may also be attributed to inefficient communication between the IT staff.

- 4) Limited IT training for staff as a result of a limited training budget

There were limited IT training budgets assigned at all the three selected councils. Consequently, this may also have contributed to the insufficient knowledge of the staff for managing the online payment system all three selected councils.

- 5) Insufficient time for task completion by the IT staff

The IT staff at all three selected councils had multiple duties and diverse tasks. In some cases, this may have resulted in some tasks being continuously unattended (personal communications, 2009, 2010). Furthermore, at all three selected councils, there was also no time left for proper documentation as mentioned in Stage 1.

- 6) Reliance on external consultants for specific IT projects

The IT departments at all three selected councils depended on outsourcing to solve their IT expertise problem which may cause a possible disadvantage in terms of a lack of knowledge transfer to the IT staff. For example, the firewall systems at all three selected councils were deployed by external consultants. Furthermore, no proper documentation was provided by these external consultants after work completion (personal communications, 2009, 2010).

## 6 Conclusions

As a result of this study, the implementation framework, presented here may be used to audit an online payment system, in particular one which uses MS SQL 2005 as an online backend database application. Any local government councils or organizations with a similar architecture may easily adopt this framework in order to use as a guideline in auditing, testing and documenting the security of their online payment system based on their exiting ICT security policies.

The framework uncovered a range of technical and human factor issues that need amelioration by the councils concerned. By reducing the risks that these deficiencies have uncovered it should provide a safer environment for citizens to transact with the councils in the future.

## 7 References

[1] Australia Post, "Postbillpay," vol. 2011: Australia Post, n.d.

[2] CIS, "Security configuration benchmark for Microsoft SQL Server 2005 version 1.2.0 January 12th, 2010," vol. 2010: The Center for Internet Security (CIS), 2010.

- [3] Commonwealth Bank of Australia, "BPOINT," vol. 2011: Commonwealth Bank of Australia n.d.
- [4] GFI, "New version release: GFI LANguard 9.0," vol. 2010. Cary, NC GFI Software, 2010.
- [5] P. Herzog, "OSSTMM 2.2: Open-source security testing methodology manual," vol. 2008: ISECOM, 2006.
- [6] Infor, "ePathway," vol. 2011: Infor, 2009.
- [7] G. Lyon, "Nmap security scanner," vol. 2009. Palo Alto, CA: NMAP.ORG, 2009.
- [8] Microsoft Corporation, "N-Tier data applications overview," vol. 2010. USA: Microsoft Corporation, 2010.
- [9] B. Rathore, M. Brunner, M. Dilaj, O. Herrera, P. Brunati, R. K. Subramaniam, S. Raman, and U. Chavan, "Information systems security assessment framework (ISSAF) draft 0.2.1," vol. 2009. USA: Open Information Systems Security Groups (OISSG), 2006.
- [10] VeriSign Authentication Services, "FAQ: Extended validation SSL," vol. 2011: Symantec Corporation, n.d.

# Information Security Policy Concerns as Case Law Shifts toward Balance between Employer Security and Employee Privacy

Kathleen Jungck

Information Assurance and Security  
Capella University  
225 South 6th Street, Minneapolis, MN 55402 USA  
KJungck@CapellaUniversity.edu

Syed (Shawon) M. Rahman, Ph.D.

Assistant Professor, University of Hawaii-Hilo and  
Adjunct Faculty at Capella University  
200 W. Kawili St, Hilo, HI 96720 USA  
SRahman@Hawaii.edu

**Abstract**— Employee expectation of privacy within the workplace diminished as information technology use increased within the enterprise. During the same time period, increased use of information technology within the enterprise created a corresponding need for heightened information security. Technology use has continued to become increasingly ingrained within society as a whole to the point that the US Supreme Court has recognized the use of some forms of technology as part of an individual's right of self expression. Trends in case law have increasingly defined specific, limited scopes of privacy in which employee use of information technology is protected from employer scrutiny. This paper<sup>1</sup> discusses the shift toward balance between employer security needs and employee privacy rights and the impact to information security policy development and implementation.

**Keywords:** *Information Security, Case Law, Employer Security, Employee Privacy, security policy, workplace environment*

## I. INTRODUCTION

As computers and other technology have become ever more ingrained within the enterprise, expectations of privacy in the workplace have diminished significantly. Legislation and case law within the United States have upheld the right of the enterprise to set company policies and perform actions to maintain information security during the course of normal business. Increasingly, legislation and case law have also defined specific areas in which employees are guaranteed an expectation of privacy within cyberspace. The law, however, often lags behind technical and cultural change [17]. Policies vary wildly from enterprise to enterprise and challenges to the statutes have received mixed rulings in regard to both employer policies and to expectations of privacy by employees [5],[10],[12],[14],[15]. Such a rapidly evolving information security environment raises several questions regarding the balance between an employee's expectation of privacy and the enterprise's need for security including where

an Information Technology (IT) security manager should draw the ethical line in developing, maintaining, and enforcing security policy when caught between these extremes.

In section 2, an overview of the topic is presented. Enterprise security is explored in more depth in section 3, followed by considerations of employee privacy in section 4, and impacts to information security policy development and implementation are discussed in section 5.

## II. AN OVERVIEW

IT, which was once seen as merely a tactical advantage, is now a necessity on which most enterprises, large or small, depend [2]. As IT continues to penetrate further into the core of the enterprise, information security concerns also increase. Employees have become more mobile, geographically dispersed, and in many cases, the workspace itself has become virtual. Personal or employer provided personal digital assistants (PDAs) or smartphones are increasingly utilized by employees and may intersect with the company's network during private transactions. Employees may utilize personal or company provided equipment from their homes or other offsite locations in order to complete work for hire, and the increasing deployment of browser based, or cloud, applications may permit employees to access private content, such as a personal e-mail account, via company resources.

Where once IT analysts investigated complaints of employee misuse of systems related to harassing e-mails, game playing, or accessing inappropriate websites, security postures have become much more proactive. Security and network analysts today often screen network traffic on a routine basis, filter access to websites, scan company e-mail and other digital work products for viruses or data integrity, and perform security audits while being under tremendous pressure to meet compliance standards from a growing number of regulations regarding information security. Historically, both legislation such as the Computer Fraud and Abuse Act (CFAA) and case law including *TBG Ins. Corp. v. Superior Court* have recognized an employer's right to enforce security policy and monitor computer usage and traffic during the course of normal business operations [5],[17].

<sup>1</sup> This work is partially supported by EPSCoR award EPS-0903833 from the National Science Foundation to the University of Hawaii

Societal use of technology has increased to the point that in *Quon v. Ontario*, the Supreme Court considered societal norms on the use of technology as part of an individual's self expression a critical factor in determining its ruling [15]. Consumers are increasingly demanding privacy in their online transaction which has been recognized in legislation such as HIPAA which protects confidential information both for customers and employees as well as recent recommendations by the Federal Trade Commission [8]. The European Union has progressed even farther with legislation ensuring privacy protections for employees within the workplace [17], and even more substantial privacy rulings within Latin America may also have a significant impact on future policy debates and legislation [3].

### III. ENTERPRISE SECURITY

Some employees may feel that employers monitor systems and utilize information security practices in order to spy on them ala Big Brother from George Orwell's 1984, but in reality most enterprises are simply mitigating risk. Full time continuous monitoring of employees is simply not cost effective [9]. Enterprises have multiple motivators for information security including legal and regulatory compliance, fiduciary responsibility including due diligence, legal liability, protecting confidential information such as trade secrets, protecting the company's reputation, and vulnerability to both internal and external attacks [5],[6]. While news reports focus on high profile economic hackers, hactivists, and cyberterrorists, the majority of information system attacks come from within the enterprise itself from either employee error or malfeasance such as theft, vandalism, or economic espionage [6],[20].

Enterprises are required to implement security controls and practices to comply with legislation including the Health Insurance Portability and Accountability Act (HIPAA), Sarbanes-Oxley Act (SOX), Gramm-Leach-Bliley Act (GLB), and Family Educational Rights and Privacy Act (FERPA), among others, which often vary by industry. Many of these controls include policies, which are in effect "organization-specific law" [19, p. 5]. Policies are high level, compulsory management directives providing generalized requirements in the form of a written document [21]. Policies provide the framework for more specific technical standards, guidelines, and procedures that detail how the policies are to be implemented. Fiduciary responsibilities of the enterprise to their clients and shareholders, including due care, are often expressed through policy.

While some laws compel security controls, other legislation including the Computer Fraud and Abuse Act (CFAA), Stored Communications Act (SCA), Electronic Communications Privacy Act (ECPA), and the Economic Espionage Act (EEA) recognize the employer's right to enforce security practices through explicit policy, including the right to monitor communications during the usual course of business. These laws also provide employers legal recourse when

employees violate policy. In *TBG Ins. Corp. v. Superior Court*, the California Court of Appeals stated that

*"more than three quarters of this country's firms monitor, record, and review employee communications and activities on the job, including their telephone calls, e-mails, Internet connections, and computer files...the use of the computers in the employment context carries with it social norms that effectively diminish the employee's reasonable expectation of privacy with regard to his use of his employer's computers" [5].*

In *US v. Simons*, the Fourth Circuit found that the employer's internet use policy, which restricted employee use of the internet for official business and informed employees that the employer would conduct audits to ensure compliance, "defeated any expectation of privacy" [5].

Policy, however, must be clear and explicit as well as consistently enforced to remain valid. Policy must also abide by federal, state, and local laws to which the enterprise and its employees are accountable. In *Convertino v. US Dept. of Justice* the court found that if employer policy is not clear and explicit in denying an expectation of privacy it is not enforceable [5]. *Long v. US Military Court* determined that if a policy is stated, but not consistently enforced, it loses its authority [14]. And *Quon v. Ontario* ruled that changes made in policy enforcement by a supervisor, termed operational realities, may void a policy [15]. So simply stating a policy is not enough; it must be understandable by a reasonable person, explicit in its direction, and able to be enforced consistently without discrimination or it will not stand under legal challenge.

### IV. EMPLOYEE PRIVACY

As human beings, even within the employer-employee relationship, the average person in the United States has certain expectations of privacy within the workplace through societal norms. The United States constitution, while not explicitly guaranteeing privacy, does contain provisions protecting freedom of speech, protected communications such as attorney-client privilege, and protection against search and seizure without due cause. Other legislation, including HIPAA, SOX, ECPA, SCA, and the National Labor Relations Act (NLRA) define specific, limited scopes in which individuals are guaranteed an expectation of privacy.

		Social presence/ Media richness		
		Low	Medium	High
Self-presentation/ Self-disclosure	High	Blogs	Social networking sites (e.g., Facebook)	Virtual social worlds (e.g., Second Life)
	Low	Collaborative projects (e.g., Wikipedia)	Content communities (e.g., YouTube)	Virtual game worlds (e.g., World of Warcraft)

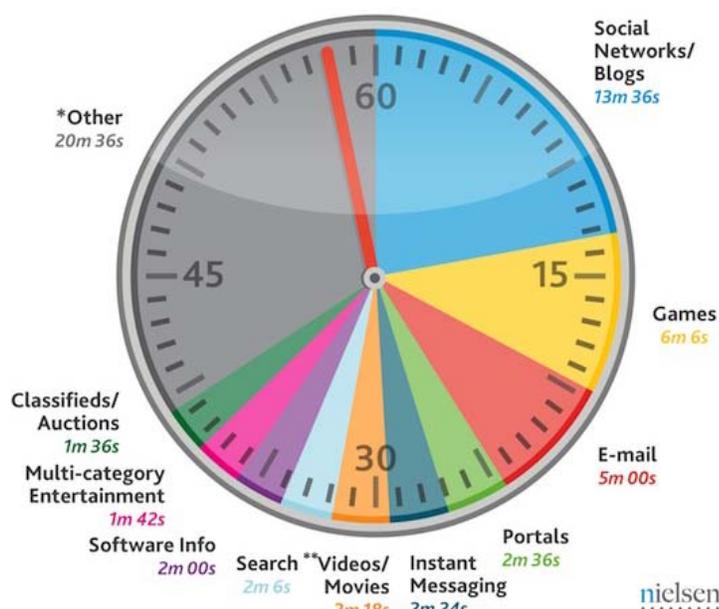
Fig. 1. US Internet Usage [19].

Case law, including *Quon v. Ontario*, *Stengart vs. Loving Care*, and *Long v. US Military Court*, has defined specific, limited boundaries in which employer policy may be not be

sufficient cause to violate an individual's expectation of privacy. In both *Convertino* and *Stengart*, privileged communications with an attorney were ruled off limits, regardless of policy [5]. In addition, the court found in *Stengart* that any reasonable person would have an expectation of privacy when utilizing a personal, password protected e-mail account and would not expect an employer to copy those communications to a cache on the hard drive, an obscure technical detail [14]. The Supreme Court in *Quon* identified seven criteria for evaluating expectations of privacy, including [14, p. 24]:

1. Existence of clear, explicit employer policy
2. Alteration of employer policy by informal practices;
3. General expectations of employers as a whole;
4. The level to which monitoring and review were to be anticipated based on the usual course of business;
5. Societal norms on the use of technology as part of an individual's self expression
6. State statutes affecting monitoring notice requirements;
7. Availability of equivalent, privately purchased devices

**If all U.S. Internet time were condensed into one hour, how much time would be spent in the most heavily used sectors?**



Source: Nielsen NetView, June 2010

\*Other refers to 74 remaining online categories visited from PC/laptops

\*\*NetView's Videos/Movies category refers to time spent on video-specific (e.g., YouTube, Bing Videos, Hulu) and movie-related websites (e.g., IMDB, MSN Movies and Netflix) only. It is not a measure of video streaming or inclusive of video streaming on non-video-specific or movie-specific websites (e.g., streamed video on sports or news sites).

Fig. 2. Types of Social Media [11, p. 62]

An important point developing in case law, as exhibited in *Quon v. Ontario*, is that while an enterprise has a right to monitor employee communications during the normal course of business, such as in operating servers and networking

equipment, they may do so only if they themselves provide the service [4]. If another party, such as an Internet service provider (ISP), wireless carrier, or hosting provider renders services on behalf of the enterprise, SCA prohibits disclosure of electronic communications without either the employee's consent or court order [4]. And in *Long v. US Military* the court specified that the right of the enterprise to audit electronic communications excludes fishing expeditions not related to the enterprise's normal course of business [14]. This trend may cause enterprises to reconsider moving some applications and operations into the cloud.

The use of the internet itself has changed with the migration toward web 2.0 and user generated content rather than the static published content seen previously. Kaplan & Haenlein define social media as "a group of Internet-based applications that allow the creation and exchange of User Generated Content" [11, p. 61]. According to the Nielsen company, more than 22% of all time spent on-line is at social media sites [16], and in many cases, the enterprise itself may utilize various types of social media in the course of normal business operations. Starbucks is perhaps an epitome of this practice, relying heavily on customer suggestions and feedback to steer their business with an entire segment of their marketing department devoted to on-line content management and delivery in partnership with companies such as SalesForce, Apple, and Yahoo.

While employee use of social media may pose potential for the enterprise including disclosure of confidential information, engaging in criminal conduct, harassing co-workers, and disparaging their employer, co-workers, or clients, employers do not have the right to simply ban social media use by employees. Employees have free speech rights in regard to their use of social media. The NLRA also affords employees, even those not unionized, the right to discuss the terms of their employment and to even criticize their employers; federal and state whistleblower statutes may protect employees who complain about conditions affecting health, safety, or financial misconduct; and federal and state constitutions protect freedom of speech, including political speech [1]. In February, 2011, American Medical Response (AMR) settled a case with the National Labor Relations board over the termination of an employee for criticizing her supervisor on Facebook [13],[18]. AMR agreed to revise their social media policy as a result.

The influence of statutes within the European Union and other global partners toward privacy must also be considered due to the increasingly global nature of today's enterprise and the rapid cultural exchange available over the internet. Treaties and other global polices may influence the development of more privacy centric legislation within the US to mirror best practices and global societal norms that place an increasing premium on personal privacy [3].

## V. SECURITY POLICY IMPLICATIONS

Evolving case law, changing societal expectations, the explosion of personal technology use, and the increased penetration of cloud based services for both personal and enterprise use affect the balance between the enterprise's need for security and the employee's expectation of privacy. Scenarios such as inadvertent discovery of privileged information, implications of cloud based or outsourced services, and the application of invasive or passive audits must be considered when developing security policy.

### A. Confidentiality of Privileged Information

Legislation and regulation, including the Americans with Disabilities Act (ADA) and HIPAA, as well as evolving case law, have directed that some types of personal employee information is privileged and must be kept confidential. In the normal course of business, security analysts may inadvertently encounter privileged information. Both training and policy are required to mandate that privileged information be kept confidential unless obligations of public health or safety mandate disclosure.

### B. Changing Nature of the Workplace

As the nature of the workplace changes an increasing number of employees are utilizing technology equipment provided by the employer when operating off company property, and, in some instances, from within their own homes. *Quon v. Ontario* found that even if equipment is provided by the employer, employee expectations of privacy may not be entirely excluded in light of operational realities. Policy could maintain that company property operating outside the employer's place of business, such as a laptop, would be subject to audit of electronic communications effected on behalf of the employer in the normal course of business but would exclude personal employee accounts outside the employer's purview, even if those communications were cached to the device.

Employees are also increasingly utilizing their own equipment, such as PDAs and smartphones, to both enact business on behalf of the enterprise as well as for personal use. Many of these employee owned devices are both web capable, equipped with cameras, capable of copying digital information through bluetooth, USB, or other communications methods, and may interact with the enterprise infrastructure. These devices may expose the enterprise to risk from viruses, employee theft of intellectual property, or industrial espionage, yet are outside of enterprise control.

The increasing penetration of cloud based services may impact security policy within the enterprise. Departments may bypass IT and contract for services on their own and employees themselves may choose to utilize cloud based services for either personal or professional use. Under such circumstances, policy may not permit the blanket proactive screening of e-mails, phone calls, instant messaging, text messaging, web conferencing, or other personal digital work products regarding possible ethical or legal violations for which the employer could be liable or for industrial espionage. If the enterprise supplies the equipment or infrastructure on

which communications operate, they may monitor communications in the normal course of business. However, *Quon v. Ontario* and *SCA* established that if the service is provided by a third party such as an ISP, communications may not be released without either the employee's consent or a court order [4]. And if the device is employee owned, the enterprise may have no control at all.

### C. Security Policy Impact

Prevention is considerably less expensive than recovering from a breach and both passive monitoring and periodic general audits are definitely covered under the course of normal business operations. Ethically, however, truly invasive forensics of employee communications without sufficient cause could damage the employer-employee trust relationship and expose the enterprise to a discrimination suit, particularly under the changing climate demonstrated by recent additions to case law. Consideration must also be made when choosing a service provider as more applications move to the cloud or are outsourced in light of *Quon v. Ontario* and other *SCA* related cases.

The current consensus in case law is that explicit, understandable, documented, consistently enforced policy is the cornerstone of an enterprise's security program. In order to remain valid, the policy must be consistently enforced and not weakened by supervisory revisions while in operation. Policies will also require frequent maintenance to remain valid. Employees have both societal expectations of limited privacy as well as explicit rights to privacy recognized in law which must be balanced with the enterprise's need for security. Policies may restrict the use of the internet and other electronic communications for business use while on the job, but cannot regulate employee use off the job; at most, policies can require standards such as professional conduct while employees are off the job. Stating that the employee is a representative of the enterprise is not sufficient justification by itself for intrusion into their private life.

When developing policy, ethics must also be considered beyond the boundaries of current legislation and case law as evolving technology frequently outpaces legislation [17]. Enterprises and their employees share a bond of trust; employees are not only agents of the enterprise, but also a valuable asset. As with information assets, employees must be protected, including reasonable provisions for personal privacy. Policies need to balance the needs of both the enterprise and the employee, including those of the information security analysts that implement and enforce that policy.

## VI. CONCLUSION

For an industry as a whole, Information Security managers and analysts need to recognize the shift in current case law. For some time, the trend by the courts was to favor the enterprise in cases where policy was explicit, understandable, and enforceable, essentially supporting that employee expectations of privacy within in the workplace were slim. However, recent decisions, particularly the Supreme Court's ruling on *Quon v. Ontario*, have set new standards for determining expectations of privacy as technology becomes increasingly ingrained within the workplace. Recent decisions

have shifted the pendulum more towards a balance between employer security and employee privacy. Policies need to be updated to reflect the new operational realities, and protections undertaken for security analysts enforcing and implementing those policies who may encounter situations in which they will need to decide what remains private when personal employee information is encountered during security audits

#### ACKNOWLEDGMENT

The authors would like to thank Dr. Bary Pollack for his review & suggestions and Tim and Corrine from the Starbucks US marketing team for sharing their experiences with social media use in a real world environment.

#### REFERENCES

- [1] Anthony, W. J. & Bovee, T. A. (2010, October 25). Employment and Immigration Law: Be Careful When Reining in Social Media. [Online]. Available: <http://www.ctlawtribune.com/getarticle.aspx?ID=38658>
- [2] Carr, N. G. *Does IT Matter? Information Technology and the Corrosion of Competitive Advantage*. Harvard Business Press. ISBN: 9781591394440, 2004.
- [3] Cerda, A., Laurant, C., Blum, R. O. (2010, April 21). Recent Privacy and Data Protection Developments in Latin America and Their Impact on North American and European Multinational companies – IAPP Global Privacy Summit. [Online]. Available: <http://www.slideshare.net/cedriclaurant/quotrecent-privacy-and-data-protection-developments-in-latin-america-and-their-impact-on-north-american-and-european-multinational-companiesquot>
- [4] Covington & Burling. (2010, June 18). Supreme Court Avoids Key "Expectation of Privacy" Question for Text Messages. [Online]. Available: <http://www.cov.com/files/publication/4a190e59-7f5d-49db-b9e2-78da18bb2ed7/presentation/publicationattachment/b5c098a8-2335-46af-b941-7f142f117b83/supreme%20court%20avoids%20key%20%27expectation%20of%20privacy%27%20question%20for%20text%20messages.pdf>
- [5] Cybertelecom Federal Internet Law & Policy. (nd). Expectation of Privacy. [Online]. Available: <http://www.cybertelecom.org/security/privacy.htm>
- [6] Dulaney, E. *Comptia Security+ Study Guide* (4th Ed.). Indianapolis, IN: Wiley Publishing, Inc. ISBN: 9780470372975, 2009.
- [7] Diver, S. (2006, July 12). Information Security Policy - A Development Guide for Large and Small companies. [Online]. Available: [http://www.sans.org/reading\\_room/whitepapers/policyissues/information-security-policy-development-guide-large-small-companies\\_1331](http://www.sans.org/reading_room/whitepapers/policyissues/information-security-policy-development-guide-large-small-companies_1331)
- [8] Federal Trade Commission (FTC). (2010, December). Protecting Consumer Privacy in an Era of Rapid Change: A Proposed Framework for Businesses and Policymakers. [Online]. Available: <http://www.ftc.gov/os/2010/12/101201privacyreport.pdf>
- [9] Franklin, C. "How Much Monitoring is Enough?", *Infoworld*, vol. 25, issue 19, p. 29. ISSN 0199-6649, May 12, 2003.
- [10] Gantz, S. (2010, April 4). Employee Expectations of Privacy in the Workplace Only Improving in Very Specific Contexts. [Online]. Available: <http://blog.securityarchitecture.com/2010/04/employee-expectations-of-privacy-in.html>
- [11] Kaplan, A. M. & Haenlein, M. "Users of the world, unite! The challenges and opportunities of social media," *Business Horizons*, vol. 53, issue 1, pp. 59–68, January-February 2010.
- [12] Mayer Brown (2009, December 21). District Court Finds Employee Had Reasonable Expectation of Privacy in Personal Communications From Workplace. [Online]. Available: <http://www.mayerbrown.com/publications/article.asp?id=8329&nid=6>
- [13] Musili, S. (2011, Feb. 7). Company settles Facebook firing case. [Online]. Cnet News. Available: [http://news.cnet.com/8301-1023\\_3-20030955-93.html](http://news.cnet.com/8301-1023_3-20030955-93.html)
- [14] Rasch, M. (2006 October 31). Employee privacy, employer policy. [Online]. Available: <http://www.securityfocus.com/columnists/421>
- [15] Schwartz, J., Chadwick, J., & Lucas, A. "U.S. Supreme Court Decision in Quon Changes Workplace Privacy Law". *Insights: The Corporate & Securities Law Advisor*, vol. 24, issue 8, pp. 23-26, August 2010.
- [16] (2010, June 15). Social Networks/Blogs Now Account for One in Every Four and a Half Minutes Online. [Online]. Available: <http://blog.nielsen.com/nielsenwire/global/social-media-accounts-for-22-percent-of-time-online/>
- [17] (2011, February 18). U.S. Supreme Court Decision and NLRB Settlement Expand Employer Exposure to Liability for Interference with Protected Activity. [On-line]. Lane Powell. Available: <http://www.lanepowell.com/13668/u-s-supreme-court-decision-and-nlr-settlement-expand-employer-exposure-to-liability-for-interference-with-protected-activity/>
- [18] Tavani, H. T. *Ethics & Technology: Controversies, Questions, and Strategies for Ethical Computing* (3rd ed.). Hoboken, NJ: Wiley. ISBN: 9780470509500, 2010.
- [19] (2010, August). What Americans Do Online: Social Media and Games Dominate Activity. [Online]. Available: [http://blog.nielsen.com/nielsenwire/online\\_mobile/what-americans-do-online-social-media-and-games-dominate-activity/](http://blog.nielsen.com/nielsenwire/online_mobile/what-americans-do-online-social-media-and-games-dominate-activity/)
- [20] Whitman, M. E., & Mattord, H. J. *Management of Information Security* (3rd ed.). Boston: Cengage Learning. ISBN: 9781435488847, 2010.
- [21] Wood, C. C. & Lineman, D. (2009). Chapter 2: Instructions in *Information Security Policies Made Easy Version 11* (Version 11 ed.). Information Shield, Inc. [Online]. Available: [http://searchsecurity.techtarget.com/searchSecurity/downloads/Wood\\_I\\_SPM\\_Ch2.pdf](http://searchsecurity.techtarget.com/searchSecurity/downloads/Wood_I_SPM_Ch2.pdf)

# PPSAM: Proactive PowerShell Anti-Malware

## Customizable Comprehensive Tool to Supplement Commercial AVs

Alejandro Villegas and Lei Chen

Department of Computer Science, Sam Houston State University, Huntsville, TX 77341, USA

**Abstract** - *This research first explores the different types of Anti-Malware solution approaches, evaluating the pros and cons, and concentrating on their potential weaknesses and drawbacks. The malware technologies analyzed include Windows Direct Kernel Object Manipulation (DKOM), Kernel Patch Protection, Data Execution Prevention, Address Space Layout Randomization, Driver Signing, Windows Service Hardening, Ghostbuster, Assembly Reverse Analysis, and Virtual CloudAV. Furthermore, a proactive comprehensive solution is provided by utilizing the Windows PowerShell 2.0 utility that is available for Windows Vista, 7, 2008 and 2008 R2. The proposed Proactive PowerShell Anti-Malware (PPSAM) is a utility that monitors the system via health checks with shell scripts that can be fully customized and have the ability to be executed on remote systems. PPSAM is designed to be a proactive complement that attempts to promote early discovery of intrusions and malicious applications, and to provide triggers and reports utilizing the scripts' output.*

**Keywords:** PowerShell, malware, anti-virus, proactive, customizable, security

## 1 Introduction

Majority of end users already have a preferred Anti-Virus (AV) solution such as Windows Defender, McAfee or Norton. In the meanwhile, anti-rootkits like VICE, GMER, and Rootkit Unhooker [8] have become fairly popular. Most of the anti-malware products take a reactive, rather than proactive, approach to detection. The first and most common strategy is to compare applications with common malware signatures stored in a database and flag them as suspicious malware, whether a virus, Trojan or rootkit. Malware will continue to evolve and make AV applications obsolete is simply the nature of the game.

Some computing essentials remain the same for all different types of malware: for instance, the best malware is the one that goes undetected (at least until is found or discovered either manually or by a new AV tool). Unfortunately, when a virus becomes too noticeable or by the time is discovered it probably has already caused a lot of

damage. Average end users could have a rootkit installed and don't even realize so, as they rely on the anti-virus to detect the compromise, or expects a call from the system or network administrator reporting "unusual" activities. In addition, hackers are well aware of major AV companies too, and they are used to code malware that detects AVs and formulate a scripting workaround to avoid detection. If the AV or Anti-Malware solution is not even able to detect the illicit computing transaction, it becomes useless.

The rest of the paper is structured as follows. Next in Section 2, we survey the existing vulnerabilities that can be used by malware to hide themselves in the system. Section 3 discusses the motivation of this research, followed by the technical details of PPSAM in Section 4 including four levels of script repository, differences from existing solutions, and hardware and software requirements. Section 5 shows a few examples how PPSAM can work along with the AVs to help monitor the system. We draw conclusion in Section 6 and propose future work in Section 7.

## 2 Background

The research performed by Woei-Jiunn Tsaur [8] clearly exposes five potential vulnerabilities that rootkit developers can exploit to maintain their applications undetected.

### 2.1 Windows DKOM

A lot of the current research focuses on kernel data schemes that aim to detect hooking driven virtual machine rootkits. Nevertheless, DKOM has been proven to be strategy inefficient [8]. Woei-Jiunn Tsaur et al. in [8] go beyond analyzing the traditional rootkits that typically are traceable within registry keys, questionable drivers or malicious API injections [4]. DKOM style rootkits exploit the kernel object implementation in Windows systems by altering EPROCESS objects [8]. The DKOM detection approach by Woei-Jiunn Tsaur et al. is to install a hidden driver as an object in order to detect DKOM activities. The proposed rootkit named hookzw.sys is a driver format (composed using Borland TASM 5.0) which executes on the Windows XP SP2 platform. One of the DKOM rootkit

techniques is to exploit 'PsLoadedModuleList' in order to hide processes. The core data structures that are modified by a DKOM rootkit are: List\_Entry data structures of Object Directory, Object Driver, Object Device and PsLoadedModuleList [8]. Woei-Jiunn Tsauro et al. outline the following five tips to detect DKOM rootkits: Removing Object Drivers and Object Devices from Object Dir, Removing Object Drivers from Driver Object\_Type, Removing Object Devices from Device Object\_Type, Removing Drivers from PsLoadedModuleList, and Altering Object Driver Appearance.

There is no doubt that DKOM rootkits are a threat, and a comprehensive tool for its prevention needs to be developed by a commercial Research and Development (R&D) entity. Furthermore, this approach is designed to discover unknown rootkits on the wild and does not necessarily provide a mechanism to create a signature based database for further detection.

## 2.2 KPP, DEP, ASLR, DS, and WSH

There is a plethora of different third party rootkit detectors for the Windows platforms such as VICE, GMER, Rootkit Unhooker, among others [8]. Nonetheless, Microsoft introduced five different software utilities to fight rootkits and malware in general for Windows Vista and newer versions [1]. These five utilities are briefly introduced as follows [1]:

- Kernel Patch Protection (KPP) – provides protection of the Windows kernel (formerly known as PatchGuard) at the System Service Descriptor Table (SSDT) level; prevents malware from hooking into system APIs.
- Date Execution Prevention (DEP) – deals with Buffer Overflow prevention.
- Address Space Layout Randomization (ASLR) – randomly arranges the positions of key data areas in a process's address space.
- Driver Signing (DS) – signs legit drivers in order to prevent the installation of new malicious drivers.
- Windows Service Hardening (WSH) – increases restriction to Windows background process.

Albeit the mentioned solutions have made the Windows operating systems more secure and stable, there are exploits such as DKOM among others that still represent a threat to the Windows kernel, not to mention that malware keeps evolving, and Windows only utilizes two of the four Intel's architecture layers leaving the OS still vulnerable [1].

## 2.3 Strider Ghostbuster

The Strider Ghostbuster [9] is a cross-view difference based approach Ghostware detector. The technique utilized

compares a high-level infected scan with a low-level clean scan. Furthermore, it runs an inside-the-box versus an outside-the-box clean scan [9].

The Strider Ghostbuster approach is essentially a differentiation of potentially infected systems with a known clean one. The term "Ghostware" refers to programs such as rootkits and Trojans with stealth hiding capabilities [9]. The main areas where Ghostbuster runs its diff approach is in the "Master File Table", the "Raw Hive Files", and the "Kernel Process List" [9]. Common Ghostware detected by Ghostbuster includes Urbin, Mersting, Vanquish, Aphex, etc. [9].

While Ghostbuster is considered a good approach to determine whether the files/registry values/drivers were modified or altered, there are some rootkits like DKOM that may bypass the diff check. In addition, it utilizes a lot of system resources when handling the system components comparison. Ghostbuster also has the capability of providing a Virtual Machine (VM) in order to run diff check scans on virtual environments [9]. It may be a good solution with questionable scalability when it comes to large IT environments.

## 2.4 Assembly Reverse Analysis

The Assembly Reverse Analysis is essentially backward engineering the rootkit assembly code, utilizing tools such as MASM, ASM, and TASM [10]. The Windows debugger Ollydbg is also useful to analyze the content of a given malware [10]. This approach is for expert computer users who are able to decompile the malware/rootkits, analyze the contents and restore the system to its original stage as applicable. It is a very hands-on strategy and does not provide the users with a sustainable solution, which varies on a case by case basis and is not recommended for the average end users.

## 2.5 Virtual CloudAV

Cloud computing has revolutionized the deployment of hardware infrastructure. There are different cloud solutions such as Amazon's Web Services and Google's AppEngine. The concept of Infrastructure as a Service (IaaS) refers to offering access to remote computer resources [5]. The architecture of this virtual cloud solution consists of running a Kernel Agent that gathers information from each virtual machine and passes the data into a ProxyScan that analyzes the data and seeks for potential malware or kernel rootkits [5]. It is considered an excellent solution for cloud providers as they typically grant root/administrator access to their VM clients, yet remain liable for their actions as they own the hardware infrastructure. Implementation with similar functionality can be costly for small and medium businesses (SMBs).

There is also a behavior based approach in [4] that hooks Native APIs in the kernel mode, however it does have an impact on system performance in addition to the standard AVs system utilization.

### 3 Motivation

In this research, we decided to develop a boutique style anti-malware tool that would address the key drawbacks of each of the algorithms researched, yet comprehensive, scalable and intuitive. The goal of this research was to target newer Windows operating systems such as Vista and 7. The proposed solution PPSAM takes advantage of the Windows PowerShell supplied on newer Windows Operating Systems to craft simple scripts that monitor suspicious activity on a given system.

PowerShell (PS) has the capability of running commands on remote systems, therefore is ideal for scalability purposes. In addition PowerShell can be utilized to create scripts and functions by combining different cmdlets [7]. PPSAM is a starting point solution that can be customized with additional PS scripts and actualize them as malware evolves. The main purpose is to keep this solution relatively obscure in order to prevent automatic detection by malware components. The script can be located in random locations, named differently, and executed on different manners. Ideally it can be scheduled to run periodically (without user intervention), create reports and flag them accordingly with triggers designed to catch suspicious activities within the system(s). The suggested method to run PPSAM is to execute it from a remote system to avoid affecting performance and ensure that the source system is "clean".

Before the discussion of the implementation and use of PPSAM, here we list the key drawbacks of each of the algorithms researched:

1. DKOM: potential security holes not currently exploited. Solution algorithm requires corporate level R&D investment.
2. Microsoft Security Implementations: Efficient. However the Windows platform only utilizes two of the four Intel's architecture rings. OS still exposed to other potential malware/exploits.
3. Strider Ghostbuster: excellent diff approach; requires a lot of resources and limits scalability.
4. Assembly Reverse Analysis: ideal for a malware research lab, not practical for average end users.
5. Virtual CloudAV: comprehensive anti-malware setup for virtual cloud environments. It requires hardware investment and extensive configuration, optimal for ISPs not a promising solution for SMBs.

## 4 PPSAM

The Windows PowerShell [6] is native to the Operating System and therefore is able to interact with the OS and Microsoft applications flawlessly. PPSAM is designed to operate from the command line via the default PS cmdlets.

### 4.1 Levels of script repository

The script repository is divided in four levels: Registry, Network, Driver, and Application. These four levels and the corresponding Cmdlets [6] are introduced as follows.

- **Registry Level**

PowerShell cmdlets that interact with the registry:

- **Get-Item** – get a file/registry object (or any other namespace object)
- **Get-ChildItem** – get child items (contents of a folder or reg key)
- **Get-Acl** – get permission settings for a file or registry key
- **Get a registry key** – PS `C:\>get-item hklm:\software\microsoft\exchange`

- **Network Level**

PowerShell cmdlets that retrieve mac-addresses:

- `$strComputer = "."`
- `$colItems = get-wmiobject -class "Win32_NetworkAdapterConfiguration" ` -computername $strComputer | Where{$_.IpEnabled -Match "True"} foreach ($objItem in $colItems) { write-host "Hardware Address:" $objItem.MACAddress}`

- **Driver Level**

PowerShell cmdlets that retrieve drivers:

- **Get-WmiObject -Class Win32\_SystemDriver | Format-List Name, Caption, Description, InstallDate, PathName, Started, StartName, Status, SystemName**

- **Application Level**

PowerShell cmdlets that verify exchange health:

- **Test-ServiceHealth** – tests if all required services have started successfully
- **Test-SystemHealth** – gathers data about Microsoft Exchange system and analyzes the data according to best practices
- **Test-UMConnectivity** – tests the operation of a computer that has the Unified Messaging (UM) server role installed
- **Test-WebServicesConnectivity** – tests the functionality of Exchange Web Services

PPSAM takes advantage of the comprehensive utility availability of PowerShell in order to run health checks at different levels and monitor the system for suspicious

activity. PPSAM utilizes a collection of PowerShell scripts fully customizable that can be easily executed and formulate reports in HTML and/or XML format for easy viewing.

### 4.2 Differences from existing solutions

PPSAM is a simple utility that is instrumental to perform proactive malware scans and flag suspicious activity that could have been overlooked by a commercial AV. PPSAM can be customized and deployed to several systems. While it does not replace traditional AVs or Anti-Rootkit programs, it is designed to complement them. PPSAM PowerShell architecture makes the scripts executable natively without third party application installation requirements. In addition, the code is transparent and there is no need to install malware freeware that might be facilitated from web sites that include malware along with the utility. It is a middle man solution – it might not be as thorough as debugging with Ollydbg or MASM [10], but has enough capabilities to detect malicious activity in a system(s) that could have bypassed a traditional AV.

### 4.3 Software and hardware requirements

#### Software Requirements:

- Windows Vista, 7, 2008 or 2008 R2
- Windows Framework Management (Installed by default in Windows 2008 R2 and Windows 7)
- PowerShell 2.0
- WinRM 2.0
- BITS 4.0

#### Hardware Requirements:

- 1 gigahertz (GHz) or faster 32-bit (x86) or 64-bit (x64) processor
- 1 gigabyte (GB) RAM (32-bit) or 2 GB RAM (64-bit)
- 16 GB available hard disk space (32-bit) or 20 GB (64-bit)
- DirectX 9 graphics device with WDDM 1.0 or higher driver

## 5 Performance and analysis

This section shows a few examples of how PowerShell cmdlets can be utilized to help AVs monitor the system. The PowerShell has a myriad of different cmdlets that make interacting with a Windows platform smoothly and flawlessly. The get-itemproperty command is useful to obtain the values of a specific registry key as shown in Figure 1. This command can be combined with the invoke-command in order to be executed on a remote system. The PowerShell uses a similar approach than the Linux BASH shell, where the scripts can be coded on a simple text editor such as notepad, saved with the .ps1 file extension and can then be executed accordingly. They can also be loaded via a .bat (batch file). The registry key HKLM:\SOFTWARE\

CLASSES\CLSID\{2781761E-28E1-4109-99FE-B9D127C57AFE} queried in Figure 1 shows that the system is running the Microsoft Malware Protection IOfficeAntiVirus Implementation.

The registry key in Figure 2 shows that Windows Defender is installed as the default AV solution HKLM:\SOFTWARE\CLASSES\CLSID\{2781761E-28E0-4109-99FE-B9D127C57AFE}.

Figure 3 shows the output from the PowerShell get-process \* | more command, essentially showing all the running processes on the system. It can also be combined with the invoke-command to be executed remotely.

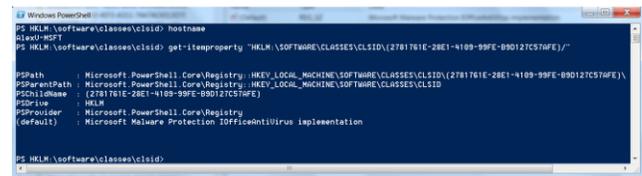


Figure 1. get-itemproperty cmdlet showing Microsoft Malware Protection IOfficeAntiVirus Implementation is running

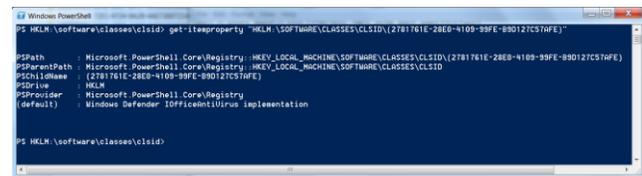


Figure 2. get-itemproperty cmdlet showing Windows Defender IOfficeAntiVirus Implementation is running

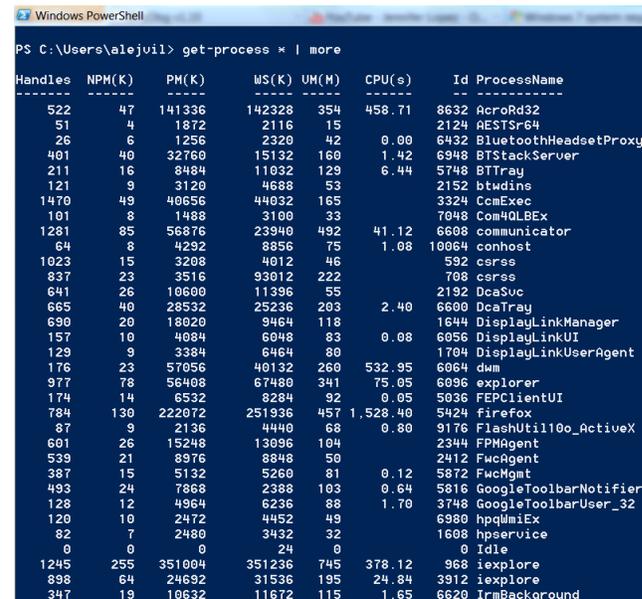


Figure 3. get-process cmdlet showing all the running processes on the system (partially shown due to length limit)

In order to parse the piped content of the PS scripts, the Out-File cmdlet can be utilized: `Get-Process | Out-File c:\output\processes.txt`.

PPSAM has a PERL written scripting implementation that controls the execution of the PowerShell commands, parses the data in HTML format and launches a browser window to display the output.

PPSAM requires a simple setup, a folder that contains the ppsam.pl PERL script, PowerShell .ps1 files containing commands to be executed, and the ppsam.html output report as shown in Figure 4.

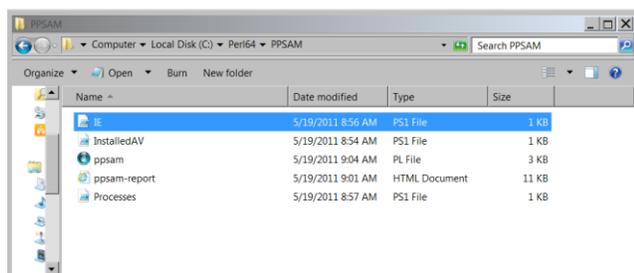


Figure 4. PPSAM file structure

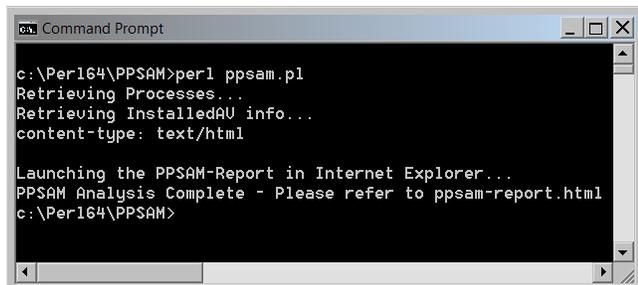
The following is a piece of sample PERL scripting code for PPSAM:

```
#!/usr/local/ActivePerl-5.6/bin/perl -w
#
# PPSAM: Proactive PowerShell Anti-Malware
# Customizable Comprehensive Tool to Supplement
# Commercial AVs
# Analyzes Windows system processes via PowerShell
# commands
#
# ppsam.pl
#
# By: Alejandro Villegas
# Department of Computer Science
# Sam Houston State University
#
# Professor: Dr. Lei Chen
# April 29, 2011
#
#Loading necessary Perl Modules
use strict;
use warnings;
use FindBin ();
use File::Copy qw(copy);
use Fcntl;
#Path to the PowerShell Executable
my $PWSpath =
"C:/Windows/System32/windowspowershell/v1.0/power
shell.exe";
#Variable to store Running-Processes
```

```
my $PS1RunningProcesses =
"C:/Perl64/PPSAM/Processes.ps1";
#Variable to store installed Antivirus Info
my $PS1InstalledAV =
"C:/Perl64/PPSAM/InstalledAV.ps1";
#Retrieve Processes via a PowerShell cmdlet: get-
process * | format-table
chomp(my @RunningProcesses = `$PWSpath -command
$PS1RunningProcesses`);
print "Retrieving Processes... \n";
#Retrieve Processes via a PowerShell cmdlet: Get-
ItemProperty "HKLM:\SOFTWARE\CLASSES\CLSID
\{2781761E-28E0-4109-99FE-B9D127C57AFE}" |
format-list | format-table
chomp(my @InstalledAVs = `$PWSpath -command
$PS1InstalledAV`);
print "Retrieving InstalledAV info... \n";
#Path to the PowerShell cmdlet to load Internet Explorer
my $IEpath = "C:/Perl64/PPSAM/IE.ps1";
#Deleting previous PPSAM-Reports
my $file = "C:/Perl64/PPSAM/ppsam-report.html";
unlink($file);
#Creating PPSAM-Report in HTML
print "content-type: text/html \n\n";
sysopen (HTML, 'ppsam-report.html',
O_RDWR|O_EXCL|O_CREAT);
printf HTML "<html>\n";
printf HTML "<head>\n";
printf HTML "<title>PPSAM: Proactive PowerShell
Anti-Malware</title>";
printf HTML "</head>\n";
printf HTML "<body bgcolor='Silver'>\n";
printf HTML "<b><h2><p align='center'>PPSAM:
Proactive PowerShell Anti-Malware</h2></b>";
printf HTML "By: Alejandro Villegas</p>";
printf HTML "<b>Processes<br></b>";
foreach (@RunningProcesses) {
printf HTML "$_ \n<br>";
}
printf HTML "<b>Anti-Malware Info<br></b>";
foreach (@InstalledAVs) {
printf HTML "$_ \n<br>";
}
printf HTML "</body>\n";
printf HTML "</html>\n";
close (HTML);
print "Launching the PPSAM-Report in Internet
Explorer... \n";
#PowerShell cmdlet to load Internet Explorer: $ie = new-
object -com
"InternetExplorer.Application"; $ie.visible = $true;
$ie.navigate("file:///C:/Perl64/PPSAM/ppsam-
report.html")
`$PWSpath -command $IEpath`;
print "PPSAM Analysis Complete - Please refer to
ppsam-report.html"
```

The PERL script ppsam.pl can be executed from the Windows command prompt as shown in Figure 5.

Lastly, the ppsam.pl script will automatically load the ppsam.html report for viewing and analysis as shown in Figure 6.



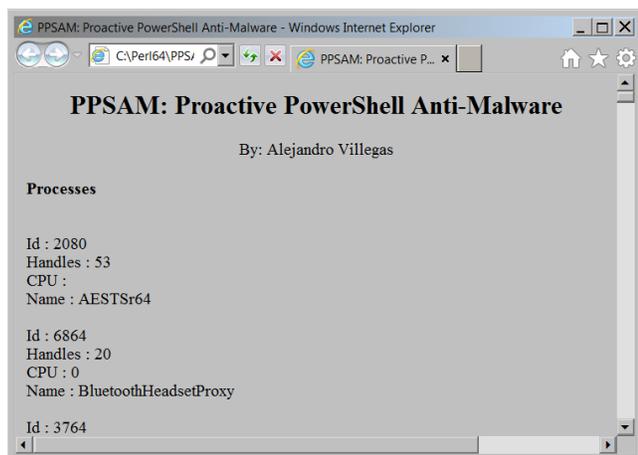
```

c:\Per164\PPSAM>perl ppsam.pl
Retrieving Processes...
Retrieving InstalledAU info...
content-type: text/html

Launching the PPSAM-Report in Internet Explorer...
PPSAM Analysis Complete - Please refer to ppsam-report.html
c:\Per164\PPSAM>

```

Figure 5. Execution of ppsam.pl via the command prompt



```

PPSAM: Proactive PowerShell Anti-Malware
By: Alejandro Villegas

Processes

Id : 2080
Handles : 53
CPU : 0
Name : AESTSr64

Id : 6864
Handles : 20
CPU : 0
Name : BluetoothHeadsetProxy

Id : 3764

```

Figure 6. ppsam.html output generated report

The Windows PowerShell is based on the .NET framework and is able to execute any functionality that is available via the traditional GUI. Therefore the complexity of each cmdlet can be customized in order to gather any necessary data to from the system to discover a potential malware proactively.

## 6 Conclusion

While commercial AV and Anti-Malware solutions overall are the first layer of protection against popular viruses, Trojans and rootkits, they typically target malware that is known and utilize a comprehensive signature database. Therefore, a more proactive approach is needed in order to promote early prevention of malware attacks, since in majority of the cases rootkits can be installed without the end users ever acknowledging the systems have been compromised. If the malware has bypassed their AV or altered system binaries, chances are it will take a while before they even discover the compromise. PPSAM provides a solution that utilizes the newly released

Windows PowerShell 2.0 which comes with the Operating System and is capable of executing a plethora of different cmdlets that can check the performance of a system including remote operations for scalability. While not an AV replacement, it is absolutely a tool that can be used in conjunction with most third party software in order to propose a more proactive approach to prevent malware adverse attacks. PPSAM can be adapted to match the requirements of every particular infrastructure. PowerShell is based on .NET, therefore there is also the option to develop new cmdlets based on clients' OS and application security priorities.

## 7 Future work

In order to provide a more user friendly application, PPSAM will possess a GUI interface coded in PERL and CGI. Such interface will have the capability of displaying, parsing and organizing the cmdlets output. In addition, archiving PPSAM scans will be an option. Furthermore, the utility will offer the feature of loading additional PowerShell scripts, as well as running a syntax and sanity check before uploading. Another proposed implementation is to be able to load PPSAM via a bootable image such as Windows PE; this alternative would be able to be utilized even on compromised systems. Additionally, the Windows Research Kernel (WRK) [2] will be used in order to create more monitoring PowerShell cmdlets at the kernel level, in order to construct a potential DKOM detection antidote. After all, new generation rootkits aim to exploit kernel memory vulnerabilities, hence the importance of kernel memory protection [3].

## 8 References

- [1] Desmond Lobo, Paul Watters, Xin-Wen Wu, Li Sun, "Windows Rootkits: Attacks and Countermeasures," etc, pp.69-78, 2010 Second Cybercrime and Trustworthy Computing Workshop, 2010.
- [2] Diomidis Spinellis, "A tale of four kernels," icse, pp.381-390, Proceedings of the 30th International Conference on Software Engineering (ICSE '08), 2008.
- [3] Dong Hwi Lee, Jae Myung Kim, Kyong-Ho Choi, Kuinam J. Kim, "The Study of Response Model & Mechanism Against Windows Kernel Compromises," ichit, pp.600-608, 2008 International Conference on Convergence and Hybrid Information Technology, 2008.
- [4] Hung-Min Sun, Hsun Wang, King-Hang Wang, Chien-Ming Chen, "A Native APIs Protection Mechanism in the Kernel Mode against Malicious Code," IEEE Transactions on Computers, 10 Feb. 2011. IEEE computer Society Digital Library. IEEE Computer Society.

- [5] Matthias Schmidt, Lars Baumgartner, Pablo Graubner, David Bock, Bernd Freisleben, "Malware Detection and Kernel Rootkit Prevention in Cloud Computing Environments," Parallel, Distributed, and Network-Based Processing, Euromicro Conference on, pp. 603-610, 2011 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2011.
- [6] Microsoft. Scripting with Windows PowerShell, 2008. <http://technet.microsoft.com/en-us/scriptcenter/dd742419.aspx>
- [7] Nicolas Bruno, Surajit Chaudhuri, "Interactive physical design tuning," icde, pp.1161-1164, 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), 2010.
- [8] Woei-Jiunn Tsaur, Yuh-Chen Chen, "Exploring Rootkit Detectors' Vulnerabilities Using a New Windows Hidden Driver Based Rootkit," socialcom, pp.842-848, 2010 IEEE Second International Conference on Social Computing, 2010.
- [9] Yi-Min Wang, Doug Beck, Binh Vo, Roussi Roussev, Chad Verbowski, "Detecting Stealth Software with Strider GhostBuster," Dependable Systems and Networks, International Conference on, pp. 368-377, 2005 International Conference on Dependable Systems and Networks (DSN'05), 2005.
- [10] Yong Wang, Dawu Gu, Jianping Xu, Fenyu Zen, "Assembly Reverse Analysis on Malicious Code of Web Rootkit Trojan," Web Information Systems and Mining, International Conference on, pp. 501-504, 2009, International Conference on Web Information Systems and Mining, 2009.

# Modeling Learningless Vulnerability Discovery using a Folded Distribution

Awad A. Younis<sup>1</sup>, HyunChul Joh<sup>1</sup>, and Yashwant K. Malaiya<sup>1</sup>

<sup>1</sup>Computer Science Department, Colorado State University, Fort Collins, CO 80523, USA

**Abstract** – A vulnerability discovery model describes the vulnerability discovery rate in a software system, and predicts the future behavior. It can allow the IT managers and developers to allocate their resources optimally by timely development and application of patches. Such models also allow the end-users to assess security risk in their systems. Recently, researchers have proposed a few vulnerability discovery models. The models are based on different assumptions, and thus differ in their accuracy and prediction capabilities. Among these models, the AML model has been found to have performed better in many cases in terms of model fitting and prediction capabilities. The AML model assumes that the discovery rate is symmetric. However, it has been noted that there are cases when the discovery trend is asymmetric. In this paper, we investigate the applicability of using a new vulnerability discovery model called Folded model, based on the Folded normal distribution, and compare it with the AML model. Results show that Folded model performs better than the AML model in general for both model fitting and prediction capabilities in cases when the learning phase is not present.

**Keywords** – Software security; vulnerability discovery model (VDM); Folded model; Risk assessment

## 1 Introduction

The society today relies on the Internet not only for activities such as sending emails, searching the net, and reading news but also security critical tasks such as checking bank account, and online purchasing. As a result, security has become the main concern for both vendors and users of services. Not only the network and communication infrastructure but also software systems themselves at end-nodes need to be secured. Having vulnerabilities, which are software defects that might be exploited by a malicious user causing loss or harm [1], in such systems, could potentially cause a lot of damage. The Code Red worm [2], a computer worm that exploited vulnerabilities existing in Microsoft's IIS (Internet Information Services) in 2001, is an example of the damage that can occur due to presence of vulnerabilities.

Evaluating security quantitatively in software systems is required to achieve an optimal security level. A few quantitative vulnerability discovery models (VDMs) have recently been proposed. They include Rescorla's exponential model

[3], Anderson's thermodynamic model [4], and Alhazmi-Malaiya Logistic (AML) model [5], each of them is based on its own assumptions and is characterized by its specific parameters. The VDMs let developers project the future behavior of the vulnerability discovery processes. The VDMs are essential for two main reasons. First, they allow developers to optimally allocate their resources that will be needed for developing patches for the security holes quickly. Also, they allow the users to assess the potential risk due to new vulnerabilities.

Investigating the prediction capability and accuracy of these models has been studied by Alhazmi and Malaiya [6]. It has been found that the AML model generally fits the data for several software systems better than other models. The AML model is obtained using the assumption that as the market share of a software increases, the rate of vulnerability discovery also increases. When the software starts losing its market share, or when there are a few vulnerabilities remaining to be found, the vulnerability discovery rate decreases [5]. Thus, the motivation of the vulnerability finders, both white hat and black hat, is driven by the market share. The AML model is logistic, and thus the increase and decrease in the discovery process is assumed to be symmetric around the peak. However, it has been noted [7][8], that the discovery rate may not be necessarily symmetrical. This limitation of the AML model can possibly be addressed using alternative models that capture asymmetric behavior.

Kim [7], and Joh and Malaiya [9] have shown that asymmetric VDMs are feasible and have better performance than the symmetric models in some cases. In this paper, we examine the Folded model suggested by Kim [7], as an alternative VDM. Kim however did not examine the model using actual datasets. Here, we examined the applicability of the Folded VDM using actual vulnerability discovery data for four popular software systems. Specifically, we compare the Folded and AML models using goodness of fit and prediction capabilities for these datasets.

The paper is organized as follows. Section 2 presents the background and the related literature on the VDMs. In Section 3, the AML model is discussed and its potential limitations are identified. In section 4, the Folded VDM will be introduced. Section 5 presents the results of the comparison of the AML and Folded models using goodness of fit tests and prediction capabilities. Finally, the concluding comments are given along with the issues that need further research.

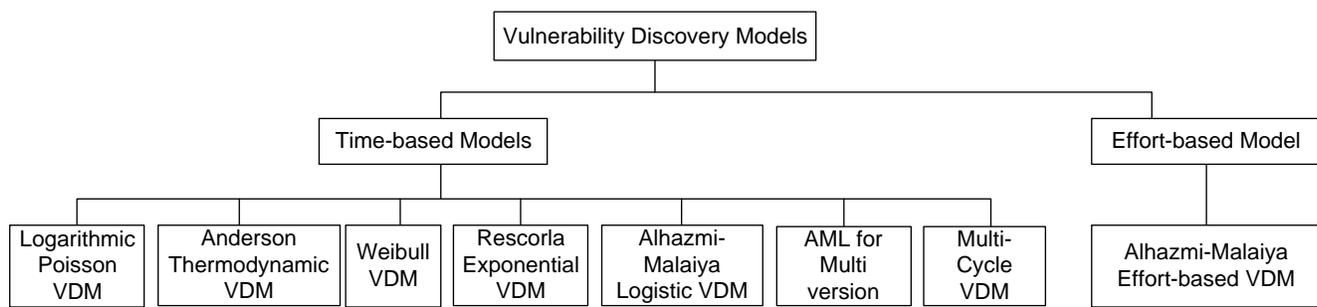


Figure 1. Taxonomy for Vulnerability Discovery Models

## 2 The Vulnerability Discovery Models

The VDMs proposed recently are somewhat analogous to software reliability growth model (SRGM), but there are significant differences. VDMs are probabilistic models for modeling the discovery rate of vulnerabilities in software systems [10]. These models use the historical data such as release date, the discovery date of vulnerabilities and possibly the system usage data. While the vulnerabilities are security related defects, they tend to be treated differently compared with ordinary software defects [11][12]. Normal defects found after release are frequently ignored and not fixed until the next release because they do not represent a high degree of risk. On the other hand, software developers need to patch vulnerabilities right after they are found, due to the high risks they represent. The security issues can greatly impact not only organizations such as banks, brokerage houses, on-line merchants, government offices but also individuals.

Quantitative risk analysis of systems with a continual vulnerability discovery has only recently started to be investigated. A few VDMs proposed by researchers include Anderson [4], Rescorla [3], Kim [7], Alhazmi and Malaiya Logistic model [13], Alhazmi and Malaiya Effort based model [14], Ozment and Schechter [15], and Chen et al. [16]. Figure 1 shows classification of vulnerability discovery models. Each model has its own mathematical representation and parameters. As a result, different VDMs can make somewhat different projections using the same data. No specific guidance is currently available about which models should be used in a given situation.

Rescorla [3] has introduced quadratic and exponential VDMs. He fitted the proposed models but did not evaluate their predictive accuracy. Anderson [4] proposed a thermodynamic vulnerability discovery model, but did not apply the model to any actual data. Alhazmi and Malaiya [5] proposed the logistic vulnerability discovery model, termed the AML model. The AML model presumes a symmetric software vulnerability discovery process. This model has shown a good statistically significant goodness-of-fit for the well-known operating systems such as Windows and Red Hat

Linux, and some Internet applications such as browsers and HTTP servers. Its predictive capability was tested by Alhazmi and Malaiya [6] and it has shown good results. In another study [13], they found that the AML model provides a better goodness-of-fit compared to Rescorla and Anderson models.

Alhazmi and Malaiya [14] have also proposed an effort-based model which utilizes the number of system installations as the independent factor instead of calendar time. They argued that it is much more rewarding to discover a vulnerability in a system which is installed on a large number of computers. However, the effort-based model requires the number of users for a target product in market share which is not always easy to be obtained. Woo et al. [2] have examined the goodness-of-fit as well as the prediction capability for the effort-based model.

Joh et al. [8] have studied Weibull VDM, which was first proposed by Kim [7]. They argued that the assumption made by the AML model that the rate of discovering vulnerability is symmetric around the peak value is not always true. They used Weibull distribution to capture the asymmetric behavior as an alternative to the AML model. However, the Weibull model did not always provide a good fit.

## 3 The Symmetrical AML VDM

The AML VDM [1] is a time-based model. It assumes that at the release of the software the vulnerability discovery rate increases gradually. This is known as the learning phase in which the software gains market share and installed bases remain small. After the learning phase, the system starts to attract more users and the number of vulnerabilities grows linearly. In this phase, which is known as the linear phase, the maximum vulnerability discovery rate is obtained by finding the slope. The learning phase is considered as the most important phase because most of the vulnerabilities will be discovered during this phase. However, when the system starts to be replaced by a newer version and users start to switch to the next version and as a result the vulnerability finders start to lose interest in finding vulnerabilities in the older version. As a result, the vulnerability discovery rate drops. Therefore, the cumulative number of vulnerabili-

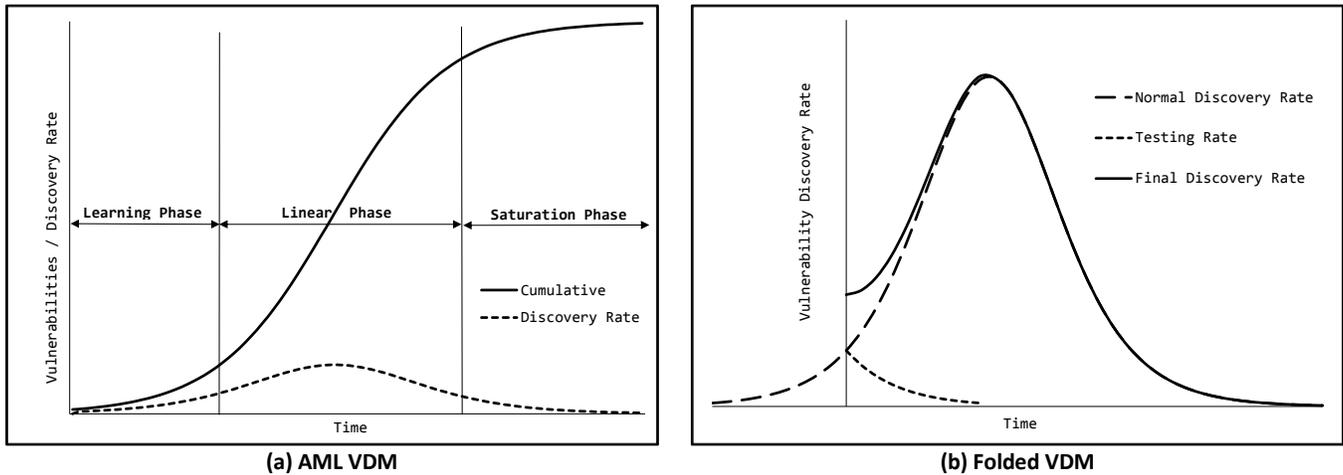


Figure 2. Vulnerability discovery process and rates for AML and Folded VDMs

ties becomes stable. The three phases are shown in Figure 2 (a).

The AML model assumes that the vulnerability discovery processes are controlled by the market share of the software and the number of the undiscovered vulnerabilities. The model assumes that the vulnerability discovery rate is given by the differential equation:

$$\frac{d\Omega}{dt} = A\Omega(B - \Omega) \quad (1)$$

Equation (1) has two factors. The first factor  $A\Omega$ , where  $A$  is a constant, increases as the market share increases, and  $(B - \Omega)$ , where  $B$  represents the total number of vulnerabilities, decreases as the remaining vulnerabilities decreases. Equation (1) can be solved to obtain the logistic expression for  $\Omega(t)$ :

$$\Omega(t) = \frac{B}{BCe^{-ABt} + 1} \quad (2)$$

Note that  $\Omega(t)$  approaches  $B$  as the calendar time  $t$  approaches infinity. The parameters  $A$  and  $C$  determine the shape of the curve [13].  $C$  is a constant introduced while solving Equation (1).

AML model assumes a symmetrical vulnerability discovery rate as shown by the dotted curve in Figure 2 (a). Although the AML model has been found to fit real data of many software systems, there is no compelling reason why the rise and fall should be symmetric since they may be controlled by different factors. Some datasets do show a noticeable asymmetry [9]. These findings violate the symmetric assumption made by this model. Thus, looking for alternative VDMs that can deal with this trend is needed.

Actual data can show a departure from the s-shape assumed by the logistic model. In many cases, a software system gradually evolves as code is modified or patched or additional code is added. This will inject new vulnerabilities into the system which will delay the onset of saturation. In many cases, a new version is widely anticipated and is adapted by many users soon after its release. This will result in the learning period to shrink or even disappear. In the

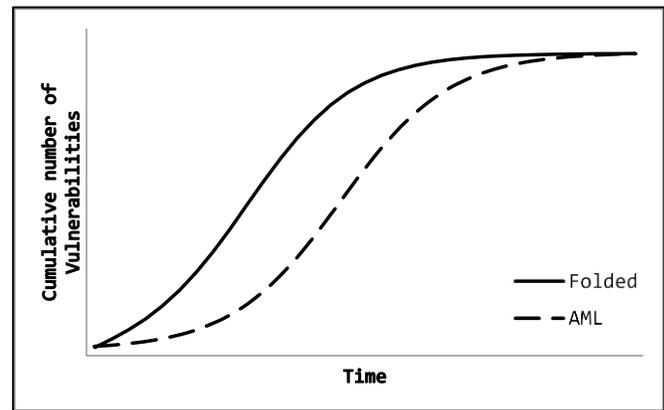


Figure 3. General cumulative vulnerability discovery trends

next section we consider the Folded VDM that offers the capability of modeling the behavior when the learning period is very small.

## 4 Asymmetrical Folded VDM

The normal distribution is symmetric around its mean and is defined for a random variable that takes values from  $-\infty$  to  $+\infty$ . In some cases, a distribution is needed that has no negative values. Daniel [17] had proposed a half-normal distribution that folds the normal distribution at the mean that now corresponds to value zero. A more general version of it was proposed by Leone et al. [18] which is termed a Folded normal distribution that is defined for a random variable taking values between 0 and  $+\infty$ . It is obtained by folding the negative values into the positive side of the distribution. Whenever measurements of a normally distributed random variable are taken and the algebraic sign is discarded, the resulting distribution will be a Folded distribution. The folded distribution has been found usable in industrial practices such as measurement of flatness, straightens, and determination of the centrality of the sprocket holes in motion picture film [18]. The probability density function (pdf)

and the cumulative distribution function (cdf) of the distribution are both derived from their counterparts in the normal distribution, pdf and cdf.

The Folded distribution, as applied to vulnerability discovery, is illustrated in Figure 2 (b). The vulnerability discovery starts at time  $t = 0$  which corresponds to the release time of the software. Since the initial value is non-zero because of the contribution of folding, the learning period is minimized as shown in Figure 3. Hence, here, we propose the Folded VDM as an asymmetrical model as suggested by Kim [7]. The proposed vulnerability discovery rate of the Folded model is given by Equation (3).

$$f(t) = \frac{\gamma}{\sqrt{2\pi}\sigma} \left[ e^{-\frac{(t-\tau)^2}{2\sigma^2}} + e^{-\frac{(t+\tau)^2}{2\sigma^2}} \right], t \geq 0 \quad (3)$$

Here,  $t$  represents the calendar time,  $\tau$  is a location parameter,  $\sigma$  is a scale parameter, and  $\gamma$  represents the number of vulnerabilities that will be eventually discovered. The second term in Equation (3) represents the part of the distribution folded to the positive side as shown in Figure 2 (b) which shows the discovery process for the Folded VDM. The cumulative number of vulnerabilities described by Folded VDM is presented in Equation (4).

$$F(t) = \frac{\gamma}{2} \left[ \operatorname{erf} \left( \frac{t-\tau}{\sqrt{2}\sigma} \right) + \operatorname{erf} \left( \frac{t+\tau}{\sqrt{2}\sigma} \right) \right], t \geq 0 \quad (4)$$

where  $\operatorname{erf}(\cdot)$  is the error function which is used to calculate the integral from zero. Figure 3 shows the cumulative Folded vulnerability discovery process along with the behavior of AML. Figure 3 also shows the lack of the learning phase for the Folded model.

Compared to AML, the Folded VDM has shorter learning phase or missing learning phase which makes the normal distribution asymmetric. It results in a higher discovery rate at the beginning which may be especially applicable to the cases where  $\Omega(t)$  plot is linear even at the beginning.

## 5 Model comparisons and observations

We have fitted the AML and Folded VDMs to the four datasets: Windows 7, OSX 5.x, Apache Web Server 2.0.x, and Internet Explorer 8. Table 1 shows released dates, market shares and the number of vulnerabilities in each system. These software systems have been chosen because they have relatively short learning phase, and thus they can be used to test whether the proposed Folded model is capable of capturing the learningless vulnerability discovery trend. Figure 4 shows model fittings for the two VDMs on the four datasets. While visually both models appear to fit well, in the next section we analyze the goodness of fit by evaluating the p-values.

### 5.1 Goodness of Fit analysis

Table 2 shows the model parameters along with the p-values of  $\chi^2$  goodness of fit tests. The  $\chi^2$  statistic ( $\chi_s^2$ ) is calculated as:

Table 1. Datasets Used

	Released	Vuln.	Share(%)
Win 7	2009-JUL	80	**25.11
OSX5.x	2007-OCT	211	**1.30
Apache 2.0.x	2000-MAR	68	***62.71
IE 8	2009-MAR	72	**33.06

\*<http://nvd.nist.gov/> on JAN 2011. Only after the released date.

\*\*<http://marketshare.hitslink.com/> on APR 2011.

\*\*\*<http://news.netcraft.com/> on May 2011. For total version.

$$\chi_s^2 = \sum_{i=1}^n \frac{(o_i - e_i)^2}{e_i} \quad (5)$$

where  $o_i$  and  $e_i$  are the observed and expected values at  $i^{th}$  time point respectively. The null hypothesis for the test is that the actual distribution is well described by model fittings. Hence, in Table 2, p-value close to 1 means good model fitting whereas less than 0.05 is considered as not being statistically significant when we select the  $\alpha$  level as 0.05.

Figure 4 suggests that all the datasets show linear discovery trends for the period examined and either do not have a learning phase or it is very short. The main reason for linearity in the early part can be because of quick adoption of the version considered as a result of the anticipation of the release. Both the users and the vulnerability finders are not waiting for the software to become sufficiently popular, they take it for granted that it will be. During the later part, the linear behavior could be that since the systems are continually evolving, new code is being injected time to time which introduces additional vulnerabilities. The saturation phases would not be seen in the vulnerability discovery process for such systems until they stop evolving. In general, we observed that Folded VDM captures the starting and ending data points better than AML model for these datasets.

P-values in Table 2 indicate that all the model fittings are statistically significant since p-value is greater than 0.05. Windows 7 and Internet Explorer 8 fit the Folded model better whereas AML fits OSX 5.x slightly better. Apache 2.0.x data fits both models very well with p-value 1. However, visual inspection tells that Folded model performs better at the beginning and the end of the time period. Folded model provides p-values which are consistently greater than 0.9 while AML has a lower value in the two cases.

### 5.2 Prediction capabilities

The main use of a model is predicting the future trends based on the available data, rather than reviewing the past behavior. In that sense, prediction capability should be considered more important than model fitting. Models having good fitting results may not necessarily possess good prediction abilities of the process behavior changes with time.

We use two normalized prediction capability measures [19], Average Error (AE) and Average Bias (AB), as given in Equation (6) and (7) respectively. AE is a measure of how well a model predicts throughout the time period, and AB indicates the general bias of the model which assesses its tendency to overestimate or underestimate.

$$AE = \frac{1}{n} \sum_{t=1}^n \left| \frac{\Omega_t - \Omega}{\Omega} \right| \quad (6)$$

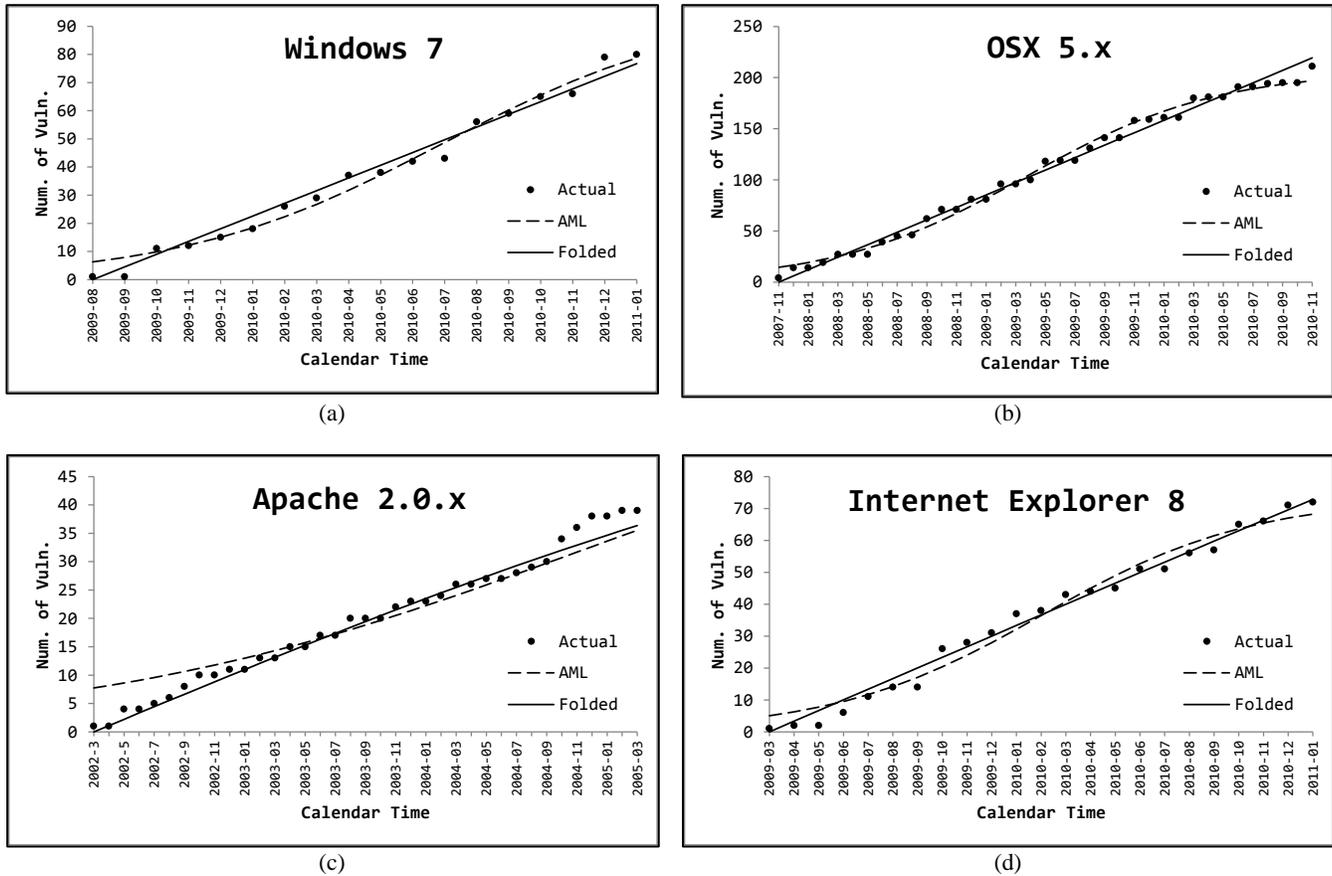


Figure 4. Model fitting for AML and Folded VDMs

Table 2.  $\chi^2$  Goodness of fit tests

	AML				Folded			
	A	B	C	P-value	$\tau$	$\sigma$	$\gamma$	P-value
Win 7	2.52E-03	96.75671	0.148963	0.6970	0.063742	11381.63	64427.18	0.9673
OSX5.x	7.49E-04	206.8057	0.064464	0.9845	0.063742	1969.209	15029.21	0.9428
Apache 2.0.x	9.88E-04	62.69023	0.113327	1.0000	0.065227	47.81145	66.26354	1.0000
IE 8	3.22E-03	73.39949	0.185473	0.7337	0.065227	97.30989	407.1494	0.9839

$$AB = \frac{1}{n} \sum_{t=1}^n \frac{\Omega_t - \Omega}{\Omega} \quad (7)$$

In the equations,  $n$  is a total number of time points (in months in this case), and  $\Omega$  is the actual number of total vulnerabilities.  $\Omega_t$  is the estimated number of total vulnerabilities at time  $t$ . The normalized prediction error values for each time point are plotted in Figure 5. The x-axis represents the time as a percentage where 0% and 100% correspond to the release date and the final data point that the model is attempting to predict. Table 3 shows the values for AE and AB.

The error plots in Figure 5 show that the Folded model provides a more stable prediction with a significantly less error in most situations. In Table 3, the AB and AE values show that the Folded model almost always performs better than AML. For Windows 7, OSX 5.x and Internet Explorer

8, Folded model outperformed the AML. For Apache 2.0.x, the two models result in somewhat similar outcomes for the AE value.

## 6 Conclusion & Future work

This paper examines a new vulnerability discovery model based on the folded normal distribution and evaluates its applicability using real datasets for four major software products. It also compares the new proposed model with the symmetrical AML vulnerability discovery model.

Software developers need to estimate the resources needed for development of patches for the vulnerabilities that are likely to be found in future. A quick patch release after the discovery of a vulnerability will significantly reduce the security risk to the organizational and individual users. An organization needs to assess the resources needed to address

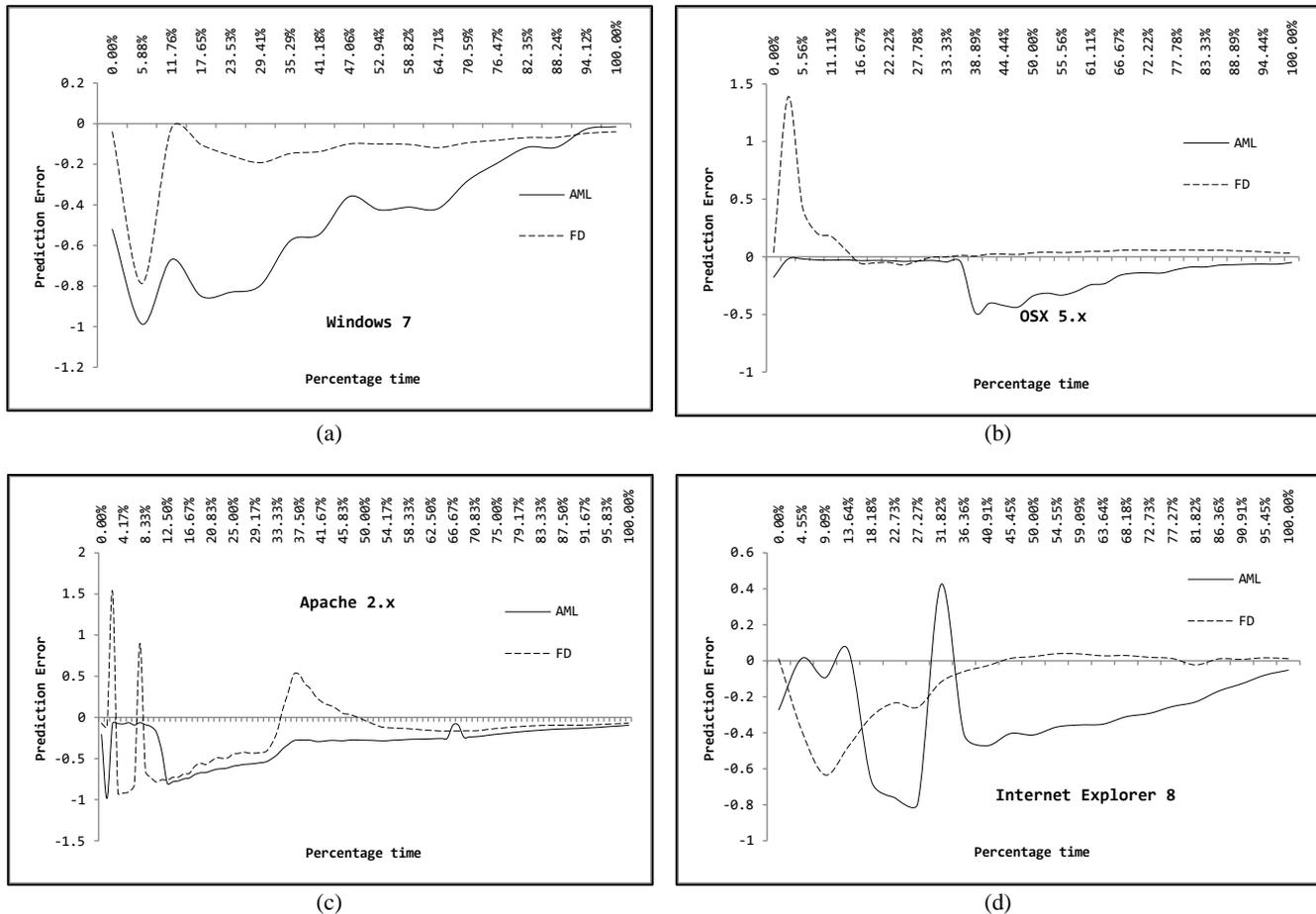


Figure 5. Prediction errors for AML and Folded VDMs

Table 3. Average Bias & Average Error (%-Time: 0% ~ 100%)

	AB		AE	
	AML	Folded	AML	Folded
Win 7	-0.45222	-0.13405	0.452221	0.134048
OSX5.x	-0.14514	0.081817	0.145141	0.096575
Apache 2.0.x	-29.6239	-17.7062	29.62394	28.87495
IE 8	-0.27722	-0.09876	0.320391	0.121494

future vulnerabilities; including the patch application effort and reserve resources needed to alleviate the impact of possible intrusions. Both of these require the use of a vulnerability discovery model that can make sufficiently accurate vulnerability discovery rate projections.

The AML model is the only model that has been formulated to specifically describe the discovery process. The fitting and prediction capability of the AML model has been found to be better than other models for most datasets. However, it has also been found that the discovery trends can be different in different circumstances. In one hand, for the software systems that has been in the market for long period of time, their behavior have been found to be better described by symmetric models such as AML logistic model which exhibits both learning and a saturation phases in addi-

tion to the linear phase. On the other hand, some systems have a vulnerability discovery rate that tends to be linear from the beginning and thus lack a learning phase.

In this paper we have formally defined and investigated the Folded vulnerability discovery model based on folded normal distribution which is asymmetric by definition and can represent a learningless discovery process. Its model fitting and prediction capabilities have been tested and compared with the AML model for four popular software systems. While both Folded and AML models have been found to fit the vulnerabilities datasets of Windows 7, OSX 5.x, Apache Web server 2.0.x and Internet Explorer 8 well, they differ significantly in the prediction capability. The short learning phase is apparently captured by the Folded model much better than the AML logistic model for the four da-

tasets. The folded model consistently outperforms the AML model in terms of the prediction capabilities for the datasets with no learning phase.

The Folded model needs to be further investigated by applying it to as many software systems as possible and comparing it with other competing models. That will allow development of guidelines as to when this model would be most suitable. The significance of the parameters also needs to be examined.

## 7 References

- [1] C. P. Pfleeger and S. L. Pfleeger. *Security in Computing*, 3rd ed. Prentice Hall PTR, 2003.
- [2] S. Woo, H. Joh, O. Alhazmi, and Y. Malaiya. Modeling vulnerability discovery process in Apache and IIS HTTP servers, *Computers & Security*, 30(1), 2011, pp. 50-62.
- [3] E. Rescorla. Is finding security holes a good idea? *IEEE Security & Privacy*, 3(1), 2005, pp.14-19.
- [4] R. Anderson. Security in open versus closed systems - the dance of boltzmann, coase and moore. Proc. Conf. on Open Source Software: Economics, Law and Policy, 2002, pp. 1-15.
- [5] O. Alhazmi, Y. Malaiya, and I. Ray. Security vulnerabilities in software systems: a quantitative perspective. Proc. IFIP WG 11.3 Working Conference on Data and Applications Security, 2005, pp. 281-294.
- [6] O. Alhazmi and Y. Malaiya. Measuring and enhancing prediction capabilities of vulnerability discovery models for apache and IIS http servers. Proc. Int. Symp. Software Reliability Eng. (ISSRE), November 2006, pp. 343-352.
- [7] J. Kim. Vulnerability discovery in multiple version software systems: open source and commercial software systems, Master thesis, Colorado State University, 2007.
- [8] H. Joh, J. Kim, and Y. Malaiya. Vulnerability discovery modeling using Weibull distribution. Proc. Int. Symp. Software Reliability Eng. (ISSRE), November 2008, pp. 343-352.
- [9] H. Joh and Y. K. Malaiya. Modeling skewness in data with S-shaped vulnerability discovery models, Proc. Int. Symp. Software Reliability Eng. (ISSRE), November 2010, pp. 406-407.
- [10] A. Ozment. *Vulnerability Discovery & Software Security*. PhD dissertation, University of Cambridge. August 31, 2007.
- [11] P. Anbalagan and M. Vouk. On reliability analysis of open source software - fedora. Proc. Int. Symp. Software Reliability Eng. (ISSRE), November 2008, pp. 325-326.
- [12] T. Zimmermann, N. Nagappan, and L. Williams. Searching for a needle in a haystack: predicting security vulnerabilities for windows vista. Proc. Int. Conf. on Software Testing, Verification and Validation. (ICST), 2010, pp. 421-428.
- [13] O. Alhazmi and Y. Malaiya. Modeling the vulnerability discovery process. Proc. Int. Symp. Software Reliability Eng. (ISSRE), November 2005, pp.129-138.
- [14] O. Alhazmi and Y. Malaiya. Quantitative vulnerability assessment of system software, Proc. Ann. IEEE Reliability and Maintainability Symp. 2005, pp. 615-620.
- [15] A. Ozment and S. Schechter. Milk or wine: does software security improve with age? Proc. 15th Usenix Security Symposium, Vancouver, Canada, 2006, pp. 93-104.
- [16] K. Chan, D-G Feng, P-R Su, C-J Nie, and X-F Zhang. Multi-cycle vulnerability discovery model for prediction. *Journal of Software*, 21(9), 2010, pp. 2367-2375.
- [17] C. Daniel. Use of Half-normal plots in interpreting factorial two-level experiments, *Technometrics*, 1(4), 1959. pp. 311-341.
- [18] F. C. Leone, L. S. Nelson, and R. B. Nottingham. The folded normal distribution. *Technometrics*, 3(4), 1961, pp. 543-550.
- [19] Y. K. Malaiya, N. Karunanithi, and P. Verma. Predictability of software reliability models. *IEEE Transactions on Reliability*, 41(4), 1992, pp. 539-546.

## Rule-Based Phishing Attack Detection

Ram B. Basnet <sup>a,b,\*</sup>, Andrew H. Sung <sup>a,b</sup>, Quingzhong Liu <sup>c</sup>

<sup>a</sup> Computer Science & Engineering Department, New Mexico Tech, Socorro, NM 87801, USA

<sup>b</sup> Institute for Complex Additive Systems Analysis (ICASA), New Mexico Tech, Socorro, NM 87801, USA

<sup>c</sup> Department of Computer Science, Sam Houston State University, Huntsville, TX 77341, USA

\*Corresponding Author, {rbasnet, sung}@cs.nmt.edu, qxl005@shsu.edu

**Abstract**— The World Wide Web has become the hotbed of a multi-billion dollar underground economy among cyber criminals whose victims range from individual Internet users to large corporations and even government organizations. As phishing attacks are increasingly being used by criminals to facilitate their cyber schemes, it is important to develop effective phishing detection tools. In this paper, we propose a rule-based method to detect phishing webpages. We first study a number of phishing websites to examine various tactics employed by phishers and generate a rule set based on observations. We then use Decision Tree and Logistic Regression learning algorithms to apply the rules and achieve 95-99% accuracy, with a false positive rate of 0.5-1.5% and modest false negatives. Thus, it is demonstrated that our rule-based method for phishing detection achieves performance comparable to learning machine based methods, with the great advantage of understandable rules derived from experience.

**Keywords**- Phishing attack, phishing website, rule-based, machine learning, phishing detection, decision tree

### I. INTRODUCTION

Phishing is a criminal mechanism employing both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials, according to AntiPhishing Working Group (APWG) [1].

Phishing emails usually act on behalf of a trusted third-party to trick email receivers into performing some actions such as giving away personal information, e.g. bank accounts, social security numbers, usernames and passwords to online banking and popular social networking websites like Facebook, Twitter, etc. Though much research on anti-phishing techniques has been done and new techniques and methodologies are being proposed regularly, online scammers manage to come up with innovative schemes to circumvent existing detection technologies and lure potential victims to their phishing campaigns.

Once the phishing email receivers are lured into a fraudulent website, even the experienced, security-minded users are often easily fooled to fulfill the website's primary goal. Data indicates that some phishing attacks have convinced up to 5% of their recipients to provide sensitive information to spoofed websites [9].

Kroll survey [2] finds that phishing is the top information theft threat to North American companies. The survey also

found that the top techniques used for information theft against U.S. companies were phishing.

While payment systems and financial sectors continued to lead the most targeted phishing brands, classifieds emerged as a major non-traditional phishing vector accounting for 6.6% of phishing attacks detected in Q2 2010, growing 142% from Q1, according to APWG quarterly report [1]. Government sector accounted for 1.3% of the phishing attacks in Q2 2010. United States continued its position as the top country for hosting phishing website during the same quarter.

As a result, the design and implementation of effective phishing detection techniques to combat cyber crime and to ensure cyber security, therefore, is an important and timely issue that—as long as the cyber criminals are proceeding unabated in scamming Internet users—requires sustained efforts from the research community.

In this paper, we propose a rule-based approach to detecting phishing webpages and present our preliminary experimental results on temporal data sets using Decision Tree and Logistic Regression learning algorithms.

### II. RELATED WORK

There is an extensive recent literature on automating the detection of phishing attack, most importantly, detection of phishing emails, phishing URLs, and phishing webpages. Phishing attack detection techniques based on the machine learning methodology has proved highly effective, due to the large phishing data set available and the advances in feature mining and learning algorithms; see e.g., [3], [4], [5], [6], [7], [10], [24], [25], [26].

Anomaly detection has been used to detect phishing webpage [32] where a number of anomaly based features are extracted from webpage and SVMs is applied on a data set with 279 phishing and 100 legitimate webpages producing 84% classification accuracy.

Besides machine learning (ML) based techniques, there exists a plethora of other approaches in phishing detection. Perhaps, the most widely used anti-phishing technology is the URL blacklist technique that most modern browsers come equipped with [14], [30]. Other popular methods are browser-based plug-ins or add-in toolbars. SpoofGuard [16] is one such tool that uses domain name, URL, link, and images to evaluate the spoof probability on a webpage. The

plug-in applies a series of tests, each resulting in a number in the range (0, 1). The total score is a weighted average of the individual test results. Other similar anti-phishing tools include SpoofStick [19], SiteAdvisor [17], Netcraft anti-phishing toolbar [23], AVG Security Toolbar [22], etc.

Visual similarity based methods have been explored for detecting phishing pages. Weynain et al. [21] compare legitimate and spoofed webpages and define visual similarity metrics. The spoofed webpage is detected as a phishing attack if the visual similarity is higher than its corresponding preset threshold. Medvet et al. [11] consider three key page features, text pieces and their style, images embedded in the page, and the overall visual appearance of the page as rendered by the browser.

### III. RULE-BASED APPROACH

In this section, we discuss motivation of our approach and the underlying techniques we propose to achieve our goal.

#### A. Motivation

Though different in goal, our approach is particularly inspired by the approach introduced by the open source intrusion detection and prevention system (IDS/IPS), Snort [31]. Snort monitors networks by matching each packet it observes against a set of rules. As the phishing attacks have been growing rapidly by the day, we feel that there is a need for Snort like phishing attack detection technology at the application level. In this paper, we try to investigate such an approach.

#### B. Our Approach

Just like a network IDS signature, a rule is a pattern that we want to look for in a webpage. The idea behind the rule-based approach is to make the process of phishing attack detection as intuitive, simple, and user-friendly as possible. One of the main goals of our approach is to make the framework flexible and simple to extend the rule set by incorporating new and emerging phishing tactics as they are encountered. We generate our rule set primarily relying on our observations and the machine learning features proposed in various existing literatures [3], [4], [5], [6], [10] on phishing attack detection. We gather various techniques and tricks used by phishers to lure their potential victims to a forged website and use those heuristics to develop our initial rule set. In this section, we briefly describe various rules that we employ in detecting whether a given webpage is phishing.

A rule is usually written in the following form:

IF *conditions* THEN *actions*

If the *conditions*, also known as *patterns*, are satisfied then the *actions* of that particular rule are fired.

A rule may range from very simple – checking a particular value in the URL – to highly complex and time-consuming that may require to analyze meta-data, query search engines and blacklists and combine several

conditions with *AND* and *OR* operators. Depending on their characteristics and the methods used to extract the rules, we broadly group them into the following categories.

#### C. Search Engine-based Rules

The idea behind using results from top search engines is to leverage their power and effectiveness in continuously crawling and indexing a large number of webpages. In [3], we show that search-engine based features are very effective in determining phishing URLs and essentially demonstrate that search engines' large and continuously growing indexes act as a rudimentary white-list. We develop two rules using search engines.

**Rule 1:** IF a webpage's URL is not present in all search engines' indexes, THEN the webpage is potentially phishing.

**Rule 2:** IF a webpage's domain is not present in all search engines' indexes, THEN the webpage is potentially phishing.

To generate Rule 1, we check if a URL exists in the search engines' (Google, Yahoo!, and Bing) indexes. Our rule generator automatically queries the search engines and retrieves top 30 results. If the results do not contain the URL, this rule considers the webpage as potentially a phishing attack. We observed that all three search engines returned the URL as the first result if they have indexed the URL. Intuitively, it makes sense because we search the URL itself not ranked relevant URLs based on keywords. But, to be on the safe side, we use top 30 results as it has been shown that going beyond the top 30 results had little effect [6].

Similarly, Rule 2 is generated by querying the search engines with the domain of a URL. If the top 30 results do not contain the domain, this rule says that the given webpage is potentially phishing.

#### D. Red Flagged Keyword-based Rule

By examining 80% of randomly selected URLs on DS1 data set, we found that certain groups of words seem to be more popular among phishers, perhaps, to lure unsuspecting users to the forged webpage. Using substring extraction algorithm, we generated a list of 62 word stems that frequently occur in our training data set. We iterate through this keyword list and check if any of the word is found in the URL. Thus, we generate our next rule:

**Rule 3:** IF a keyword is present in the URL, THEN the webpage is likely phishing.

#### E. Obfuscation-based Rules

Phishers often obfuscate URLs to trick users into thinking that the malicious URL belongs to a legitimate website users are familiar with. Obfuscating URLs with certain characters such as “-”, soft hyphen, Unicode, and visually similar looking characters are very common techniques employed by phishers. We try to identify these tactics and generate rules from them. For example, we check if certain

characters such as “-”, “\_”, “=”, “@”, digits, and non-standard port etc. are present in a webpage’s URL.

These tactics used by phishers lead us to our next set of rules.

**Rule 4:** IF a webpage’s URL is IP based (hex-based, octal, or decimal-based), THEN the webpage is potentially a phishing attack.

**Rule 5:** IF a URL contains any of the following characters [-, \_, 0-9, @, “, ”, ;] OR contains a non-standard port, THEN the webpage is potentially phishing.

**Rule 6:** IF host part of a URL has 5 or more dots OR length of the URL is longer than 75 characters OR length of the host is longer than 30 characters, THEN the webpage is potentially a phishing attack.

#### F. Blacklist-based Rule

We employ Google Safe Browsing API [14] to check URLs against Google’s constantly updated blacklists of suspected phishing and malware pages and generate our next rule.

**Rule 7:** IF a URL is in Blacklist(s), THEN it is potentially a phishing webpage.

#### G. Reputation-based Rule

We generate our next set of rules from historical stats on top IPs and domains that have a bad reputation of hosting the most phishing webpages. We use 3 types of statistics: Top 10 Domains, Top 10 IPs, and Top 10 Popular Targets published by PhishTank [33]. We also use top 50 IP address stat produced by StopBadware.org [13].

**Rule 8:** IF a URL contains a top phishing target OR its IP or domain is in the statistical reports produced by PhishTank, Stopbadware, etc., THEN the webpage is potentially a phishing attack.

#### H. Content-based Rules

The rules in this category are rooted in the HTML contents of the phishing webpages. An ingenious phishing webpage resembles the look and feel of the target legitimate website. Nevertheless, the same tactics employed by phishers also give us opportunities to discover our content-based rules. By observing HTML structures of hundreds of phishing webpages, we’ve generated the following rules:

**Rule 9:** IF a webpage contains *password* input field AND (the corresponding form content is sent in plain text without using Transport Layer Security (TLS)/Secure Sockets Layer (SSL) OR the form content is sent by using ‘get’ method), THEN the webpage is potentially phishing.

**Rule 10:** IF a webpage contains *password* input field AND the corresponding form content is sent to external domain regardless of TLS/SSL, THEN the webpage is potentially phishing.

**Rule 11:** IF a webpage contains META tag AND the refresh property’s destination URL is in external domain OR it belongs to a blacklist, THEN the webpage is potentially phishing.

**Rule 12:** IF a webpage is redirected by its server AND the page contains *password* field, THEN the webpage is potentially phishing.

**Rule 13:** IF a webpage has IFrame tag AND its source URL belongs to a blacklist, THEN the webpage is potentially a phishing attack.

**Rule 14:** IF a webpage contains *password* input field AND the webpage has more external than internal links, THEN the webpage is potentially phishing.

**Rule 15:** IF a webpage has bad HTML markups AND contains *password* input field, THEN the webpage is potentially a phishing attack.

Figure 1 shows the histograms of Rules 1-15 obtained on data set DS1. The histogram (see Figure 1) confirms that Rule 1 and Rule 2 have high prominence in phishing webpages and are very strong indicators of whether a webpage is phishing. These rules by themselves can detect more than 97% of phishing webpages, while correctly classifying 100% of legitimate webpages. Rule 3 has high prominence in phishing webpages as well compared to non-phishing webpages.

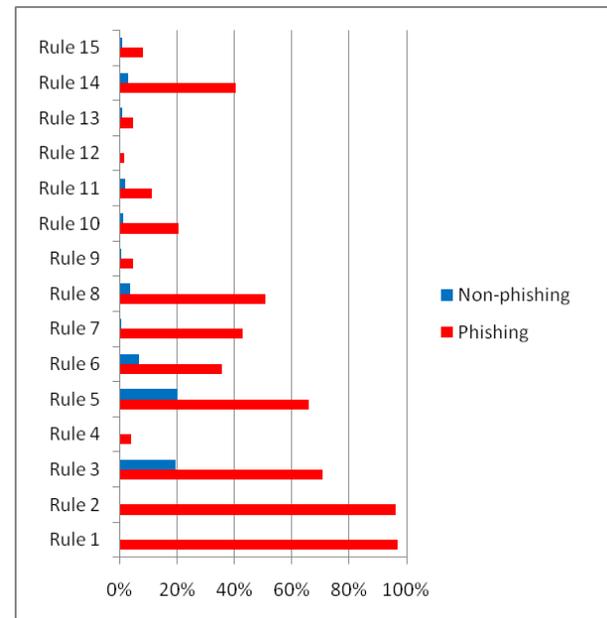


Figure 1: Histogram of Rules 1-15 on DS1 data set

Some phishing webpages (~ 4%) satisfy Rule 4, while not a single non-phishing webpage satisfies it. Though very sparsely present, this rule can be a good indicator of whether a webpage is phishing. Roughly 66% of phishing webpages and surprisingly 20% of non-phishing webpages satisfy Rule 5.

43% of phishing webpages satisfy Rule 7, while 0% of non-phishing webpages satisfy the same. This suggests that Rule 7 is a strong indicator of whether a page is phishing. Rule 8 is present in about 51% of phishing webpages and in roughly 4% of non-phishing webpages. Rule 9 has relatively small presence among phishing webpages but no presence

on non-phishing webpages, indicating that this rule is not universally applicable, but still a strong indicator of phishing webpage.

About 21% of phishing and 1% of non-phishing webpages satisfy Rule 10. Relying on this rule alone would miss a large percentage of phishing webpages while it would also misclassify some legitimate webpages as phishing. Rule 12 is satisfied by a very small number of phishing webpages (~1%). However, no single non-phishing webpage satisfy the same suggesting that this rule may not aid in false positives.

Relatively more phishing webpages satisfy Rule 13, 14, and 15 compared to non-phishing webpages.

We point out that these are not the exhaustive list of rules. One of the major advantages of rule-based approach is to be able to quickly tune the rules to ones' needs and easily modify or add rules as and when needed to detect new and ever changing phishing attacks.

#### IV. EXPERIMENTAL EVALUATION

In this section, we briefly describe the data sets we use and present the results of experimental validation of our approach on these data sets. The experiments were carried out on a machine with Core 2 Duo 2 GHz Intel processors and 3 GB RAM.

##### A. Data Sets

For phishing webpages, we wrote Python scripts to automatically download confirmed phishing URLs from PhishTank [8]. PhishTank, operated by OpenDNS, is a collaborative clearing house for data and information about phishing on the Internet. A potential phishing URL once submitted is verified by a number of registered users to confirm it as phishing. We collected first set of phishing URLs from June 1 to October 31, 2010. Phishing tactics used by scammers evolve over time. In order to investigate these evolving tactics and to closely mimic the real-world *in the wild* scenario, we collected second batch of confirmed phishing URLs that were submitted for verification from January 1 to May 3, 2011.

We collected our legitimate webpages from two public data sources. One is the Yahoo! directory<sup>1</sup>, the web links in which are randomly provided by Yahoo's server redirection service [34]. We used this service to randomly select a URL and download its page contents along with server header information. In order to cover wider URL structures and varieties and page contents, we also made a list of URLs of most commonly phished targets (using statistics from PhishTank [33]). We then downloaded those URLs, parsed the retrieved HTML pages, and harvested and crawled the hyperlinks therein to also use as benign webpages. We made the assumption, which we think is reasonable, to treat those webpages as benign, since their URLs were extracted from a legitimate sources. These webpages were crawled between

September 15 and October 31 of 2010. The other source of legitimate webpages is the DMOZ Open Directory Project<sup>2</sup>. DMOZ is a directory whose entries are vetted manually by editors.

Based on the date on which phishing URLs were submitted to PhishTank for verification, we generated two data sets. The first data set, we refer to it as DS1, contains 11,341 phishing webpages submitted before October 31, 2010 and 14,450 legitimate webpages from Yahoo! and seed URLs. The second data set, we refer to it as DS2, contains 5,456 phishing webpages submitted for verification between January 1 and May 3 of 2011 and 9,636 randomly selected legitimate webpages from DMOZ. Table I summarizes these data sets.

TABLE I  
SUMMARY OF DATA SETS

Data Set	Phishing	Non-phishing	Total Samples
DS1	11,341	14,450	25,791
DS2	5,456	9,636	15,092
DS1+DS2	16,797	24,086	40,883

We discarded the URLs that were no longer valid as the page couldn't be accessed to extract features from their contents.

##### B. Counting Rules to Detect Phishing Webpages

In order to detect a phishing webpage based on rules, one naïve yet simple approach is to give equal weight to each rule and count the number of rules satisfied by the page. Using a carefully chosen threshold, if the total number of rules satisfied by an instance is more than the threshold value, we can alert that the webpage is phishing. However, as the histogram shows (see Figure 2), choosing the best threshold value that would give the balanced and best false positive and negative rates is not a trivial task. Histogram in Figure 2 shows rule count from 0 up to 10 as only a very few phishing instances in the data set satisfied more than 10 rules.

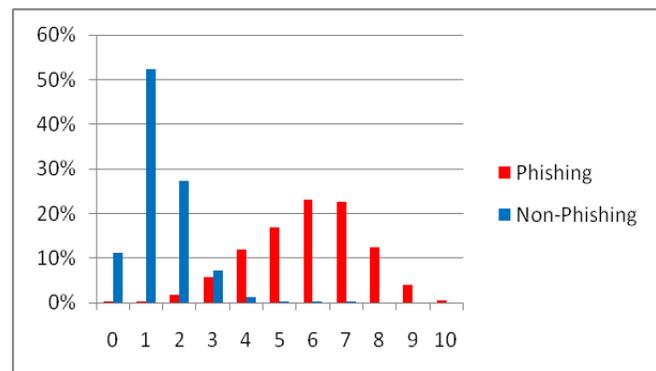


Figure 2: Histogram of rules count on the data set DS1. Horizontal axis is rule count and vertical axis is instance count

<sup>1</sup> <http://dir.yahoo.com>

<sup>2</sup> <http://www.dmoz.org>

Nonetheless, we experimented with a few thresholds and present the results, True Positive Rate (TPR), False Positive Rate (FPR), False Negative Rate (FNR), and True Negative Rate (TNR) in Table II.

TABLE II  
PERFORMANCE MEASURES FOR VARIOUS THRESHOLD VALUES

Rule Count Threshold	TPR	TNR	FPR	FNR
2	99.55%	63.60%	36.40%	0.45%
3	97.64%	91.04%	8.96%	2.36%
<b>4</b>	<b>91.87%</b>	<b>98.20%</b>	<b>1.80%</b>	<b>8.13%</b>
5	79.96%	99.63%	0.37%	20.04%

As the equally weighted count-based approach resulted in unsatisfactory results, we resorted to ML-based approach to automatically prioritize each rule and generate decision rules. The next experiments detail this approach.

### C. Training with Rules

In these experiments, we used the rules identified in Section III.B as binary features and applied them to train classifiers. We used Decision Tree (DT) and Logistic Regression (LR) algorithms implemented in the WEKA data mining library [15]. We employed 10-fold cross validation method to test the models and perform our analysis.

**DT:** DT is represented by tree structure where each internal node tests the corresponding attribute, each branch corresponds to attribute value, and each leaf node represents a classification decision [35].

Using a given input/output data set, DT can be learned by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. Once the tree is trained, an unknown sample is classified by successive tests from the root of a DT down to a leaf.

For DT learning, we chose the C4.5 [20] algorithm which is implemented as J48 classifier in WEKA. On data set DS1, we obtained the pruned decision tree of size 19 with 10 leaf nodes (see Figure 3).

Rules 2, 4, 9, 10, 11, and 12 are removed from the model as a result of pruning.

Besides simple to understand and interpret, DT is robust and uses a white box model. DT model can be converted to rules using boolean logic as following: IF (Rule<sub>1</sub> <= 0) AND (Rule<sub>7</sub> <= 0) AND (Rule<sub>14</sub> <= 0) THEN phishing = No. Using this sequence of tests, 14,154 of non-phishing samples from DS1 data set are correctly classified, while 173 phishing webpages are misclassified. Similarly, IF (Rule<sub>1</sub> > 0) THEN phishing = Yes. Using this rule, 10,976 phishing webpages are correctly classified, while producing 0 false positive. The rest of the 8 rules can be generated and interpreted in the similar manner.

It took 3.35 seconds to build C4.5 model. The detailed test results are shown in Table IV.

```

Rule_1 <= 0
| Rule_7 <= 0
| | Rule_14 <= 0: -1 (14154.0/173.0)
| | Rule_14 > 0
| | | Rule_8 <= 0
| | | | Rule_3 <= 0: -1 (384.0/27.0)
| | | | Rule_3 > 0
| | | | | Rule_5 <= 0: -1 (88.0/20.0)
| | | | | Rule_5 > 0
| | | | | | Rule_13 <= 0: +1 (33.0/7.0)
| | | | | | Rule_13 > 0
| | | | | | | Rule_6 <= 0
| | | | | | | | Rule_15 <= 0: +1 (5.0/1.0)
| | | | | | | | Rule_15 > 0: -1 (8.0/3.0)
| | | | | | | | Rule_6 > 0: -1 (14.0)
| | | Rule_8 > 0: +1 (45.0/6.0)
| Rule_7 > 0: +1 (84.0/11.0)
Rule_1 > 0: +1 (10976.0)

```

Figure 3: DT model using C4.5 on data set DS1

**LR:** LR is a statistical model used for prediction of the probability of occurrence of an event by fitting data to a sigma function logistic curve [18]. Besides high classification accuracy, LR has the advantage of performing automatic feature ranking as well as providing an interpretable linear model of the training data. Because the output of a linear model depends on the weighted sum of the features, the sign and magnitude of the individual parameter vector coefficients can tell us how individual features contribute to a ‘phishing’ or a ‘non-phishing’ prediction. Positive coefficients correspond with phishing features while negative coefficients correspond with legitimate non-phishing features. A zero coefficient means that the corresponding feature will not contribute to the prediction outcome.

The coefficients and the odds ratio obtained from LR model on data set DS1 are displayed in Table III.

As indicated by the high odds ratio, Rule 4 is found to be the most useful in detecting whether a webpage is phishing. Similarly, Rules 1, 2, and 12 are strong indicators that a webpage is phishing attack. Interestingly, Rules 10, 13, and 15, on the other hand, seem to indicate that a webpage is non-phishing. LR took 2.33 seconds to build model and gave classification error rate of 1.02%, TPR of 97.88%, and FPR of 0.15%. The performance difference between C4.5 and LR is insignificant.

### D. Data Drift

Phishing tactics and attack techniques keep changing as attackers come up with novel ways to circumvent the existing filters. Rules developed from observing a particular data set can yield a highly accurate classification results when trained and tested on disjoint sets of the same data source. But do these results hold when testing new phishing

TABLE III  
FEATURES AND THEIR COEFFICIENTS USING LOGISTIC REGRESSION

Feature	Logistic Coefficient	Odds Ratio
Rule 1	80.4784	8.94E+34
Rule 2	63.5746	4.07E+27
Rule 3	1.4302	4.1796
Rule 4	138.2439	1.09E+60
Rule 5	1.5243	4.5917
Rule 6	0.3788	1.4606
Rule 7	5.008	149.6037
Rule 8	2.2699	9.6781
Rule 9	0.9032	2.4675
Rule 10	-0.155	0.8564
Rule 11	0.9984	2.714
Rule 12	58.7186	3.17E+25
Rule 13	-0.3982	0.6715
Rule 14	3.2524	25.852
Rule 15	-0.497	0.6084
Constant	-5.8523	

webpages using the same rule set extracted from old phishing webpages? To investigate this question, we tested new phishing data set DS2 against the model obtained by training the C4.5 classifier on old data set DS1. Table IV shows classification results using C4.5 classifier on various combinations of temporal data sets.

TABLE IV  
OVERALL ERROR RATES ON TRAINING ON ONE DATA SET AND TESTING ON ANOTHER (POSSIBLY DIFFERENT OR TEMPORAL-BASED) DATA SET

Training	Testing	
	DS1	DS2
DS1	0.98%	4.86%
DS2	1.22%	4.51%
DS1+DS2	0.96%	4.18%

As expected, when trained and tested using the same data set DS1, the error yielded is the lowest due to low FPR (0.21%) and FNR (1.9%). When training and testing sources are completely mismatched, the error ranges from 1.2% to 4.8%. Surprisingly, the error received on training and testing using DS2 is comparatively higher (4.5%) with 1.3% FPR and 10.2% FNR. Although, error rate doesn't significantly decrease with the newer phishing data, the disparity in accuracy emphasizes that rules and training data should be selected judiciously. Thus, it is important to collect data that is representative and retrain the deployed classifier with new data often. Finding an optimal time interval to retrain the system for optimum performance represents an interesting direction for future study but is beyond the scope of this paper.

## V. DISCUSSION

In this section, we discuss some of the limitations of rule-based approach and some possible ways to address them.

### A. Tuning False Positives & Negatives

Machine learning models allow us to tune the tradeoff between false positives and negatives. Figure 4 shows the results of this experiment as an ROC graph with respect to the decision threshold  $t$  over an instance of DS1 data set using C4.5 classifier.

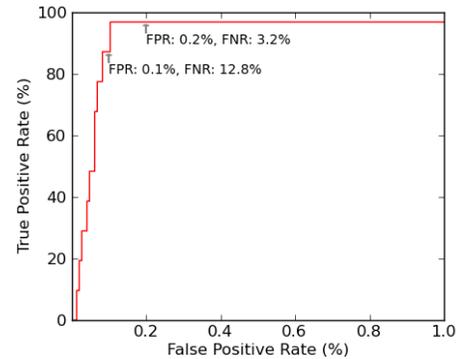


Figure 4: ROC showing tradeoff between false positives and false negatives using C4.5 on DS1 data set

Instead of using decision threshold  $t$  to minimize the overall error rate, Internet users may want to tune the threshold to have very low false positives at the expense of more false negatives or vice versa. By tuning false positives to conservatively low 0.1%, we can achieve false negatives of 12.8%. By tolerating slightly higher false positives of 0.2%, however, we can achieve significantly lower false negatives of 3.2%.

### B. Limitations of Our Approach

The rule set is premature and we emphasize its needs for expansion and thorough scrutiny. The system, if deployed, will likely produce some false alarms, while also missing a good number of phishing webpages. Attackers may thwart the system by minimizing the phishing tricks matching none or a small number of rules on their crafted phishing webpage. One such scenario is when attackers hack a legitimate webpage to host their phishing campaign. Such legitimate webpages are highly likely to appear on search engines' results. Phishers may design flash-based webpages virtually hiding all the HTML contents for analysis. However, expanding our rule set may address such potential attacks. We can add visual similarity-based rules, for instance. An interesting research area would be to expand the rule set using the tactics used in phishing emails and use it to detect phishing attack via emails.

URL-shortening services are growing in popularity thanks to micro-blogging websites such as Twitter<sup>3</sup>. In order to take advantage of the popularity and the obscurity provided by these shortening services, scammers are now establishing their own fake URL-shortening services [27]. Under this scheme, shortened links created on these fake

<sup>3</sup> <http://twitter.com>

URL-shortening services are further shortened by legitimate URL-shortening sites. These links are then distributed via phishing emails, blogs, micro-blogs, and social networking websites. We use the Python library [12] to automatically detect and expand shortened URLs.

To address this, additional rule could be determined such as: IF a URL is shortened by unsupported URL-shortening service, THEN the webpage is potentially phishing attack. Because our data set doesn't have any webpage satisfying this rule, we do not include it in our current rule set.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed and evaluated a rule-based phishing attack detection technique. By analyzing a large number of phishing webpages and combining various features used in ML approach, we generated our 15 initial rule set. These rules were then used as features in Decision Tree and Logistic Regression learning algorithms and their performance results were compared. C4.5 and LR gave competitive accuracy of 99% and FPR of 0.5% and FNR of 2.5%. Their performance slightly degraded, however, when tested with new data sets against models trained with old data set.

As future work, we plan to work on refining the rules to improve on false positives and negatives on newer data sets. Then we plan to develop a rule-based, light-weight, real-time phishing attack detection system like Snort; deploy and test the system in the real world.

## ACKNOWLEDGMENT

The authors would like to acknowledge the generous support received from ICASA (the Institute for Complex Additive Systems Analysis), a research division of New Mexico Tech.

## REFERENCES

- [1] "Antiphishing.org. 2010 2<sup>nd</sup> Quarter Report," 2011. [Online]. Available: [http://apwg.org/reports/apwg\\_report\\_q2\\_2010.pdf](http://apwg.org/reports/apwg_report_q2_2010.pdf).
- [2] KROLL, Global Fraud Report. Accessed on April 10, 2011. [http://www.kroll.com/about/library/fraud/Oct2010/region\\_northamerica.aspx](http://www.kroll.com/about/library/fraud/Oct2010/region_northamerica.aspx).
- [3] R. B. Basnet, A. H. Sung, D. Ackley, and Q. Liu, "A Web Mining Approach to Detecting Phishing URLs," *Computers & Security*, Elsevier, (submitted), 2011.
- [4] R. B. Basnet, S. Mukkamala, and A. H. Sung, *Detection of phishing attacks: A machine learning approach*. Studies in Fuzziness and Soft Computing, 226:373-383, Springer, 2008.
- [5] J. Ma, L. K. Saul, S. Safage, and G. M. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," in *Proc. ACM SIGKDD Conference*, pp. 1245-1253, Paris, France, June 2009.
- [6] Y. Zhang, J. Hong, and L. Cranor, "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites," in *WWW 2007*, Banff, Alberta, Canada, May 2007, ACM Press.
- [7] C. Whittaker, B. Ryner, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," in *Proc. 17<sup>th</sup> Annual Network and Distributed System Security Symposium*, CA, USA, March 2010.
- [8] PhishTank, Out of the Net, into the Tank. [Online]. Available: [http://www.phishtank.com/developer\\_info.php](http://www.phishtank.com/developer_info.php).
- [9] R. Dhamija, J. D. Tygar, and M. Hearst, "Why Phishing Works," *CHI 2006*, Montreal, Quebec, Canada, April 2006.
- [10] S. Doshi, N. Provos, M. Chew, and A. D. Rubin, "A Framework for Detection and Measurement of Phishing Attacks," in *Proc. ACM Workshop on Rapid Malcode (WORM)*, Alexandria, VA, Nov. 2007.
- [11] E. Medvet, E. Kirda, and C. Kruegel, "Visual-Similarity-Based Phishing Detection," in *Proc. 4<sup>th</sup> International Conference on Security and Privacy in Communication Networks*, New York, NY, USA, 2008.
- [12] PyLongURL. Python Library for LongURL.org. [Online]. Available: <http://code.google.com/p/pylongurl/>.
- [13] StopBadware – IP Address Report – Top 50 by Number of Reported URLs. [Online]. Available: <http://stopbadware.org/reports/ip>.
- [14] Google Safe Browsing API. [Online]. Available: <http://code.google.com/apis/safebrowsing/>.
- [15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, Vol. 11, Issue 1, 2009.
- [16] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. Mitchell, "Client-side defense against web-based identity theft," in *11<sup>th</sup> Annual Network and Distributed System Security Symposium (NDSS '04)*, San Diego, USA, 2004.
- [17] McAfee SiteAdvisor Software – Website Safety Ratings and Secure Search. [Online]. Available: <http://www.siteadvisor.com/>.
- [18] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: A Statistical view of Boosting," *The Annals of Statistics* 2000, Vol. 28, No. 2, pp. 337-407, 2000.
- [19] Spooftick Home. [Online]. Available: <http://www.spooftick.com/>
- [20] J. R. Quinlan, "C4.5 Programs for Machine Learning," Morgan Kaufmann Publishers, San Mateo, CA, USA, 1993.
- [21] L. Weynin, G. Huan, L. Xiaoyue, Z. Min, and X. Deng, "Detection of Phishing Webpages based on Visual Similarity," in *WWW '05: Special interest tracks and posters of the 14<sup>th</sup> International Conference on World Wide Web*, pp. 1060-1061, New York, NY, USA, 2005. ACM Press.
- [22] AVG Security Toolbar. [Online]. Available: <http://www.avg.com/product-avg-toolbar-tlbr#tba2>
- [23] Netcraft Anti-Phishing Toolbar. [Online]. Available: <http://toolbar.netcraft.com/>.
- [24] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi, "A Proposal of the AdaBoost-Based Detection of Phishing Sites," in *Proc. 2<sup>nd</sup> Joint Workshop on Information Security (JWIS)*, Aug. 2007.
- [25] I. Fette, N. Sadeh, and A. Tomasic, "Learning to Detect Phishing Emails," in *Proc. 16<sup>th</sup> International Conference on World Wide Web*, pp. 649-656, May 2007.
- [26] R. B. Basnet and A. H. Sung, "Classifying Phishing Emails Using Confidence-Weighted Linear Classifiers," in *Proc. International Conference on Information Security and Artificial Intelligence*, pp. 108-112, Chengdu, China, Dec. 2010.
- [27] Symantec.cloud. MessageLabs Intelligence Reports. [Online]. Available: [http://www.messagelabs.com/mlireport/MLI\\_2011\\_05\\_May\\_FINAL-en.pdf](http://www.messagelabs.com/mlireport/MLI_2011_05_May_FINAL-en.pdf). Accessed May 25, 2011.
- [28] L. Richardson. Beautiful Soup. [Online]. Available: <http://www.crummy.com/software/BeautifulSoup/>.
- [29] Microsoft Security Intelligence Report, Vol.10, 2011. [Online]. Available: <http://www.microsoft.com/security/sir/default.aspx>.
- [30] SmartScreen Filter – Microsoft Windows. [Online]. Available: <http://windows.microsoft.com/en-US/internet-explorer/products/ie-9/features/smartscreen-filter>, 2011.
- [31] M. Roesch, "Snort – Lightweight Intrusion Detection for Networks." [Online]. Available: <http://assets.sourcefire.com/snort/developmentpapers/Lisapaper.txt>.
- [32] Y. Pan and X. Ding, "Anomaly Based Web Phishing Page Detection," in *Proc. 22<sup>nd</sup> Annual Computer Security Application Conference, (ACSAC '06)*, pp. 381-392, Miami Beach FL, USA, Dec. 2006.
- [33] PhishTank – Statistics about phishing activity and PhishTank usage. [Online]. Available: <http://www.phishtank.com/stats.php>
- [34] Yahoo! Inc.: Random Link - random. [Online]. Available: <http://random.yahoo.com/fast/ryl>.
- [35] T. M. Mitchell, "Machine Learning," McGraw Hill, 1997.

# Technology Risk Management Plan for an Online University

Frizella Donegan

Information Assurance and Security  
Capella University  
225 South 6th Street,  
Minneapolis, MN 55402 USA  
Email: frizelladonegan@yahoo.com

Syed (Shawon) M. Rahman, Ph.D.

Assistant Professor, University of Hawaii-Hilo and  
Adjunct Faculty at Capella University  
200 W. Kawili St,  
Hilo, HI 96720 USA  
Email: SRahman@hawaii.edu

**Abstract** - An online University has more than enough reasons to be concerned about information security. The major concerns revolve mitigating risk around securing corporate financial data, keeping students/faculty/staff personal information safe, continuing daily operation, and protecting other assets. In an effort to be secure, organizations are creating comprehensive Risk Management Plans to ensure mission critical systems and process are secure as well as able to be up and running or recover in a reasonable amount of time in the case of a natural disaster or cyber attack. This paper<sup>1</sup> has focused on an international online university's Technology Risk Management team's Risk Management Plan and the process the teams performs. However, regardless to the size of the university or organization, risk mitigation management is a crucial component of any organization's security framework.

**Keywords**- Information Security, Risk Management Plan, Incident Reporting, Business Continuity, Risk Mitigation

## 1. INTRODUCTION

An organization's Risk Management plan is utilized to identify risk and document a plan to reduce risk and actions to take. Furthermore, it is used to monitor and control risk in an effective manner. Risk management plans can include various components which some are covered in this research study. Input error controls are highlighted as it pertains to enterprise risk reduction activities. Strategies are discussed with focus on risk mitigation plan and risk planning; in addition, change control and disaster recovery procedures are noted along with email based security tools. Risk management includes elements of incident reporting and handling to mitigate risk, this section details risk reporting and communication plan. Business continuity plan (BCP) and disaster recovery are important elements of mission critical online university functions, this paper provides an outline necessary parts of the BCP. The termination process of the overall program addresses risk from within. Lastly, a description of a physical off-site facility in case a disaster shuts down primary locations. This study highlights important features for a solid foundational facility.

<sup>1</sup> This work is partially supported by EPSCoR award EPS-0903833 from the National Science Foundation to the University of Hawaii

## 2. INPUT ERROR CONTROLS

This section describes and identifies controls that can be applied to applications in order to best prevent input errors and develop a robust and extensive focus on enterprise risk management. Within an organization, each application area are required to provide risk-reduction control activities that specify whether the activity is preventive or detective, manual or automated. The activity supports the following control objectives: completeness, accuracy, validity, and restricted access (CAVR). The application area identifies its key processes and sub-processes, and document existing controls [1].

The application team determines which controls need to be modified, implemented, or enhanced and then implement the changes. Further controls can be governed by ISO 27001:2005 & COBIT 4.0 frameworks, ISO & COBIT frameworks, and consideration is given to Technology Risk Baselines provided by the Financial Services Authority (FSA) and the Monetary Authority of Singapore (MAS). As a final step, the application teams are required to review and test documentation periodically to ensure that controls remain in place and are functioning as designed. Application teams implementing changes should understand as the technologies changes, controls will also need to updated and modified.

Today enterprises are advancing their technologies and in doing so validation checks are being incorporated. The following are some principles to be considered [4]:

1. Validate everything. Inspect what you expect, and reject anything unexpected.
2. Perform all validations on the server. Client side validation is suspect for reasons given earlier in this article.
3. Use positive filtering instead of negative filtering; check for what should be present instead of for what shouldn't. Possible filtering checks include:
  - a) Is it the right data type (string, integer, etc.)?
  - b) Is the parameter required?
  - c) Is the character set allowed?
  - d) Does the input meet minimum and maximum length constraints?
  - e) Is NULL allowed?

- f) If numeric, is the input within range constraints?
  - g) Does the input cause data duplication, and if so is it acceptable?
  - h) Does the input meet format requirements (e.g., when compared to a regular expression)?
  - i) If selected from a list, does the parameter contain a valid value?
4. Perform internal code reviews or “buddy checks”. The first line of defense is having programmers check each other’s work.

Quality Assurance checks must ensure processes cover testing not only valid inputs but also possible invalid inputs.

### 3. RISK MANAGEMENT STRATEGY

#### 3.1 Risk mitigation plan

Technology Risk Management can utilize the sample as found below in Table 1: Risk Management Strategy Sample to rank High Risk as 1-5 domains to be reviewed and risk strategies to be developed and implemented. Action plans are developed and deployed for each of the remaining domains 6-11. For each identified risk a response is associated to it. The possible responses are as follows [1]:

TABLE 1: RISK MANAGEMENT STRATEGY SAMPLE

Risk Rank	High-Risk Factors	Risk Management Activities	Risk Response
1	Technical Security Standards & Guidelines	Risk Reduction will be achieved by formalizing the publication, maintenance and deviation processes associated with these documents	Mitigate
2	Information Access Control	Ensure that access is appropriate based on roles and the separation of duties is adequate	Mitigate
3	Application Security & Availability	Ensure that applications undergo security analysis to minimize potential exposure	Mitigate
4	Change Management & Testing	Review data used for testing; ensure appropriate level of diligence based on sensitivity	Mitigate
5	Vulnerability Management	A need was identified to enhance the internal vulnerability assessment program	Mitigate
6	Information Security Data Classification	Sub-classification of non-public data is being reviewed. A task force has been formed to enhance asset classification standard(s)	Mitigate
7	Technology Asset Management	A study of the environment took place. E-discovery processes have been streamlined	Avoidance
8	Separation of Duties	An extensive technology project was undertaken. Applications were addressed according to exposure	Deferred
9	Network Security Management	Firewall rule changes are now reviewed by the technology risk team.	Transfer
10	Vulnerability Management (Internal)	Tools were brought in-house to evaluate potential vulnerabilities. Enterprise-wide deployment is scheduled	Deferred
11	Availability & Capacity Management	Multiple work streams are in place addressing system availability, change management & stability	Mitigate

1. Avoidance – Change the project to avoid the risk. Change scope, objectives, etc.

- 2. Transference – Shift the impact of a risk to a third party (like a subcontractor). It does not eliminate it, it simply shifts responsibility.
- 3. Mitigation – Take steps to reduce the probability and/or impact of a risk. Taking early action, close monitoring, more testing, etc.
- 4. Acceptance – Simply accept that this is a risk. When choosing acceptance as a response the IMPD is stating that given the probability of occurring and the associated impact to the project that results, they are not going to take any actions and will accept the cost, schedule, scope, and quality impacts if the risk event occurs.
- 5. Deferred – A determination of how to address this risk will be addressed at a later time.

#### 3.2 Importance of risk planning

Risk is the effect (positive or negative) of an event or series of events that take place in one or several locations. It is computed from the probability of the event becoming an issue and the impact it would have (Risk = Probability X Impact).

This becomes important for several reasons, according to Kobel & Gimpert [3], being in compliance with state and regulatory such as Sarbanes-Oxley (SOX), Gramm-Leach-Bliley Act (GLBA), Payment Card Industry (PCI) data security standard and others complicate the efforts of identifying the approach and developing the appropriate strategy to mitigate risk. Global organizations has to take

into consideration both domestic and international laws and regulatory requirements. When risks are considered an important element of an organization, the focus of good risk management becomes the identification and treatment of

*these risks. The objective of risk is to add maximum sustainable value to all the activities of the organization. Risk marshals the understanding of the potential ups and downs of all the factors which can affect the organization. It increases the probability of success, and reduces both the probability of failure and the uncertainty of achieving the organization's overall objectives.*

### 3.3 Change controls and disaster recovery procedures

Change controls or change management is a set of procedures or rules that ensure that changes to the hardware, software, application, and data on a system are authorized, scheduled, and tested. This process is set in place to ensure no unauthorized access take place. With specific guidelines in place, it will assist in preventing unnecessary system outages, which in turn protects the system's integrity. With adequate change management, application and hardware systems have greater stability, and new problems are easier to debug [5].

On the other hand, successful disaster recovery planning is a process designed and developed specifically to deal with catastrophic, large-scale interruptions in service to allow timely resumption of operations. These interruptions can be caused by disasters such as fire, flood, earthquakes, or malicious attacks. The basic assumption is that the building where the data center and computers reside may not be accessible, and that the operations need to resume elsewhere [5].

The relationship between adequate change control and successful disaster recovery is that they have to work together. With the change control remaining up-to-date at all times by involved parties; and in the event of a disaster the disaster recovery plan will coincide and mitigation will be lessened. To further build a direct relationship, planning scheduled outages regularly will ensure that all plans are in working order.

### 3.4 Email based security tools

Organizations today rely more on the email as a major form of communications, therefore protecting email from attacks has become an important and necessary activity. If left unprotected, email can potentially become the doorway for network viruses. Most must meet industry or governmental compliance regulations by managing, tracking, and archiving email [6]. Below is a list of available tools for consideration:

1. Email Integrity Suite (EIS) – contains four modules: PrivacyPost, PrivacyMobile, PrivacyLock and PrivacyVault.
2. Barracuda Message Archiver's Exchange stubbing feature, helping administrators to minimize a user's Exchange storage footprint. With the Exchange stubbing feature, administrators can increase storage efficiency by moving email attachments from the Exchange server to the Barracuda Message Archiver

while maintaining seamless access to those attachments for end users.

3. Symantec Mail Security for Domino provides real-time protection against viruses, spam, spyware, phishing, and other attacks while enforcing content policies for Lotus Domino email servers, documents, and databases.
4. Security Gateway email spam firewall for Exchange/SMTP Servers provides affordable emails security with a powerful spam filter that serves as a Microsoft Exchange firewall or SMTP firewall. It also protects against viruses, phishing, spoofing, and other forms of malware that present an ongoing threat to the legitimate email communications of your business.
5. Perimeter Email Protection (PEP) is a complete email protection service for businesses hosting their own on-premise email servers. It includes tools to filter and authenticate incoming email before it reaches your email infrastructure.

## 4. INCIDENT REPORTING

From an Information Security perspective, reporting an incident is highly important; the organization must have a plan in place that will set the direction of reporting and handling. This is important to mitigate the risk. Some important facts are to ensure the right people get the right information at the right time. Shareholder, Senior Management, Board of Directors, Business Units and individuals all require knowledge of the incident; however, each has different levels of information requirements. Receiving the correct information ensures knowledge of the significance of the risk to the organization and the potential effects. The following procedures are suggestive for the process of reporting incidents when service disruption has been affected:

1. Call to the application team hotline number using the appropriate phone number for normal business hours or after hours
2. Submit a Service-Now Incident (INC) ticket describing the issue/incident; select the appropriate level for escalation
3. INC status of "URGENT" alerts the Availability team
4. Availability team assess the issue for impact
5. Change Advisory Board (CAB) team is engaged to assess high-risk changes for Risk Level 5
6. A Service-Now Change Management Request (CRQ) is created by requestor to document incident
7. CRQ is submitted for approval – Risk Level 3-5 requires management approval. Risk Level 1-2 requires standard team leader approval.

8. CRQ risk level 4-5 requires Delta Con engagement to communicate and discuss higher-risk changes; this is a forum for technology and application representatives to review and discuss changes
9. Email communication is sent from ITSM Critical Incident to AVAIL-IM-BusinessPartner-Status@xxx.com;ITLeadership@xxx.com, DailyOperationalMeetingStatus@xxx.com, DailyApplicationMeetingStatus@xxx.com,TechnologyExtendedDRGroup@xxx.com, and the individual that opened the INC. The following should be communicated
  - a) Current Status
  - b) Next Steps
  - c) Next email update
  - d) Incident History (previous current status)
  - e) Contact groups/individual with Cell Phone

Organizations implementing Service-Now can take advantage of several integrated applications, including Change Management Request (CRQ), Incident Management (INC) and Knowledge Management (KM). This tool combines ITIL guidelines with WEB 2.0 technology. The reporting module offers reports for Executive Management and Management as well as operational reports. Reports are exportable into Excel or as a PDF file. Data is presented in a drill-down format. Report types include charts, lists, calendars and pivot tables which can be customized.

## 5. BUSINESS CONTINUITY

The goal of an organizations' business continuity plan is to recover mission critical business functions and related tier 1 systems within 12 hours or less following a disaster. Each department/division that has partners must have or be covered in a written BCP that has been reviewed and approved by applicable management (Division Manager or higher as defined by Human Resources). The business continuity plan can contain the following sections in the document:

1. Business Impact Analysis - Performs a Business Impact Analysis (BIA) focusing on business functions performed by the department/division. BIAs should be reviewed and updated, as necessary, every twelve months or in conjunction with BCP update schedule.
2. What To Do If A Disaster Strikes - Department/Divisions should document resumption procedures for at least the following five scenarios when developing their BCPs for disasters that occur during and after business hours.
  - a) Loss of Building - Scenario should document key meeting locations, alternate workspace locations, communication procedures for contacting members of the department/division

- and the strategy for how business function will be maintained/resumed.
- b) Loss of Mission Critical System(s) - For those computer applications used by a department/division that will not be available as soon as required by the business, the strategy for manual work-around, where possible, must be designed, implemented, tested, and documented in this scenario. Provisions for recreating, recovering, or replacing end-user developed computing tools.
  - c) Loss of Key Staff - Strategy for how work will be prioritized, how the business would function in the event of a significant reduction in staff availability. For Pandemic planning ensure the following key tasks are addressed in this section.
    - i. Address possible loss of 25-35% of key staff for prolonged periods of time and significant interruptions in services provided by essential vendors and providers.
    - ii. Determine and document essential business functions which must be sustained during a pandemic event.
    - iii. Document training plans and resiliency for key functions and staffing needs.
    - iv. Review the feasibility of alternate work options for essential business functions and note what options will be used during a pandemic event.
  - d) Loss of Key Service Provider(s) - Document strategy for how service would continue in the event an outside provider who typically performs this responsibility is not available.
  - e) Loss of (non-electronic) Vital Records, where applicable - Vital records are irreplaceable and provide evidence of legal status, ownership, financial status, etc. (e.g. contract with original signatures).
  - f) Decision Tree For When To Escalate The Recovery Process - When a problem arises, this section documents the process for evaluating the situation and determining what triggers are necessary to begin implementing actions documented in the BCP. Consider the following when documenting this process.
    - i. When to inform management of the problem.
    - ii. When to inform other partner areas of the problem.
    - iii. When to bring in other resources to help solve the problem.
    - iv. When to begin some of the recovery process as a hedge (e.g. changing

work priorities, implementing manual processes).

- g) Staff Call Tree - Include all available telephone numbers for staff in the department/division including home, work, and mobile. A call order should be established so it is clear who is responsible for calling who. A call completion step should be noted to confirm all members of the call tree received the message. Plans should provide for utilization of several types of communication methods such as telephone (land line and mobile), email, text messaging, etc.
  - i. Key Partner Telephone List - Document contact information for partners that the specific area would need to contact during or following a disaster.
  - ii. Vendors/External Parties/Government Agencies And Others Telephone Lists - Document contact information for anyone outside of the organization that the area would need to contact during or following a disaster.
  - iii. Where Area Will Be After A Disaster Strikes - Loss of Building procedures in the "What to Do If a Disaster Strikes" would include alternate workspace locations. List the locations where business functions will be restored, telephone numbers, fax numbers and machine name.
  - iv. Special Operating Procedures - Documentation in the "What To Do If A Disaster Strikes" section may be sufficient to note special operating procedures that would be followed in a disaster.
  - v. List Of Functions That Will Be Provided By Area During First 24 Hours - If work volumes or available resources prevent all business functions from being performed immediately following a disaster, this section should list what functions will be provided.
  - vi. Partner Areas That This Area Depends Upon - List department/divisions that this area depends upon in order to be able to perform its business functions; for example, if asset pricing is required to complete processing, include the department/divisions that provide these prices.
  - vii. Plan For Returning To Normal Operations - Once the disaster is over and normal operations are ready to continue; document how business-as-usual will be resumed.

## 6. TERMINATION PROCEDURES

To continue with mitigating risk, a solid termination process should be put in place. Termination is initiated by the direct manager communication to the Human Resource Consultant (HRC). HRC initiate a termination of system access removal request and submit an urgent email to the Access Control (AC) team to immediately disconnect access on Active Directory, Remote Access, Lotus Notes and all phone and credit cards. The termination access request by policy is to be completed that same day that it arrives in the AC team work QUEUE. Lastly, the termination request also kicks off equipment confiscation request for university owned laptop, blackberry phone, and remote Securid card.

## 7. PHYSICAL SITE SECURITY

The goal is to provide business continuity with minimal downtime. The facility area houses the data center, office area, energy center and common space. Transportation is available between the primary office and the remote data center. Remote data center provides real-time data transfer from the technology building and the data center. To mitigate phone service, another office provides telecommunication capabilities when the processing center experience loss from the central office phone switch. A few generators provide emergency electricity to the entire building. Uninterrupted Power Supply (UPS) allow for standard electrical feed to be cut over to the generator automatically as needed. State of art fire suppression system and enunciator panels are monitored continuously on-site and remotely from primary Security Console providing partner and computer protection. The security of the premises includes 24/7 guarded security and building engineer on staff. The building of this center had critical processes in mind that will protect the University's assets.

## 8. CONCLUSION

Online Universities are experiencing progressive IT security challenges to protect their valuable assets from insider/outsider attackers, natural disasters, and other sources. Therefore, building a solid Technology Risk Management team with the ability to manage risks around processes, tools, and procedures is imperative. Enterprise risk management focus should be on the preventing and detecting input errors with controls that verify completeness, accuracy, validity, and restricts access. Risk management strategies includes a well-thought-out risk management plan that identifies risks and address the risks response whether it is avoidance, transference, mitigation, acceptance to add maximum sustainable value to all the activities within the organization. This sustainable value is obtained with adequate change controls and successful disaster recovery procedures that work. This will make incident reporting, handling, and communication more effective in the event of a disaster ensuring the right people receive the right

information. Mitigating risks includes a working termination process which will address the removal of access that will potentially allow unauthorized access to the organization's network system. Lastly, it is crucial for medium to large size organization to build an off-site data/backup center that will enable them to avoid mission critical process going down and/or would help the organization up and running quickly.

#### REFERENCES

- [1]. Interoperability Montana Risk Management Plan (2007) retrieved November 17, 2010 from [http://interop.mt.gov/content/docs/IM\\_Risk\\_Management\\_Plan\\_v4\\_0.pdf](http://interop.mt.gov/content/docs/IM_Risk_Management_Plan_v4_0.pdf)
- [2]. Kobel, B., Gimpert, J. The importance of IT risk management in M&A (2008) retrieved November 22, 2010 from <http://cio.co.nz/cio.nsf/depth/1F0E5A42575B4D1CCC2574F5006DF8D7>
- [3]. Olzak, T. Web Application Security: Unvalidated Input (2006) Retrieved Nov 22, 2010 from [http://www.infosecwriters.com/text\\_resources/pdf/Unvalidated\\_Input\\_TOlzak.pdf](http://www.infosecwriters.com/text_resources/pdf/Unvalidated_Input_TOlzak.pdf)
- [4]. Oracle® Database High Availability Architecture and Best Practices 10g Release 1 (10.1) Part Number B10726-02 (2004) Retrieved Nov 22, 2010 from <http://www.stanford.edu/dept/itss/docs/oracle/10g/server.101/b10726/operbp.htm>
- [5]. Simonds, L. Keep your e-mail safe and stored (2006) Retrieved November 23, 2010 from <http://www.smallbusinesscomputing.com/biztools/article.php/3583551/Keep-Your-E-mail-Safe-and-Stored.htm>
- [6]. 2008 Annual Technology Risk Assessment (2009) Retrieved from internal resources from The Northern Trust Company
- [7]. Mullikin, Arwen and Rahman, Syed (Shawon); "The Ethical Dilemma of the USA Government Wiretapping"; International Journal of Managing Information Technology (IJMIT); ISSN : 0975-5586
- [8]. Rahman, Syed (Shawon) and Donahue, Shannon; "Convergence of Corporate and Information Security"; International Journal of Computer Science and Information Security, Vol. 7, No. 1, 2010; ISSN 1947-5500
- [9]. Bisong, Anthony and Rahman, Syed (Shawon); " An Overview of the Security Concerns in Enterprise Cloud Computing " ; International journal of Network Security & Its Applications (IJNSA)ISSN: 0975 - 2307
- [10]. A Risk Management Standard (2002) Retrieved Nov 22, 2010 from [http://www.theirm.org/publications/documents/Risk\\_Management\\_Standard\\_030820.pdf](http://www.theirm.org/publications/documents/Risk_Management_Standard_030820.pdf)
- [11]. How to Develop a Risk Management Plan (2010) Retrieved Nov 22, 2010 from <http://www.wikihow.com/Develop-a-Risk-Management-Plan>
- [12]. Zherui Hou, "Application of GB/T20984 in Electric Power Information Security Risk Assessment," icmtma, vol. 1, pp.616-619, 2010 International Conference on Measuring Technology and Mechatronics Automation, Changsha, China, 2010
- [13]. Vandana Gandotra, Archana Singhal, Punam Bedi; "Threat Mitigation, Monitoring and Management Plan - A New Approach in Risk Management " ARTCOM '09: Proceedings of the 2009 International Conference on Advances in Recent Technologies in Communication and Computing, October 2009
- [14]. Susan Dunnivant, M. Jean Childress; "A sideways approach to data security and privacy awareness" SIGUCCS '10: Proceedings of the 38th annual fall conference on SIGUCCS, October 2010

# Towards Self-Protecting Security for e-Health CDA Documents

G. Hsieh<sup>1</sup>

<sup>1</sup>Department of Computer Science, Norfolk State University, Norfolk, Virginia, USA

**Abstract** – *To protect the security and privacy of electronic medical records, it is often necessary to employ a variety of security mechanisms such as encryption, integrity control, authentication, and access control. This paper proposes a framework that extends HL7 Clinical Document Architecture (CDA) documents with markups from XML based security standards, including eXtensible Access Control Markup Language, XML Encryption, and XML Signature. This integrated structure uses a CDA document as the container while access control policies, digital signatures and encrypted data are all embedded within the same CDA document in a fine-grained manner. This approach can be used to provide self-protecting security for CDA documents no matter where they reside: in transit within HL7 messages or in existence as independent persistent information objects outside messages.*

**Keywords:** e-Health, HL7 CDA, XACML, XML Encryption, XML Signature.

## 1 Introduction

Clinical Document Architecture (CDA) is a document markup standard that specifies the structure and semantics of a clinical document [1], [2]. It is a Health Level Seven (HL7) [3] standard approved by American National Standards Institute (ANSI).

“A CDA document is a defined and complete information object that can include text, images, sounds, and other multimedia content. It can be transferred within a message and can exist independently, outside the transferring message. CDA documents are encoded in Extensible Markup Language (XML)” [2].

Since the release of its first version in 2000 and the second release in 2005, the HL7 CDA standard has received strong acceptance among the Electronic Medical Records (EMR) standardization, development, and user communities. For example, HL7 and ASTM International [4], formerly known as the American Society for Testing and Materials (ASTM), jointly created the Continuity of Care Document (CCD) to integrate ASTM's Continuity of Care Record (CCR) specification and HL7's CDA [5].

The CCD is an implementation guide for sharing CCR patient data using the HL7 CDA. It is an XML-based standard that specifies the structure and encoding of a patient summary clinical document. It provides a "snapshot in time," containing a summary of the pertinent clinical,

demographic, and administrative data for a specific patient [5].

The CDA/CCD documents can be used to exchange medical documents among health information systems [6]. They can also be used to store/exchange medical records for Portable or Personal Controlled Health Records (PHR) applications [7], [8].

To comply with national standards for protecting the privacy and security of individuals' electronic personal health information, such as those established by the Health Insurance Portability and Accountability Act (HIPAA) in the U.S., the CDA/CCD documents must be protected to ensure the confidentiality, integrity, and security of electronic protected health information [9].

It is very challenging to implement comprehensive, cost-effective, and user-friendly security mechanisms for EMR applications due to the stringent requirements on availability, safety, accessibility from multiple locations and devices, and the need for sharing information among potential participants while maintaining sufficient safeguards for confidentiality, integrity, authentication, and access control.

This paper proposes an integrated, secure, embedded and fine-grained access control framework that can be used to provide self-protecting security for CDA documents.

The fundamental concept underlying this framework is the utilization of a variety of open standards that are commonly used for web services security [10], e.g., eXensible Access Control Markup Language (XACML) [11], XML Encryption (XML-ENC) [12], XML Signature (XML-DSIG) [13], and XML Key Management Specification (XKMS) [14] to specify or represent access control policies, results of encryption, digital signatures, and key management information, respectively.

These standards are extended and used in an integrated manner such that the access control policies, encrypted data, digital signatures, and key management information can all be embedded within a CDA document. In addition, these security mechanisms for confidentiality, authentication, authorization, and integrity control can be applied in a fine-grained manner, i.e., different parts of a CDA document can be protected with different access control policies, cryptographic algorithms or keys.

Overall, this framework approach is designed to provide self-protecting security for the CDA documents throughout their lifecycles and no matter where they reside: in transit or at rest, within or across organizational boundaries.

The remainder of the paper is organized as follows. In the next section, we provide an overview of the security requirements for CDA documents. Section 3 presents a high level view of the proposed CDA self-protecting security framework. In Section 4, we describe a proposed prototype implementation of the framework. Section 5 discusses related works. In Section 6, we conclude the paper with a summary and discussion on future work.

## 2 Security Requirements

The ASTM E2369 - 05e1 Standard Specification for CCR outlines a set of security requirements and recommendations [15]:

- a) The data contained within the CCR are patient data and, if those are identifiable, then end-to-end CCR document integrity and confidentiality must be provided.
- b) Conditions of security and privacy for a CCR instance must be established in a way that allows only properly authenticated and authorized access to the CCR document instance or its elements.
- c) The CCR document instance must be self-protecting when possible, and carry sufficient data embedded in the document instance to permit access decisions to be made based upon confidentiality constraints or limitations specific to that instance.
- d) For profiles that require digital signatures, W3C's XML digital signature standard will be used with digital signatures. Encryption will be provided using W3C's XML encryption standard.

These security requirements and recommendations should be applied to the CDA/CCD documents as well.

J. Olsik also described in a white paper on Enterprise Rights Management (ERM) [16], "To overcome this problem of losing control, the information must be protected through persistent usage policies which remain with the information no matter where it goes. In other words, digital information should be extended so that it can carry persistent security policies that define and enforce the access and use of the information at all times regardless of disposition or location: creation, replication, transmission, consumption, modification, expiration, etc."

## 3 CDA Security Framework

The proposed CDA self-protecting security framework is designed to help meet the ASTM and ERM requirements and recommendations listed above. It is adapted from a previously developed framework that was designed to be general purpose for protecting digital information of any type [17]-[20].

### 3.1 CDA Document Structure

The structure of CDA documents is very suitable for the embedded and fine-grained approach. A CDA document is wrapped by the <ClinicalDocument> element which

can be used as the root element of the container for the security framework.

A CDA document contains a header and a body. The header identifies and classifies the document and provides information on the author, custodian, confidentiality status, the patient, and the involved providers for the document. The body contains the clinical report, and can be either an unstructured text, or can be comprised of structured markup.

A structured body is wrapped by the <StructuredBody> element, and can be divided into recursively nestable document sections. A CDA document section is wrapped by the <section> element. Each section can contain a single narrative block enclosed by the <text> element, and any number of CDA entries which can also nest.

All these CDA elements can be candidates for fine-grained embedding and security protection. For example, if the <ClinicalDocument> element is chosen as the target for applying the embedding and security mechanisms, then the entire CDA document is protected. On the other hand, if multiple sections are chosen as targets for applying such mechanisms, then these sections are protected with possibly different policies and/or cryptographic algorithms/keys.

Furthermore, the fine-grained approach is facilitated (and also necessitated) by the concept of CDA context which defines the applicable scope of the assertions in the document. CDA context is set in the CDA header and applies to the entire document, and it can be overridden at the level of the body, section, and/or CDA entry.

For example, the <confidentialityCode> element is a contextual component of CDA, and can be used to express the confidentiality rules (*normal*, *restricted access*, or *very restricted access*) for the part of the document covered by the scope of this element. Multiple <confidentialityCode> elements can be used in the same CDA document to express different levels of confidentiality restrictions for different parts of the document.

Figure 1 shows a simple sample CDA document containing two sections within the structured body.

```

<ClinicalDocument>
<!-- CDA Header -->
<StructuredBody>

  <section>
    <text>"Text1"
  </text>
  </section>

  <section>
    <text>"Text2"
  </text>
  </section>

</StructuredBody>
</ClinicalDocument>

```

Fig. 1. Sample CDA Document

### 3.2 Embedding XACML Policy

XACML [11] is an open standard established by the Organization for the Advancement of Structured Information Standards (OASIS). It specifies both an access control policy language and a request/response language. The policy language can be used to construct expressions that make up an access control policy that describes who can do what and when. The request/response language defines the format and values of elements that can be used to compose a request for access to a resource and to convey a response (authorization decision) granting or denying an access request.

XACML can be used for controlling access to any type of resources, not just XML documents. It supports an architectural model of separating the policy decision logic from the policy enforcement logic. Three key logical functions are defined by XACML. A *Policy Decision Point* (PDP) is an entity that evaluates applicable policy and renders an authorization decision. A *Policy Enforcement Point* (PEP) is an entity that performs access control by making decision requests to a PDP and enforcing the authorization decisions returned by the PDP. The third XACML logical function, *Policy Administration Point* (PAP), is an entity that creates and manages access control policies.

The base construct of all XACML policies is a `<Policy>` which represents a single access control policy, expressed through a set of `<Rule>`'s. Each XACML policy document contains exactly one Policy (or *PolicySet*) root XML tag. A policy can have any number of Rules which contain the core logic of an XACML policy. The decision logic of most rules is expressed in a `<Condition>`, which is a Boolean function. If the condition evaluates to true, then the Rule's *Effect* (Permit or Deny) is returned. Otherwise, the Condition does not apply. XACML also provides another feature called *Target* which is basically a set of simplified conditions that must be met for a Policy or Rule to apply to a given request.

To support embedding, the standard XACML policy and response languages are extended to allow a `<ResourceContent>` element, which is already defined for the standard XACML request language, within a `<Resource>` element. The original content to be protected by this XACML policy is first encoded into the base64 format and the result is then encapsulated within the `<ResourceContent>` element.

Figure 2 shows the sample CDA document with each of its two sections embedded with an XACML `<Policy>` element. Note that the content of each `<text>` element is replaced by an XACML `<Policy>` element which is used to express an XACML access control policy. The original content of the `<text>` element is embedded in an XACML `<ResourceContent>` element nested within the XACML `<Policy>` element.

```
<ClinicalDocument>
<!-- CDA Header -->
<StructuredBody>
```

```
<section>
<text>
  <Policy>
    <Rule>
      <Target>
        <Resource>
          <ResourceContent>base64("Text1")
        </ResourceContent>
        </Resource>
      </Target>
      <Condition>"PolicyRule1"</Condition>
    </Rule>
  </Policy>
</text>
</section>

<section>
<text>
  <Policy>
    <Rule>
      <Target>
        <Resource>
          <ResourceContent>base64("Text2")
        </ResourceContent>
        </Resource>
      </Target>
      <Condition>"PolicyRule2"</Condition>
    </Rule>
  </Policy>
</text>
</section>

</StructuredBody>
</ClinicalDocument>
```

Fig. 2. Sample CDA Document with Embedded XACML Policies

In summary, the extensions to the XACML languages and the conventions required to support the embedded and fine-grained approach are minor in nature, and thus they remain backward compatible with the standard XACML languages and processing models.

### 3.3 Applying XML-ENC and XML-DSIG

The XML Encryption Syntax and Processing standard [12], established by the World Wide Web Consortium (W3C), specifies a process for encrypting data and representing the result in XML. If the data is an XML element or XML element content, the result of encrypting data is an `<EncryptedData>` element, containing the ciphertext, which replaces the element or element content (respectively) in the encrypted version of the XML document.

The XML Signature Syntax and Processing standard [13], also established by the W3C, specifies XML syntax and processing rules for creating and representing digital signatures. XML signatures can be applied to any digital content, including XML. An XML digital signature is represented by a `<Signature>` element.

The XML-ENC and XML-DSIG mechanisms can be used together to encrypt and sign the CDA/XACML

document in support of the embedded and fine-grained approach. Figure 3 shows the sample CDA+XACML document whose two embedded `<ResourceContent>` elements are encrypted and then the entire `<ClinicalDocument>` element is signed with an enveloped XML signature. Note that the digital signature, XACML policy statements, and ciphertexts are all embedded in the same CDA document.

```

<ClinicalDocument>
  <!-- CDA Header -->
  <StructuredBody>

    <section>
      <text>
        <Policy>
          <Rule>
            <Target>
              <Resource>
                <ResourceContent>
                  <EncryptedData>
                    <EncryptionMethod/>
                    <KeyInfo>
                      <KeyName>Key1</KeyName>
                    </KeyInfo>
                    <CipherData>
                      <CipherValue>"Ciphertext1"
                    </CipherValue>
                    </CipherData>
                  </EncryptedData>
                </ResourceContent>
              </Resource>
            </Target>
            <Condition>"PolicyRule1"/<Condition>
          </Rule>
        </Policy>
      </text>
    </section>

    <section>
      <text>
        <Policy>
          <Rule>
            <Target>
              <Resource>
                <ResourceContent>
                  <EncryptedData>
                    <EncryptionMethod/>
                    <KeyInfo>
                      <KeyName>Key2</KeyName>
                    </KeyInfo>
                    <CipherData>
                      <CipherValue>"Ciphertext2"
                    </CipherValue>
                    </CipherData>
                  </EncryptedData>
                </ResourceContent>
              </Resource>
            </Target>
            <Condition>"PolicyRule2"</Condition>
          </Rule>
        </Policy>
      </text>
    </section>
  </StructuredBody>
</ClinicalDocument>

```

```

</StructuredBody>

<Signature>
  <SignedInfo>
    <CanonicalizationMethod/>
    <SignatureMethod/>
    <Reference>
      <DigestMethod/>
      <DigestValue/>
    </Reference>
  </SignedInfo>
  <SignatureValue>"DigitalSignature"
</SignatureValue>
<KeyInfo>
</KeyInfo>
</Signature>

</ClinicalDocument>

```

Fig. 3. Sample CDA Document Embedded with XACML Policies and XML Signature and XML Encryption

Note that the content of each `<ResourceContent>` element is replaced by an XML-ENC `<EncryptedData>` element which is used to encapsulate the encryption result and process information. A new XML-DSIG `<Signature>` element is inserted into the content of the CDA `<ClinicalDocument>` element using the enveloped signature mode.

### 3.4 Leveraging XKMS

The XML Key Management Specification [14], also published by W3C, is a specification of protocols for distributing and registering public keys, suitable for use in conjunction with the W3C Recommendations for XML Encryption and XML Signature.

The XKMS is designed to simplify the key management tasks for users and applications by allowing these tasks to be delegated to separate XKMS service providers. For example, by becoming a client of an XKMS service, an application is relieved of the complexity and syntax of the underlying Public Key Infrastructure (PKI) used to establish trust relationships which may be based on different specifications such as X.509/PKIX, SPKI or PGP.

The proposed framework leverages the XKMS by embedding within the CDA document the proper information needed for XKMS key management and usage, and supporting fine-grained encryption and signature operations.

### 3.5 Policy-Driven Security

The CDA self-protecting security framework can leverage certain information elements contained within the CDA documents, in consultation with the CDA implementation guides and templates, to implement policy-driven security.

For example, an implementation guide could specify that the CDA document should be digitally signed to safeguard its integrity and authenticity, and the information

contained within the <custodian> element, which represents the organization from which the document originates and that is in charge of maintaining the document, should be used to select the Digital Signature Algorithm (DSA) private key for signing the CDA document. The framework can then use these instructions to generate the appropriate digital signature and embed it within the CDA document.

Correspondingly, a recipient of the CDA document can use the same information contained within the <custodian> element to obtain the DSA public key for verifying the digital signature embedded within the received CDA document.

The use of implementation guides, templates, and profiles is a practical and attractive approach to develop, implement, and enforce policy-driven security in a user-friendly and efficient manner.

### 3.6 Benefits

The proposed framework and its associated security mechanisms are designed to provide the following benefits:

- The embedding capability facilitates the management, protection and enforcement of the content and associated access control policies, by having all information contained in a single data object and thus avoiding many of the problems commonly associated with managing these two types of information in separate entities. Furthermore, the access control policy is persistent with the content at all times and thus reducing the risk of losing control, especially after the information leaves the perimeter of an enterprise.
- The fine-grained access control capability facilitates the management and sharing of sensitive information at, for instance, multiple authorization levels, as it allows the use of a single version of the digital object to generate different “views” for different authorization levels without the need to maintain multiple, separately “redacted” versions of the content.
- The digital content and access control policies are secured with encryption and digital signature mechanisms, with additional key management support for leveraging the PKI.
- It leverages four open standards that are all very expressive, flexible, extensible, general-purpose, and widely deployed for real-world applications.

## 4 Framework Prototype Development

For feasibility study and experimentation purposes, we propose to develop a prototype software system for the CDA self-protecting security framework.

This prototype software system will be developed by leveraging the software base and development tools previously developed and used for the general-purpose framework prototype software system [17]-[19].

Figure 4 shows the high level architecture of the proposed CDA self-protecting security framework prototype

software system. The focus of the prototyping effort will be on the integration and processing of the CDA, XACML, XML-ENC, XML-DSIG, and XKMS documents.

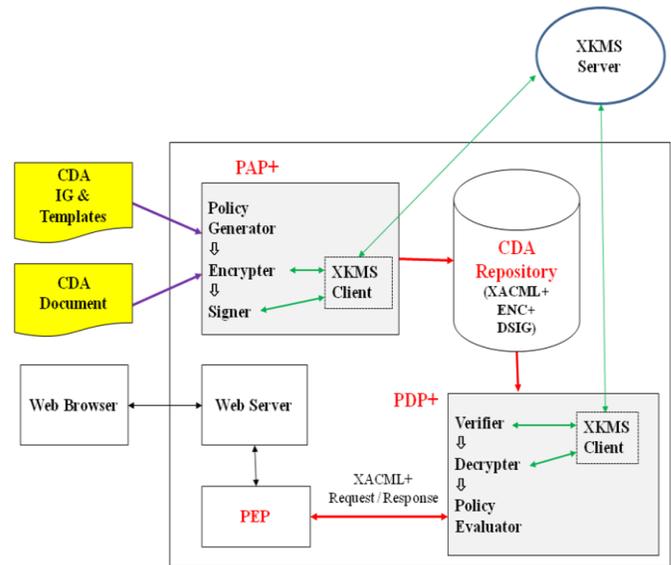


Fig. 4. CDA Framework Prototype Software System Architecture

The prototype software system will be implemented primarily in Java. It will leverage several open source software packages: XACML Java Library from Oracle Sun Labs [21], XML Security libraries from Apache Software Foundation [22], and Open XKMS [23] packages for XKMS and interworking with PKI, as those packages have been used for developing the prototype software system for the general-purpose framework.

The prototype PAP will perform three major functions. First, the PAP generates a CDA document which is embedded with appropriate XACML access control policy statements, using a source CDA document and applicable CDA implementation guides and templates as inputs. Second, it encrypts the selected sections of the CDA document. Third, the PAP signs the CDA document.

The prototype PDP will also perform three major functions. First, the PDP verifies the digital signature embedded in the CDA document. If the digital signature is valid, it then decrypts the ciphertext(s) embedded within the CDA document. If the decryption operation is successful, the PDP next evaluates the XACML access control policy, also embedded in the CDA document, and returns an authorization decision along with any authorized content.

## 5 Related Works

The policy embedding concept traces its root to the practice of applying security markings on documents within the security and intelligence community. For example, the U.S. Department of Defense DISA Net-Centric Enterprise Services Techguide [24] places special emphasis on standards and mechanisms that support XML and SOAP, such as XACML, XML-ENC, and XML-DSIG. It further specifies that data assets be tagged with security markings, i.e., using the Intelligence Community Information Security

Markings (IC-ISM) standard [25]. The framework's policy embedding approach is designed to take the tagging concept further by expressing the access control rules in a standard policy language such as XACML to facilitate wider adoption (and thus enforcement) of the practice.

The framework's policy embedding approach is similar to that of the Enterprise Rights Management (ERM) which is defined as a digital document-based security model that enforces access, usage, confidentiality, and storage policies [16]. However, the framework's method is based on XACML which is an open standard, while ERM is generally built upon Microsoft's proprietary Windows Rights Management Services (RMS) technology that works with RMS-enabled applications to help safeguard digital information from unauthorized use [26].

On the other hand, the framework's embedding approach is different from typical XACML applications which favor a strategy of separating the policies from the resources. This latter approach is very convenient for implementing a centralized entitlement management system, but it lacks sufficient support for protecting the information throughout its lifetime no matter where it resides. The Enterprise Policy Manager [27] from Cisco Systems, Inc. is a good example of a centralized entitlement management system for enterprises.

As noted in [28], the XML-based security standards would be used more and more in terms of an integrated security system, and the possible interaction of different standards was a basic goal in the evolution of XML-based security standards.

For example, the Telecare information platform described in [6] uses WS-Security mechanisms including XML-ENC and XML-DSIG for transmission security. However, it does not use them for end-to-end information-level security. It also does not support XACML, policy embedding, or fine-grained control.

The Portable CDA application for secure clinical-document exchange described in [7] uses XML-DSIG for digital signature operations to safeguard the integrity of the CDA documents. It uses encryption for confidentiality protection, without using XML-ENC. The system does not support XACML, policy embedding, or fine-grained control.

The peer-to-peer medical document exchange system described in [8] uses XACML for safeguarding the privacy of CDA documents. It uses encryption and digital signatures, without using XML-ENC or XML-DSIG. It relies on transport security, without support for policy embedding or fine-grained control.

As far as we know, the proposed framework is the first illustrated use of XACML, XML-ENC, XML-DSIG, and XKMS to provide self-protection security for CDA documents in an integrated, secure, embedded, and fine-grained manner.

## 6 Summary and Future Work

This paper proposes a CDA-based integrated secure embedded and fine-grained access control framework designed to help meet the stringent security and privacy requirements for protecting CDA documents: end-to-end integrity and confidentiality, authentication and authorization (access control), and self-protecting security with security policy and information embedded within the same CDA document.

This framework leverages XML-based security standards such as XACML, XML Encryption, XML Signature, and XKMS. Thus, it benefits from the rich expressiveness, flexibility, extensibility, and general-purpose applicability of all these open standards. In addition, this framework supports fine-grained security protection which is necessary for protecting CDA documents such that sensitive information in certain parts of the documents can be accessed only by properly authenticated and authorized entities.

Going forward, we plan to continue refining the detailed level description of the proposed CDA self-protecting security framework, developing the proposed prototype software system for the framework, and enhancing the integration and interoperability with CDA infrastructures such as software, tools, implementation guides and templates.

To facilitate the development and deployment of such a framework, especially in cloud computing or healthcare exchange environments, we plan to explore advanced cryptographic and key management schemes, such as hierarchical identity-based encryption and secret key sharing schemes, that can be leveraged to enhance the flexibility, scalability, efficiency, and ease of use of the framework.

## 7 References

- [1] R. H. Dolin, L. Alschuler, C. Beebe, P. V. Boyer, D. Essin, E. Kimber, et al. "The HL7 Clinical Document Architecture, Release 2," in *J. Am Med Inform Assoc.*, vol. 13(1), Jan.-Feb. 2006, pp. 30–39.
- [2] R. H. Dolin, L. Alschuler, C. Beebe, P. V. Boyer, D. Essin, E. Kimber, et al. "The HL7 Clinical Document Architecture," in *J. Am Med Inform Assoc.*, vol. 8(6), Nov.-Dec. 2001, pp. 552–569.
- [3] Health Level Seven International. <http://www.hl7.org>.
- [4] ASTM International. <http://www.astm.org/index.shtml>.
- [5] Product CCD – HL7Wiki. <http://wiki.hl7.org/index.php?title=ProductCCD>.
- [6] S.-H. Li, C.-Y. Wang, W.-H. Lu, Y.-Y. Lin, and D. C. Yen. "Design and Implementation of a Telecare Information Platform," in *J. Med. Syst.* Published online: Dec 2010.

- [7] K.-H. Huang, S.-H. Hsieh, Y.-J. Chang, F. Lai, S.-L. Hsieh, and H.-H. Lee. "Application of Portable CDA for Secure Clinical-document Exchange," in *J. Med. Syst.*, vol. 34, 2010, pp. 531-539.
- [8] J. H. Weber-Jahnke, and C. Obry. "Protecting privacy during peer-to-peer exchange of medical documents," in *Inf. Syst. Front.* Published online: Apr 2011.
- [9] HIPAA Security Rule. <http://www.hhs.gov/ocr/privacy/hipaa/administrative/securityrule/index.html>.
- [10] E. Bertino, L. D. Martino, F. Paci, and A. C. Squicciarini. *Security for Web Services and Service-Oriented Architectures*. Springer-Verlag, 2010.
- [11] eXtensible Access Control Markup Language (XACML) Version 2.0. OASIS Standard Specification. 1 Feb 2006. [http://docs.oasis-open.org/xacml/2.0/access\\_control-xacml-2.0-core-spec-os.pdf](http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf).
- [12] XML Encryption Syntax and Processing. W3C Recommendation. 10 Dec 2002. <http://www.w3.org/TR/xmlenc-core/>.
- [13] XML Signature Syntax and Processing (Second Edition). W3C Recommendation. 10 June 2008. <http://www.w3.org/TR/xmldsig-core/>.
- [14] XML Key Management Specification (XKMS 2.0). Version 2.0. W3C Recommendation. 28 June 2005. <http://www.w3.org/TR/xkms2/>.
- [15] ASTM E2369 – 05e1 Standard Specification for Continuity of Care Record (CCR). <http://www.astm.org/Standards/E2369.htm>.
- [16] J. Oltsik. *Enterprise Rights Management: A Superior Approach to Confidential Data Security*. White paper. Enterprise Strategy Group. 2006. [http://ahca.myflorida.com/dhit/Board/erm\\_a\\_superior\\_approach\\_to\\_confidential\\_data\\_security\\_final\\_\\_2\\_\\_050206.pdf](http://ahca.myflorida.com/dhit/Board/erm_a_superior_approach_to_confidential_data_security_final__2__050206.pdf).
- [17] G. Hsieh, and M. Masiane. "Towards an Integrated Embedded Fine-Grained Information Protection Framework," in *Proc. 2011 Intl. Conf. on Information Science and Applications*, Korea, 2011.
- [18] G. Hsieh, R. Meeks, and L. Marvel. "Supporting Secure Embedded Access Control Policy with XACML+XML Security," in *Proc. 5th Int. Conf. on Future Information Technology*, Korea, 2010, pp.1-6.
- [19] G. Hsieh, K. Foster, G. Emamali, G. Patrick, and L. Marvel. "Using XACML for Embedded and Fine-Grained Access Control Policy," in *Proc. 4<sup>th</sup> Int. Conf. on Availability, Reliability and Security*, Japan, 2009, pp. 462-468.
- [20] G. Hsieh, G. Patrick, K. Foster, G. Emamali, and L. Marvel. "Integrated mandatory access control for digital data," in *Proc. SPIE 2008 Defense + Security Conf.*, Florida, 2008, vol. 6973, pp. 697302-1 to 697302-10.
- [21] Sun's XACML Implementation. <http://sunxacml.sourceforge.net/>.
- [22] Apache Santuario – The Java Section. <http://santuario.apache.org/Java/index.html>.
- [23] Open XKMS. <http://sourceforge.net/projects/xkms/>.
- [24] Information Assurance/Security – Techguide. DOD DISA Net-Centric Enterprise Services. <http://metadata.dod.mil/mdr/ns/ces/techguide/security.html>.
- [25] Intelligence Community Information Security Marking (IC ISM) XML Schema. <http://www.niem.gov/IC-ISMv2.xsd>.
- [26] Microsoft Active Directory Rights Management Services. <http://www.microsoft.com/windowsserver2008/en/us/ad-rmsoverview.aspx>.
- [27] Cisco Policy Management. [http://www.cisco.com/en/US/products/ps9519/Products\\_Sub\\_Category\\_Home.htm](http://www.cisco.com/en/US/products/ps9519/Products_Sub_Category_Home.htm).
- [28] A. Ekelhart, S. Fenz, G. Goluch, M. Steinkellner, and E. Weippl. "XML Security - A Comparative Literature Review," in *Systems and Software*, vol. 81(10), pp. 1715-1724, Oct 2008.

# Analysis of security requirements in telemedicine networks

Edward Guillen, Paola Estupiñan, Camilo Lemus, Leonardo Ramirez

Telecommunications Engineering Department. GISSIC Investigation Group  
Nueva Granada Military University  
Bogotá D.C., Colombia  
{edward.guillen, gissic}@unimilitar.edu.co

**Abstract**— *Telemedicine networks' privacy and security are the most important issues for patient's medical information maintenance, access and transmission. Possible threatens or attacks to the systems such as not authorized logged in, data changes or destruction can be avoided considering the worth of these two aspects. Any weakness in any part of the system can affect the entire system. In fact, it is necessary to establish the security requirements and mechanisms that keep the data integrity by analyzing the standards that control the system. The intention of this paper is to examine the kind of services delivered by a telemedicine network and to propose a cluster of minimum security requirements that must be taken into account in aspects such as access, data transmission, human resources, network devices and medical diagnostic equipment based on the international standards HIPAA, COBIT, CALDICOTT, ITU-T and ISO.*

**Key words:** Telemedicine requirements, HIPAA, COBIT, CALDICOTT, security.

## I. INTRODUCTION

Telemedicine was firstly proposed in earlier 1970 and its function was limited to offer consultation medical services. [1] Nowadays, it is considered as a new medical assistance way characterized by delivering distant health services that helps to improve patients' attention. For instance, it is possible to give a faster diagnose and generate medical data registers for each patient. This also makes possible that the system exchanges data to prevent and control the non- authorized staff access.

Telemedicine is a system that provides medical care assistance where the medical staff remotely examines to distant patients through communication and information technologies. This is one of the strong bases that help in reducing costs and patients conglomeration at health entities [2]. Informatics and telecommunication usage in the health area must, in fact, accomplish the correct security measures in order to guarantee the medical information's confidentiality, availability and integrity and indeed to offer protection to the patient, the medical staff and to the general human resource.

This paper classifies the most common services and specialties in telemedicine by evaluating its usage statistics and analyzing the requirements in order to keep security and integrity of the

patient's medical information. According to standards such as HIPAA, COBIT and CALDICOT, it is possible to protect the important aspects in a Health system in order to maintain the security on the patient's medical information is private and it should be protected of non-authorized persona to avoid obtaining personal advantages and fraud. Finally the importance of this analysis is defined in the conclusions. [3]

## II. TELEMEDICINE NETWORK KEY FEATURES

### I. Telemedicine Services

Telemedicine provide medical information and services when distance separates the participants through information technologies which allow exchanging data in order to examine, diagnose and treat a patient. Telemedicine makes easier to access to all communities regardless the geographic area.

Telemedicine services can be classified in different ways, there is a classification for data, audio or images transmission [4], there is other classification according to the sort of applications or specialties services, finally the last classification is based on sophisticated technologies that can be depicted according to the sent information type and the used media to transmit the data. [5]

Along time, the number of services and specialties increase supplying different health needs, these applications growth day by day given different services telemedicine. [6] [7] [8].

### B. International Standards for Telemedicine security

For a correct medical information security management, it is necessary to create controls and procedures that assure precautionary measures to maintain the medical information security, confidentiality, integrity and availability according to the standard security pattern of three levels known as CIA which pursues to minimize the risks of the confidential information and to establish security policies. [9]

Although there are some rules that allow accomplishing the CIA security pattern for the patient's medical data, such as HIPAA, COBIT, CALDICOTT, ISO and ITU-T.

HIPAA (Health Insurance Portability and Accountability Act), is defined as a group of standards that guarantee the medical information safety in aspects as transmission, storage and access to the protected health information (PHI). It specifies

general management requirements through the 45 CFR Part 160 – General Administrative Requirements, 45 CFR Part 164 –security and privacy, Basics of Security Risk Analysis and Risk Management. [10]

The COBIT rule (Control Objectives for Information Systems and related Technology), is a group of the best practices for security, quality, efficiency and efficacy in the information technology; these practices are necessary to identify risks, to manage resources and to measure the performance that allow to achieve the society objectives. An evaluation of the security measures of high and medium level was made by the document given by COBIT version 4.1 [11]

The Health department, in its organization CALDICOTT establishes parameters that allow the personal data protection in the health services as the responsibility related to the entire staff; this allows having high quality and protected against improper divulging information. The requirements to maintain the patient's medical data integrity rule are compiled by the given documents (The confidentiality, security and sharing of personal data-policy and procedures for the local health Report on the Review of Patient-Identifiable Information). [12]

The standards ISO 27000 are designed for the data security where specifies the necessary aspects to introduce an ideal Management System of Information Security, general definitions, good practices guidelines that describe the advisable controls and the requirements to the audit entities accreditation and certification of the Management System of Information Security. [13]

The ITU-T standards guarantee the network's operations compatibility and efficacy, the help to protect the telecommunication facilities and services, in addition they strength the Next Generation Networks, that allow the security requirements review to be applied in the telemedicine networks in the recommendation X.805 y X1051.[14]

### C. Telemedicine Network

Telemedicine is other level of the communication and information technologies usage; this includes a real diagnosis, exams and a distance patient's treatment. [15] A telemedicine network usually has low and high velocity links communication. The low velocity links are used for the communication between the medical staff, patients and the Health care facility, on the other hand, the high velocity links help to establish the communication between the health care entities. [16]

The personal in charge registers the vital signs, takes pictures or does a videoconference in a telemedicine network from a far-off location. This information is sent through a network (3G, GSM, Internet, etc.), which is kept in a server and a specialist accesses and makes a data analysis. The specialist personal finally makes a diagnosis in order that the personal in charged does the correct procedures or the patient takes the appropriate decisions (Figure1). [17], [18]

The telemedicine should take into account that additionally to the data transmission, there are aspects as:

- Medical data compilation (laboratory analysis, X-Rays or other medical pictures, and real time following up)
- Data compression, storage and data recovery
- It is important to implement a security mechanism in order to maintain the personal and medical information.

These tasks give support and complete the telemedicine principal function; however they require mechanisms and tools that support functions as: qualified personal, complexes information systems, and networks.

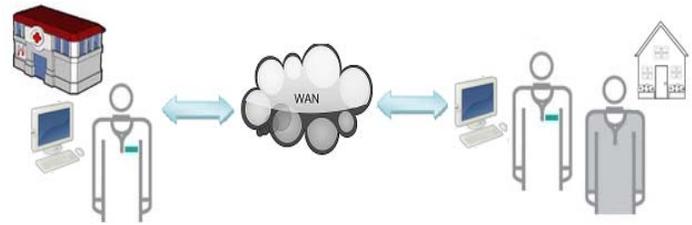


Figure 1. Telemedicine Network Basic

### III. TELEMEDICINE SERVICES

Telemedicine's requirements varies according to the applications given by the system, this is the reason It is important to know the most often offered services in a telemedicine network (Figure 2), and the types of services and specialties according to Pan American Health Organization (PAHO) (Figure 3), knowing these characteristics it is possible to classify and allocate the requirements to a telemedicine. [19]

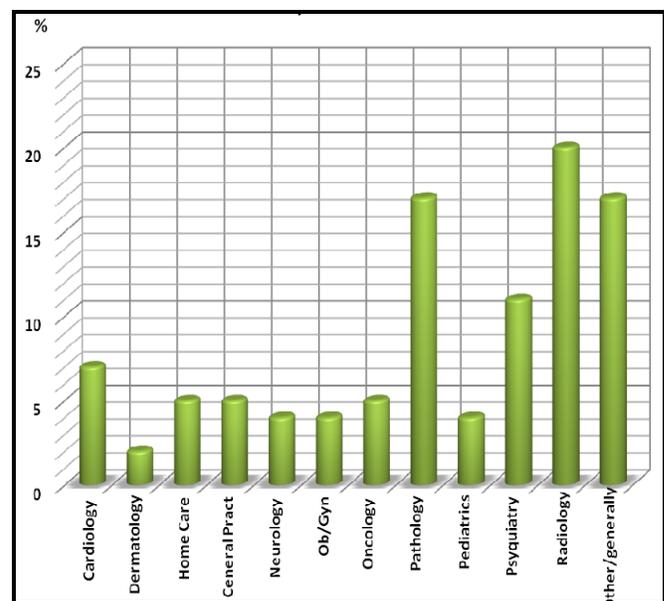


Figure 2. Use telemedicine applications. [20]

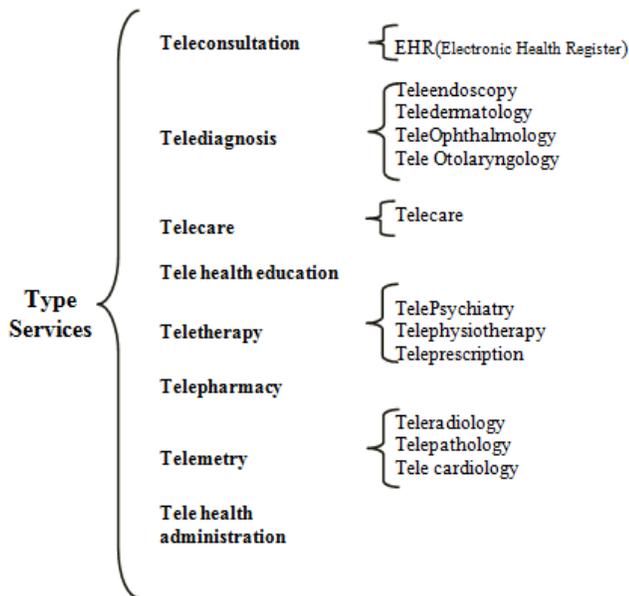


Figure 3. Telemedicine Services Classification

IV. POSSIBLE VULNERABILITIES TELEMEDICINE NETWORK.

In order to establish precautionary and protectionary mechanisms for the network medical information, it must identify weakness, and faults named vulnerabilities that can be used to attack and violate a system and its data. [21]. Because of that security and interoperability are considered important requirements in a telemedicine system, due that medical data is exchanged pretending the protection of the data and the prevention of the non-authorized usage, mean while a level of high accessibility is maintained, although the interoperability is important to establish security agreements for exchanging data between health entities.

It is important to analyze all web systems failures, and evaluate them from the telemedicine network point of view, subsequently; it is possible to know the standards offered by the supervisión and mechanisms to prevent network vulnerabilities. This is caused by the vulnerabilities that affect aspects such as storage, access, and data transmission, these vulnerabilities disrupt the networks equipment used for medical diagnosis and for human resources, these vulnerabilities are [22] [23]. The most common vulnerabilities are shown in the next Figure 4.

After the vulnerabilities are determined, mechanisms appear to maintain the security telemedicine network, these mechanisms condense in different standards such as HIPAA, COBIT, CALDICOTT, ITU-T and ISO, these allow to every country to have resource to any standard, these regulations detailly expose the requirements for storage, access, data transmission, network hardware (network equipment Network equipment, diagnostic medical equipment and human

resources). It is possible to determine which the requirements for each service thanks to the standards; the table 1 shows the services qualification and the requirements to do a relationship between them.

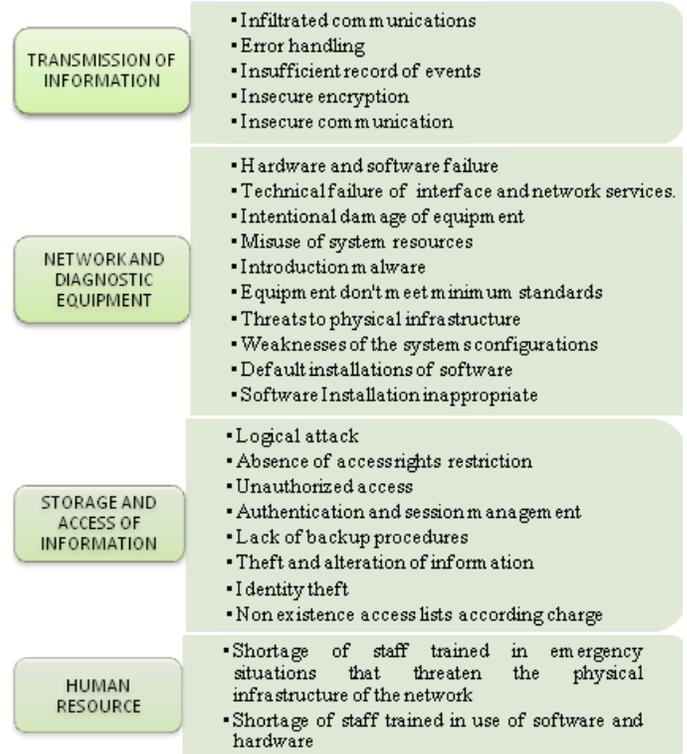


Figure 4. Vulnerability Classifications.

V. ANALYSIS NETWORK TELEMEDICINE REQUIERIMENTS

After looking at the different telemedicine services that can be implemented in a health care entity and after doing a research of the vulnerabilities in a network, the requirements are classified according to their sort as it is shown in Figure 5, the applied requirements are evaluating (Table2, 3).

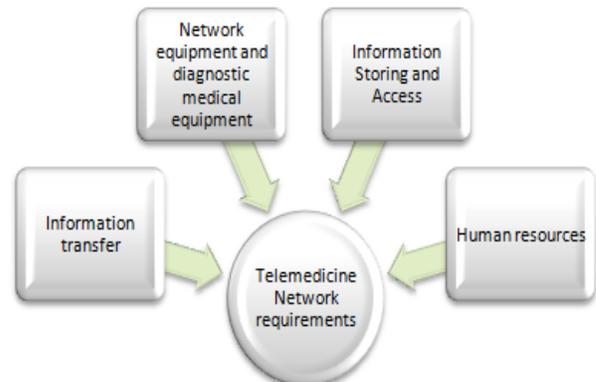


Figure5. Telemedicine Network requirements

- Type 1 Storing and access Information
- Type 2: Information transfer
- Type 3: Human Resource
- Type 4: Network equipment and diagnostic medical equipment

**Table 1 . Storing, Access and transmission of information Requirements**

Type	ID	Features
1	r1	Implement procedures to verify that a person or entity seeking access to electronic protected health information is the one claimed.
	r2	Assign a unique name and/or number for identifying and user identity
	r3	Implement procedures to limit access to health information system
	r4	Information technology and appropriately designed clinical information systems have to permit the collection and analysis of considerable amounts of information about patients in computerized databases so that patients' conditions and progress can be checked and evaluated, as well as supporting the further development of an evidence base.
	r5	Implement some biometric identifiers, including finger, voice prints or iris.
	r6	Implement electronic mechanisms to corroborate that electronic protected health information has not been altered or destroyed in an unauthorized manner.
2	r7	Implement network protocols to ensure data sent and received are equal and verify its integrity.
	r8	Implement a mechanism to encrypt and decrypt electronic health information
	r9	Implement hardware, software, and/or procedural mechanisms that record and examine activity in information systems that contain or use electronic protected health information.
	r10	Establishes a communication protocol basis for information sharing both within the local Health entity, and between no-local health entities.

The previous table is a requirements compilation to control the transmission and storage medical information because the telemedicine networks should consider and adapt to requisites that allow the interoperability and a good performance. On the other hand, there are requirements for the staff that manages the medical data in a health entity; in the next table some requirements are mentioned.

While Internet develops, the mobility and supply of new services are strengthen the telemedicine services such as the medical assistance, but, all these services must watch over the information security through privacy regulations as HIPAA. [24]

**Table 2. Human Resource requirements**

Type	ID	Features
3	r11	The workforce must be accredited to offer their services as required by regulations for each profession.
	r12	Implementing policies and procedures to ensure that all staff members have appropriate access to electronic health information.
	r13	Establishing and implementing policies and procedures for responding to an emergency or other occurrence that damages systems that contain electronic health information.
	r14	The health entity must certify that they have personnel to handle the technology used in telemedicine procedures.
	r15	Implementing policies and procedures to ensure that all staff members have appropriate access to electronic protected health information and know explicitly the obligations and the consequences of breaches of confidentiality
	r16	Everyone should know what they must do to keep secure and confidential information, being aware that health information cannot be disclosed outside their immediate working environment.

Finally, it is established requirements that the hardware network should consider medical diagnostic and data storage.

**Table 3. Common requirements for network equipment**

Type	ID	Features
4	r17	The Capture peripheral that comes in contact with the patient must meet the minimum requirements to safeguard the physical integrity of this.
	r18	The video signals, these images are usually analog and they require to be digitized
	r19	Information systems require remote connections.
	r20	The network equipments and diagnostic medical equipment require bandwidth and sophisticated user interfaces
	r21	The health entity must have electronic storage media including memory devices in computers (hard drives) and any removable/transportable digital memory medium, such as optical disk, or digital memory card, and computers with internet access to certain features of hard drive capacity, RAM and processor
	r22	The health entity must have inventories of network equipment and diagnostic medical equipment and update them constantly.
	r23	The configuration of network equipment and diagnostic medical equipment must meet the minimum standards to ensure the proper interpretation of the information by the receiver.

The telemedicine services needed the storage requirements, Access, data transmission, human resource and hardware can be defined according to every telemedicine service and requirements exposed on the studied standards.

## VI. CONCLUSIONS

There is a constant growth in the development of Telemedicine applications and because of them it is possible to offer some advantages such as: Minimizing the visiting number emergencies, reducing the necessary inputs and the tracking visits done by the patients. To achieve these advantages the entities that assistance this sort of services should know the mechanisms for the medical data protection regulated by each country and the international standards.

Telemedicine has moved to a point where it is recognized as an important and practical aspect, that is why it is important to use the requirements proposed for security, assuring issues as confidentiality, integrity and availability of the medical information.

## REFERENCES

- [1] G. H. Zhang, Carmen C. Y. Poon, Member, IEEE, Ye. Li, and Y. T. Zhang. A Biometric Method to Secure Telemedicine Systems. 31st Annual International Conference of the IEEE EMBS Minneapolis, Minnesota, USA, September 2-6, 2009
- [2] Jader Wallauer, Aldo von Wangenheim, Rafael Andrade, Douglas D. J. de Macedo, A Telemedicine Network Using Secure Techniques and Intelligent User Access Control, 21st IEEE International Symposium on Computer-Based Medical Systems p. 1-3.
- [3] G. H. Zhang, Carmen C. Y. Poon, Member, IEEE, Ye. Li, and Y. T. Zhang. A Biometric Method to Secure Telemedicine Systems. 31st Annual International Conference of the IEEE EMBS Minneapolis, Minnesota, USA, September 2-6, 2009
- [4] Toninelli Alessandra , Montanari Rebecca, Corradi Antonio, " Enabling secure service discovery in mobile healthcare enterprise networks". University of bologna. IEEE Wireless Communications Magazine. June 2009

- [5] Sicurello Francesco, "Some Aspects on Telemedicine and Health Network", Referent of Italian Ministry of Foreign Affairs for Telemedicine, CNR- Institute of Biomedical Advanced Technology Milan
- [6] International Telecommunication Union, "UIT-T International Telecommunication Union – Telecommunication Sector", March 1993.
- [7] Sicurello Francesco, "Some Aspects on Telemedicine and Health Network", Referent of Italian Ministry of Foreign Affairs for Telemedicine, CNR- Institute of Biomedical Advanced Technology Milan.
- [8] Consejo Nacional de Seguridad Social en Salud, " Resolución número 1448 de 2006", Ministerio de Salud, República de Colombia, 2006.
- [9] International Telecommunication Union, "UIT-T International Telecommunication Union – Telecommunication Sector", March 1993
- [10] Health Insurance Portability and Accountability Act (HIPAA), "Reassessing Your Security Practices in a Health IT Environment: A Guide for Small Health Care Practices", 1996.
- [11] Brand, Koen & Boonen, Harry, "IT Governance based on COBIT 4.0: a management guide", Van Haren Publishing, 2004.
- [12] Caldicott, Dame Fiona "report on the review of patient-identifiable information", Department on Health and British Medical association, December 1997
- [13] International Organization for Standardization ISO and International Electro technical Commission IEC, "International Standard ISO/IEC 27000:2009", First edition, 2009.
- [14] International Telecommunication Union, "UIT-T International Telecommunication Union – Telecommunication Sector", March 1993.
- [15] International Telecommunication Union, "UIT-T International Telecommunication Union – Telecommunication Sector", March 1993.
- [16] Bingyi Hu, Jing Bai,, Datian Ye. An internet based communication server for telemedicine. The Department of Electrical Engineering, The School of Life Science and Engineering Tsinghua University, Beijing, 100084, P.R.China. Proceedings - 19th International Conference - IEEE/EMBS Oct. 30 - Nov. 2, 1997 Chicago, IL. USA
- [17] Shaikh, Asadullah y Misbahuddin, Muhammad. Título: "A system design for a tele-medicine health care system" Göteborg (Suiza), Tesis de Maestría en Ingeniería de Software y Gestión. IT University of Goteborg. Department of Applied information Technology, 2007.
- [18] Zvikhachevskaya, Anna, Markarian, Garik, Mihaylova, Lyudmila, "Quality of Service consideration for the wireless telemedicine and e-health services", WCNC 2009 proceedings, IEEE, 2009
- [19] Organización Panamericana de la Salud OPS/OMS, ORAS-CONHU Organismo Andino de Salud, "Aplicaciones de Telecomunicaciones en la salud en la Subregión Andina", Serie Documentos Institucionales. 2000.
- [20] Walid G. Tohmet, Silas Olsson. Developments and Directions of Technology Assessment in Telemedicine. Department of Radiology, Georgetown University Medical Center, Washington, DC 20007. Swedish Institute for Health Development (Spri), Stockholm, Sweden.
- [21] International Telecommunication Union, "UIT-T International Telecommunication Union – Telecommunication Sector", March 1993.
- [22] Amiya K. Maji, Arpita Mukhoty, Arun K. Majumdar, Jayanta Mukhopadhyay, Shamik Sural, Soubhik Paul, Bandana Majumdar. Security Analysis and Implementation of Web-based Telemedicine Services with a Four-tier Architecture. IEEE. Indian Institute of Technology Kharagpur, India
- [23] Ilias Maglogiannis Elias Zafiroopoulos, Modeling Risk in Distributed Healthcare Information Systems. Proceedings of the 28th IEEE EMBS Annual International Conference New York City, USA, Aug 30-Sept 3.
- [24] Frank E. Ferrante, MSEE, MSEPP. Maintaining Security and Privacy of Patient Information Proceedings of the 28th IEEE EMBS Annual International Conference New York City, USA, Aug 30-Sept 3, 2006

# The Knowledge Based Authentication Attacks

Farnaz Towhidi<sup>1</sup>, Azizah Abdul Manaf<sup>1</sup>, Salwani Mohd Daud<sup>1</sup>, Arash Habibi Lashkari<sup>1</sup>

<sup>1</sup>Advanced Informatics School, Universiti Teknologi Malaysia (UTM), Kuala Lumpur, Kuala Lumpur, Malaysia

**Abstract** - Knowledge Based authentication is still the most widely used and accepted technique for securing resources from unauthorized access for its simplicity, ease of revocation and legacy deployment which divides to textual and graphical password. Over the last decade several attacks records for stealing user's identity and confidential information using a single or combination of attacks. In this paper the attacks pattern of textual and graphical password describes according to CAPEC standard, following describing their effects on both conventional and image password. More over some categories lacks from detail research which highlighted and will select as future work.

**Keywords:** Authentication Attack, Graphical Password Attacks, Knowledge Based Attacks, Recognition Based Attacks, Recall based Attacks.

## 1 Introduction

The tradeoff between security and usability of knowledge based authentication schemes was always a topic for experts. Although textual password is still use as an entrance in most secure environments, graphical password has been selected as an alternative due to the drawbacks of textual password. The image password is classified into three categories: Recognition Based; Pure Recall Based and Cued Recall Based [1].

These recognition categories provide a gallery of faces, objects, random arts or icons for users and some of them can be selected as the user's password. In pure recall based or draw based [2], the user needs to draw a shape or signature in an empty grid as password. For the other category, cued recall based or click based, the system provides user special facilities to remember his password, for instance in Passpoint algorithm, the user is given a chance to select some points in the background image as his passwords. [2-4].

The Common Attack Pattern Enumeration Classification's (CAPEC) Release 1.6, defines and describes the common attack pattern along with the observation of the Department of Homeland Security, which will be used to help users find the subset pattern of enumeration. In this paper, the attack pattern of knowledge based authentication are as follow:

## 2 Password Brute Force

Also known as exhaustive search or guessing attack, it uses "Probabilistic Techniques" as a method of attack. Probabilistic technique shows that when one object is selected randomly from a class of objects, the probability that the result would be the same as the desired result is more than zero, so password brute forcing works by an attacker who uses trial and error method to explore all possible passwords of the user [5].

In textual password brute forcing, the attacker should trying millions of passwords by testing every combination of letters, digits, special symbols and punctuation symbol until a password is found. So, an attacker will require years, and in some cases hundreds or thousands of years, to completely reveal a password. In the recognition category of graphical password, when the attacker physically accesses the database of pictures, this attack is possible using the trial and error using stolen images. In draw based, an attack can be launched using smart programs to automatically generate accurate mouse motion to imitate human [2].

To prevent successful attacks, password management policies can force user to change his password before a brute force attack managed to check all possible combinations. This attack has two different versions which use brute force as a method for attacking, namely "Dictionary Based Password Attack" and "Rainbow Table Password Attack" with the following description [5].

### 2.1 Dictionary Based Password Attack

This is one of the branches of brute force attack which the attacker creates a dictionary of textual or graphical possible password, and then tries to compromise an account with one user name and the passwords in the created dictionary [5].

In textual password, the attacker creates a dictionary of memorable words like dates of birth as passwords. On the other hand, there are also several free wordlists or software tools that automate dictionary attacks [6]. In click based graphical password, the attacker creates a dictionary of popular spots of image or points which can attract the user. There are several algorithms which show the visual attention of user like "Bottom up" and "Top down". In Bottom up, the human attention goes towards "hotspots" or "salient" which are recognizable shapes, bright colors, or objects that are more likely to be selected [7-9]. In top down visual search,

humans control his attention by searching for a specific thing in a picture [9-10]. When the dictionary is created, attacker uses software to check the login page for one username and hundreds of password in the dictionary.

To prevent this attack, a mechanism named CAPTCHA needs to identify whether the password is entered by the user or by an automated program. The Captcha is a challenge response program that generates a test to identify whether the third party is a human or a system. This test can be solved easily by the user but hard to bypass by a system [11].

## 2.2 Rainbow Table Password Attack

Nowadays administrators try to save the password of users in hash form. If the attackers want to find the plain text of hash value, he has two choices. One is to calculate the hash value of many plain texts to find the same hash which might take a long time. Or the other choice is pre computing the hash of billions of passwords and store them in a Rainbow table in order to find the correct password [12]. These tables take a very long time and uses large space to generate, but once the attacker has the tables, it facilitates attacks by cracking a large number of passwords in a second. The main idea of the rainbow table is using chain of hashing and reduction function. A hash function maps the plaintexts to hashes, and the reduction function reduces the length of hash function to a fix value. The chain of rainbow table starts with a plaintext and finishes with a hash value.

To prevent the risk of Rainbow table, the administrators adds a random character named salt before hashing. The salt value is stored in the database for each user. During every authentication, a new challenge is generated by the server, the Rainbow tables need to either include all the salt combinations which would make them unmanageably large, or recalculate the table every time which makes them similar in terms of efficiency to brute force attacks [12]. In this situation, the attacker needs to find the correct salt for each of his hashing which makes the process much too long. If the selected salt key is long enough, compromising the password would be much harder for the attacker [12-13].

## 3 Sniffing Attack

Sniffing uses the weakness found in design of application to reveal more information to the intruder than what it intends to show by using sniffer to monitor and eavesdropping the input or output data [5, 14-15]. In textual password, the user starts to send "Towhidi7958F" as his password which transferred in sequence of packets. The packets go up and down through network along with the packet's destination address to show which computer is permitted to accept it. At the same time the attacker uses a sniffer software to change the configuration of his Network Interface Card to a promiscuous mode to collect all these packets [14]. In Recognition based Graphical Password, the attacker can sniff the ID of image password. This ID is usable

only if user can attack the image gallery at the same time. No research has found the impact of sniffing attacks on click based and draw based algorithms.

To protect from password sniffing attacks, sensitive information must be properly encrypted [16], another choice is using "IP Security Protocol (IPsec)" that secure data in the network layer by authenticating and encrypting each packet in communication [17].

## 4 Spoofing Attack

The attacker uses various techniques like Action Spoofing, Content Spoofing, or Identity Spoofing to masquerade his message as a legitimate one in order to trick user. In "Action Spoofing" the attacker changes the mechanism of actions to lead victim to a wrong way. For instance the user thinks by clicking on the return button of the page, he will redirect to the home page but in return, an executable file is run by attacker. In "Content Spoofing", the attacker changes the contents of one page to show his messages rather than the original one. For instance in the bank portal, the attacker change the account number of bank to his. In the "Identity Spoofing", the attacker impersonates a legitimate user.

A computer may be protected from this attack by restricting the IP addresses that sends data. A router may have a list of IP numbers and it allows only data from these numbers to enter the computer. So if the attacker gains the IP list, he can start sending data that appears to come from a legitimate IP address. But when the attacker do not have the list, he starts to send packets with consecutive IP numbers until a packet gain one of allowed IP in the list, which in this case the packet gains access to computer [15]. Identity Spoofing have several subsets like "Man in the Middle Attack" and "Phishing Attack" as follow [5].

### 4.1 Man in the Middle Attack

Derive from basketball, when two player want to pass a ball to each other, another player interrupts the passing ball without prior knowledge [18]. In the man in the middle attack (MITM), the intruder uses spoofing method by sitting somewhere between the client and the server and starts sniffing packets or even alter message from first party and send the changed message to the second, so although the two parties thought they are directly talking to each other, the attackers actually control all the conversation without any sign [5, 19]. In case of successful MITM, the attacker can have several consequences like DNS poisoning, denial of service attack or even Https sniffing. In case of sniffing, any textual or graphical password can be observed by the intruder, especially when data transfer in TCP protocol as data are transferred without any encryption [18].

To protect password against such attacks, hashing password, multi factor authentication, digital certificate, channel encryption, and integrity protection is recommended [20].

## 4.2 Phishing Attack

Phishing is the act of stealing a user's confidential information by pretending to be a legitimate entity. For example, an attacker design a fake website exactly like a bank's portal, then starts to send out a spam e-mails to a large random number of users trying to convinces the user to visit the cracked website and enter their account number and password. The method of convincing user can done by social engineering techniques like telephone call, SMS, and so on. [20-22].

In graphical password under the click based category, phishing is possible by creating a faked login page and simulating the area for drawing password. Once the user draws his password, the sketch can be used in the legitimate website. During recognition, the username is retrieved by the faked website, and then it will be passed to the legitimate website for retrieving the correct image gallery. Again this gallery is shown to the user in the faked website which causes the user to select the password [2]. Even click based can be simulate in phishing website, for instance the attacker can include the background images in the login page, when the user starts to click on special point of picture as his password, the area of password is revealed to the attacker.

Using "List Based" technique is recommended for mitigating this attack which divides to black list and white list. The black list contains the list of all phishing website which gathered by web crawlers or list maintainers but these list are helpful only if their data is accurate and fresh. The white list on the other hand includes the list of all trusted domains. So by visiting any website that does not record in white list, an alert message will be shown to the user [22].

## 5 Exploitation of Authentication

In the case that user does not have username and passwords; there are several methods for exploiting authentication like "Authentication Bypass" and "Exploitation of Session Variables, Resource IDs and other Trusted Credentials". The description of these two password attacks are as a follows:

### 5.1 Authentication Bypass

In this attack the attacker can gain access to resources, application, service and credential information with the privilege of an authorized person [5]. For instance an intruder can firstly gain physical access to local computer and then change the setting to administrator privilege. This privilege bypass all authentication for accessing files and folders [6]. In web application, most financial systems have authentication page as the heart for entering secure transactions. In some cases, the attacker bypasses the authentication page and

directly type the URL of the page which will show after authentication.

### 5.2 Exploitation of Session Variables, Resource IDs

"Session Side Jacking" is a sort of Exploitation of Authentication by exploiting session variable and resource ID [5]. This attack controls the communication channel of two endpoints. When the user establishes a new session and authenticate successfully, the attacker can assume the identity of this user, and then install a Trojan horse on the target's computer to watch the activity and records user names and textual or graphical password. If the data transfer is in a "Secure Socket Layer", the attacker cannot intercept the data, otherwise the data can be captured and then "Session Replay" by the attacker [15].

In "Session Replay" or "Authentication Replay" the user steals a valid session ID or password and reused it again to gain privilege. Authentication replay is mostly used when the user uses encrypted password. For instance, suppose one client usually sends an encrypted string to the server that provides for user authentication, this encrypted string can be captured by the attacker and presented to the server by the attacker. Authentication schemes that use static authentication parameters are susceptible to password replay attacks.

Many authentication protocols uses a challenge-response mechanisms for user authentication like when the authenticator generates a random string and present it as a challenge to the supplicant. The supplicant will typically manipulate the server challenge in some way and will typically encrypt it using the user password or a derivative as the key, or generate a hash based on the challenge and the user password. In any case, the user plaintext password will be used by the client to generate a response to the server challenge. If an attacker manages to capture a challenge-response authentication session, he may be able to see the encrypted supplicant response that depend on the server-provided challenge.

## 6 Social Engineering Attacks

This is one of the oldest attacks that simply involves psychological and technical methods of tricking the user into believing that he needs to provide his confidential information.

Text based password can be easily described or written down on a paper or even describe verbally, so social engineering can easily lunch in this sort of authentication [2]. For graphical password, describing password by telephone or email is harder than conventional passwords because verbalizing a click points in a picture or even drawing shape is very hard. For instance in Passpoint scheme, explaining the exact click point of a user is very hard since there are millions of available spots in a picture. A research on Passface

graphical password shows although the algorithm is vulnerable to description attack, a wise choice of decoy pictures can increase or decrease its vulnerabilities [23].

Countermeasure of this attack is very hard because it does not relate to any bugs or weakness in the system. Since the weakest link in any security system is humans, using users awareness training and security policies and procedures is highly recommended [24-25].

## 7 Physical Security Attacks

When an attacker has physical access to a computer, there is a chance of bypassing authentication and easily get access to resources even without authenticating. In case of physical attack of textual or even graphical password, an attacker can steal the password database from the server and launch offline attacks against it. For instance in the recognition category, the image password of the user is stored in the database, so anyone who gain access to the password bank can retrieve the credential information [5].

Although preventing security attack is hard, cryptography may be the solution to mitigate the risks of such attacks by encrypting password in database using a key. The key should be stored in a different computer to prevent an attacker from accessing information.

## 8 Shoulder Surfing Attack

The attacker tries to use direct observation like looking over the user's shoulder, using binoculars or even closed-circuit television cameras for capturing user's credential. For instance, this attack happens when user try to enter his password using the keyboard, mouse or even touch screen [26]. Graphical password schemes are more vulnerable to shoulder surfing than textual passwords [27]. In the recognition category, some of the algorithms design a challenge response method to resist this attack which forces the user to not clicking directly on password. For instance, the triangle algorithm , CDS Algorithm [28], DWT algorithm [29] and color login algorithm [30]. In the click based category, the attacker needs to capture exactly the position clicked by mouse in the image. Also in the draw based scheme, the process of drawing password is entirely visible to the attacker to memorize or even record [31]

To prevent this attack, users should make sure that no one is looking behind them when they are typing their passwords or even shielding the keypad from view by using their body or cupping their hands. [6].

## 9 Conclusion

Nowadays security of authentication remains an issue of paramount importance to verify the identity of person or process. Among various methods of authentication like biometric, smartcards and password authentication, textual

and graphical password have been used for centuries and still remain the most popular mechanism. This paper reviews the common attacks of knowledge base authentication and the reflection in textual and graphical password.

On the other hand the limitation of human memory on memorizing strong and secure textual password led to focusing more on the security of graphical password. In the future the attacks on graphical password will describe in detail.

## 10 Acknowledgement

This paper is supported by project UTM-J-13-01/25.10/3/02H07(1) from Research University Grant (RUG) of University Technology Malaysia (UTM).

## 11 References

- [1] Farmand, S. and O.B. Zakaria, Impro Passwving Graphicalord Resistant to Shoulder-Surfing Using 4-way Recognition-Based Sequence Reproduction (RBSR4), in The 2nd IEEE International Conference on Information Management and Engineering. 2010: Chengdu, China.
- [2] Biddle, R., S. Chiasson, and P.C.v. Oorschot, Graphical Passwords: Learning from the First Generation. 2009: Ottawa, Canada.
- [3] Towhidi, F. and M. Masrom, A Survey on Recognition-Based Graphical User Authentication Algorithms. International Journal of Computer Science and Information Security (IJCSIS) 2009. 6(2).
- [4] Masrom, M., F. Towhidi, and A.H. Lashkari, Pure and Cued Recall-Based Graphical User Authentication, in The 3rd International Conference on Application of Information and Communication Technologies (AICT2009). 2009: Azerbaijan, Baku.
- [5] CAPEC-Release1.6, Common attack Pattern Enumeration and Classification: <http://capec.mitre.org>.
- [6] Todorov, D., Mechanics of User Identification and Authentication. 2007: Taylor & Francis Group.
- [7] Oorschot, P.C.v., A. Salehi-Abari, and J. Thorpe, Purely Automated Attacks on PassPoints-Style Graphical Passwords. IEEE Transactions on Information Forensics and Security, 2010. 5(3): p. 393 - 405
- [8] Thorpe, J. and P.C.v. Oorschot. Human-Seeded Attacks and Exploiting Hot-Spots in Graphical Passwords. in 16th USENIX Security Symposium on USENIX Security. 2007. CA, USA.

- [9] LeBlanc, D., et al., Can Eye Gaze Reveal Graphical Passwords?, in ACM Symposium on Usable Privacy and Security (SOUPS). 2008, ACM: Pittsburgh, USA.
- [10] Henry, P.T., Toward usable, robust memometric authentication: An evaluation of selected password generation assistance, in College of Information. 2007, Florida State University. p. 207.
- [11] Wang, L., et al., Against Spyware Using CAPTCHA in Graphical Password Scheme. 2010.
- [12] Thorpe, J., *On the predictability and security of user choice in password*, in *Computer Science*. 2008, CARLETON UNIVERSITY: Ottawa, Ontario. p. 197.
- [13] S. H. Khayal, et al., *Analysis of Password Login Phishing Based Protocols for Security Improvements*, in *International Conference on Emerging Technologies, 2009. ICET 2009*. . 2009, IEEE. p. 368 - 371.
- [14] Qadeer, M.A., et al., *Bottleneck analysis and traffic congestion avoidance*, in *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*. 2010, ACM: Mumbai, Maharashtra, India. p. 273-278.
- [15] Salomon, D., *Network Security*, in *Elements of Computer Security*. 2010, Springer London. p. 179-208.
- [16] Spangler, R., *Packet Sniffer Detection with AntiSniff*. 2003.
- [17] Yin, H. and H. Wang, *Building an application-aware IPsec policy system*. IEEE/ACM Trans. Netw., 2007. **15**(6): p. 1502-1513.
- [18] Nayak, i.N. and S.G. Samaddar, *Different flavours of Man-In-The-Middle attack, consequences and feasible solutions*, in *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)* 2010, IEEE: Chengdu p. 491 - 495
- [19] YUAN, X., et al., *Visualization Tools for Teaching Computer Security*. 2010, ACM.
- [20] Joshi, Y., D. Das, and S. Saha, *Mitigating Man in the Middle Attack over Secure Sockets Layer*. 2009.
- [21] Butler, R., *A framework of anti-phishing measures aimed at protecting the online consumer's identity*. 2007.
- [22] Huang, C.-Y., S. PinMaa, and K. TaChen, *Using one-time passwords to prevent password phishing attacks*. JournalbofbNetworkbandbComputer Applications, 2010.
- [23] Dunphy, P., J. Nicholson, and P. Olivier, *Securing Passfaces for Description*. 2008.
- [24] Sandouka, H., A. Cullen, and I. Mann, *Social Engineering Detection using Neural Networks*, in *2009 International Conference on CyberWorlds*. 2009, IEEE.
- [25] al, A.A.G.e., *Network Attacks*, in *Network Intrusion Detection and Prevention: Concepts and Techniques*, Springer Science, Business Media.
- [26] Kumar, M., et al., *Reducing Shoulder-surfing by Using Gaze-based Password Entry*, in *Symposium On Usable Privacy and Security (SOUPS)*. 2007: Pittsburgh, PA, USA.
- [27] Tari, F., A.A. Ozok, and S.H. Holden, *A Comparison of Perceived and Real Shoulder-surfing Risks between Alphanumeric and Graphical Passwords*, in *Symposium On Usable Privacy and Security (SOUPS)*. 2006: Pittsburgh, PA, USA.
- [28] Gao, H., et al., *A New Graphical Password Scheme Resistant to Shoulder-Surfing*, in *International Conference on Cyberworlds*. 2010, IEEE: Singapore p. 194 - 199
- [29] Hasegawa, M., Y. Tanaka, and S. Kato, *A Study on an Image Synthesis Method for Graphical Passwords*, in *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2009)*. 2009.
- [30] Gao, H., et al., *Analysis and Evaluation of the ColorLogin Graphical Password Scheme*, in *Fifth International Conference on Image and Graphics (ICIG)*. 2009, IEEE. p. 722 - 727
- [31] Lashkari, A.H., et al., *Shoulder Surfing attack in graphical password authentication*. International Journal of Computer Science and Information Security, 2009. 6(9).

## User Authentication Platform using Provisioning in Cloud Computing Environment

Hyosik Ahn, Hyokyung Chang, Changbok Jang, Euiin Choi\*

Dept. Of Computer Engineering, Hannam University, Daejeon, Korea  
{hsahn, hkjang, chbjang}@dmlab.hannam.ac.kr, eichoi@hnu.kr

**Abstract.** Cloud computing is a computing environment centered on users and can use programs or documents stored respectively in servers by operating an applied software such as Web browser through diverse devices on where users can access Internet as an on-demand outsourcing service of IT resources using Internet. In order to use these cloud computing service, a user authentication is needed. This paper proposed the platform for user authentication using provisioning in Cloud computing environment.

**Keywords:** Cloud Computing, User Authentication, Provisioning, Security

### 1 Introduction

Gartner announced Top 10 of IT strategy technologies of the year of 2010 in 2009, October. Strategy technologies Gartner mentioned are the technologies which importantly affect enterprises for the next 3 years and have a powerful effect on IT and business. They may affect long-term plans, programs, and major projects of enterprises and help enterprises get strategic advantages if enterprises adopt them a head start. Cloud Computing got the top rank (2nd rank in 2009)[1][2]. Cloud Computing is a model of performance business and also infrastructure management methodology. It lets users use hardware, software and network resources as much as possible so as to provide innovative services through Web in these business performance models and also enables to provision a server according to needs of the logical by using automated advanced tools[3][4].

Also, Cloud Computing models offer both users and IT managers user interface which helps manage provisioned resources easy through the entire life cycle of a service request[5][6]. When resources requested by the user arrives through Cloud, the user can track the order consisted of a certain number of servers and software and operate jobs such as checking the state of the resources provided, adding a server, changing the installed software, removing a server, increasing or decreasing allocated processing power and memory or storage, even starting, aborting, and restarting server[7].

---

\* Corresponding Author

However, the diffusion of cloud computing incites users' desires for more improved, faster and more secure service delivery. Hence, security issues in the Cloud Computing environment are constantly emerging and authentication and access control has been studying. A user in the Cloud Computing environment has to complete the personal authentication process required by the service provider whenever using a new Cloud service. In this process, in the case that the characteristic and safety have been invaded by any attack, there will be a severe damage because personal information stored in the database and business processing service have been exposed or information related to individuals or organizations will be exposed as well.

Therefore, this paper designs a platform for user authentication using provisioning in the Cloud Computing environment.

## 2 Related Works

### 2.1 Definition of Cloud computing and the Appearance Background

Cloud Computing is a kind of on-demand computing method that lets users use IT resources such as network, server, storage, service, application, and so on via Internet when needing them rather than owning them[5]. Cloud Computing can be considered as a sum of SaaS (Software as a Service) and utility computing and Figure 1 shows the roles of users or providers in the Cloud Computing under the concept[9].

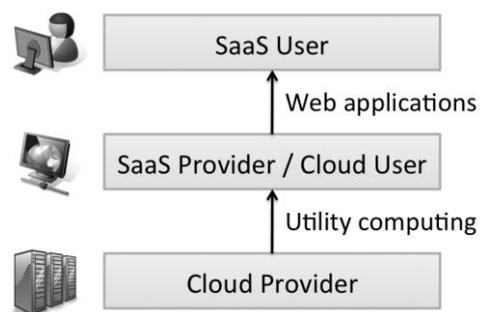


Figure 1. Users and Providers of Cloud Computing

### 2.2 Provisioning

Provisioning is a procedure and behavior that prepares required knowledge in advance and provides by requests in order to find the best thing. That is, it allocates, deploys,

and distributes infrastructure resources to meet the needs of users or business and helps use the resources in the system[9].

- Server Resources Provisioning: prepares resources like the CPU of the server and memory by allocating and placing the resources appropriately
- OS Provisioning: prepares OS operative by installing OS in the server and configuring tasks
- Software Provisioning: prepares to operate by installing/distributing Software(WAS, DBMS, Application, etc.) in the system and doing configuration settings needed
- Storage Provisioning: enables to identify wasted or unused storage and put it into a common pool. If there is a request of storage since then, the administrator gets one out of the pool and uses. Possible to construct infrastructure to heighten the efficiency of storage
- Account Provisioning: the process that HR manager and IT manager take the appropriate approval process and then generate accounts needed for various applications such as e-mail. groupware, ERP, etc. or change the access authorities when the category of the resources that users access changes[10]

Figure 2 shows such a provisioning service.

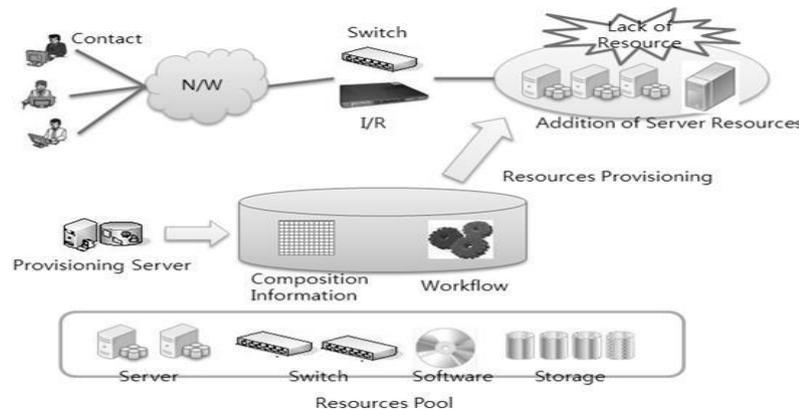


Figure 2. Provisioning Service

### 2.3 Security Technology in the Cloud computing Environment

There are no concrete security technologies in Cloud Computing, however, if we regard Cloud Computing as an extension of the existing IT technologies, it is possible to divide some of them by each component of Cloud Computing and apply to[12]. Access control and user authentication are representative as security technologies used for platforms. Access control is the technology that controls a process in the operating system not to approach the area of another process. There are DAC

(Discretionary Access Control), MAC (Media Access Control), and RBAC (Role-Based Access Control). DAC helps a user establish the access authority to the resources that he/she owns as he/she pleases. MAC establishes the vertical/horizontal access rules in the system at the standard of security level and area for the resources and uses them. RBAC gives an access authority to a user group based on the role the user group plays in the organization, not to a user. RBAC is widely used because it is fit for the commercial organizations. Technologies used to authenticate a user are Id/password, Public Key Infrastructure, multi-factor authentication, SSO (Single Sign On), MTM (Mobile Trusted Module), and i-Pin[11].

### **3 User Authentication in the Cloud computing Environment**

#### **3.1 User Authentication Technology in the Cloud computing Environment**

Users in the Cloud Computing environment have to complete the user authentication process required by the service provider whenever they use new Cloud service. Generally a user registers with offering personal information and a service provider provides a user's own ID (identification) and an authentication method for user authentication after a registration is done. Then the user uses the ID and the authentication method to operate the user authentication when the user accesses to use a Cloud Computing service[13]. Unfortunately, there is a possibility that the characteristics and safety of authentication method can be invaded by an attack during the process of authentication, and then it could cause severe damages. Hence, there must be not only security but also interoperability for user authentication of Cloud Computing.

#### **3.2 Weakness of User Authentication Technology in Cloud computing**

The representative user authentication security technologies described above have some weaknesses[11][12].

- Id/password: the representative user authentication method. It is simple and easy to use, but it has to have a certain level of complication and regular renewal to keep the security.
- PKI(Public Key Infrastructure): an authentication means using a public-key cryptography. It enables to authenticate the other party based on the certificate without shared secret information. In PKI structure, it is impossible to manage and inspect the process of client side.
- Multi-factor: a method to heighten the security intensity by combining a few means of authentication. Id, password, biometrics like fingerprint and iris, certificate, OTP (One Time Pad), etc. are used. OTP information as well as Id/password can be disclosed to an attacker.

- SSO (Single Sign On): a kind of passport if it gets authentication from a site, then it can go through to other sites with assertion and has no authentication process. The representative standard of assertion is SAML.
- MTM (Mobile Trusted Module): a hardware-based security module. It is a proposed standard by TCG (Trusted Computing Group) which Nokia, Samsung, France Telecom, Ericson, etc. take part in. It is mainly applied to authenticate terminals from telecommunications, however, it is being considered as a Cloud Computing authentication method with SIM (Subscriber Identity Module) because of generalization of smartphone[15].
- i-Pin: a technique to use to confirm a user's identification when the user uses Internet in Korea now. It operates in a way that the organization which itself performed the identification of the user issues the assertion.

#### 4 Platform Design for user authentication using Provisioning in Cloud Computing Environment

##### 4.1 User Authentication using Provisioning

User authentication platform using provisioning first authenticates by using ID/Password, PKI, SSO, etc. which a user input. Second, it authenticates with Authentication Manager through the user profile and patterns and stores the changes of the state via Monitor. When using Cloud Computing services, to solve the inconvenience of user authentication, user's information is stored in the User Information. Figure 3 shows a conceptual architecture.

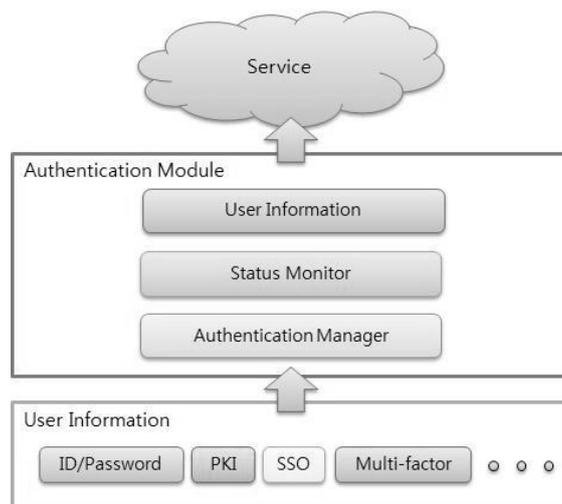


Figure 3. Conceptual Architecture

For user authentication, Authentication Module has the previous user profile and log data record by Provisioning strategy.

#### 4.2 Design of User Authentication Platform Using Provisioning

Figure 4 shows the composition of User authentication platform using provisioning proposed in this paper.

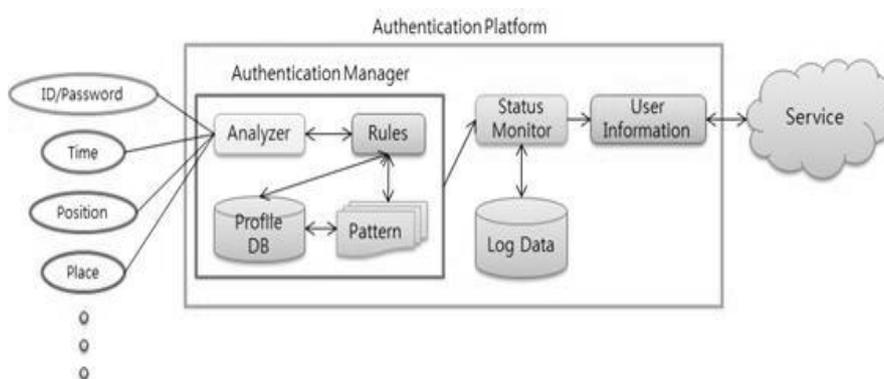


Figure 4. Authentication Platform

To authenticate a user first, Analyzer analyzes information such as ID/Password, Time, Position, Place, etc. it analyzes user pattern by using Rules for user authentication. To analyze user pattern, it authenticates a user by analyzing with current status and Rules of Profile DB. Profile DB stores user profile like existing user login time, location, position, etc. Also, it monitors the changes of user's status and situations via Status Monitor, records them in the database, and puts user information into User Information. That is how the user authentication process which the service provider asks to users whenever they use Cloud Computing services.

## 5 Conclusions

In this paper, user authentication platform using Provisioning in Cloud Computing environment was proposed and the features of this proposal are as follows.

There is some troublesome for users to get user authentication in the existing Cloud Computing environment because they have to go through the user authentication process to use the service every single time by using an ID and authentication method that the service provider provided. So, user authentication platform using Provisioning solves the existing inconvenience and helps use Cloud Computing services easily.

The proposed platform architecture analyzes user information and authenticates a user through user profile. Also, it has user information stored through user monitoring and there is an advantage that the user authentication process required by the service provider can be omitted when using other Cloud Computing services.

As further study, there should be a study on protection of user information, which is profile and log data in the platform proposed by this paper.

**Acknowledgments.** This work was supported by the Security Engineering Research Center, granted by the Korea Ministry of Knowledge Economy

## References

1. Lee, T.: Features of Cloud Computing and Offered Service State by Individual Vender. In: Broadcasting and Communication Policy, vol. 22, pp. 18—22 (2000)
2. <http://www.gartner.com/technology/symposium/2009/sym19/about.jsp>
3. Lee, J., Choi, D.: New trend of IT led by Cloud Computing. In: LG Economic Research Institute, (2010)
4. Lee, H., Chung, M.: Context-Aware Security for Cloud Computing Environment. In: IEEK, vol. 47, pp. 561—568 (2010)
5. Kim, J., Kim, H.: Cloud Computing Industry Trend and Introduction Effect. In: IT Insight, National IT Industry promotion Agency (2010)
6. [http://en.wikipedia.org/wiki/Cloud\\_computing#History](http://en.wikipedia.org/wiki/Cloud_computing#History)
7. Dikaiakos, M.D., et al.: Cloud Computing Distributed Internet Computing for IT and Scientific Research. In: IEEEInternet Computing, pp. 10-13, September/October (2009)
8. Harauz, J., et al.: Data Security in the World of Cloud Computing. In: IEEE Security & Privacy, pp. 61-64 (2009)
9. Lee, J.: Cloud Compting, Changes IT Industry Paradigm. In: LG Business Insight, pp. 40-46 (2009)
10. Kim, H., Park, C.: Cloud Computing and Personal Authentication Service. In: KIISC, vol. 20, pp. 11-19 (2010)
11. Armbust, M., et al.: Above the Clouds: A Berkeley View of Cloud Computing. In: Technical Report. <http://www.eeec.berkeley.edu/Pubs/TechRpts/2009/EEEC-2009-28.html> (2009)
12. Un, S., et al.: Cloud Computing Security Technology. In: ETRI, vol. 24, no. 4, pp. 79-88 (2009)

# Use of 2D Codes and Mobile Technology for Monitoring of Machines in Manufacturing Systems

Boleslaw Fabisiak<sup>1</sup>

<sup>1</sup>West Pomeranian University of Technology in Szczecin, Poland

**Abstract** - *Monitoring of machines, machine tools and robots in local manufacturing systems, especially the monitoring of communication between machines, robots work centers and other networked devices in computer integrated manufacturing is an important part of security assurance in a global manufacturing environment. The presented monitoring methods use 2D bar code technology including QR codes and Datamatrix, mobile devices currently available on the market, including smart phones (capable to utilize WiFi, 3G and UMTS technology) such as iPhone, Android, Windows Mobile and Symbian systems as well as regular UNIX-based computer systems and selected diagnostic software tools available via the Internet on GNU license. The compilation of 2D bar codes, mobile devices with WiFi, 3G and HSPDA and UMTS technology, local networks and UNIX systems, software and commands - allows the introduction of a useful, flexible and scalable monitoring system, capable of being implemented in any size and any type of manufacturing systems.*

**Keywords:** 2D codes, monitoring, machines, machine tools, manufacturing systems

## 1 Introduction

The basic idea is to use 2D bar codes, mobile devices and standard hardware, protocols and software modules (which are already built into the latest manufacturing machines, computer systems and mobile devices) – for inventory management, monitoring and periodic audits of all networked resources within manufacturing systems, especially production networks, machine tools, robots and technological devices.

The wide offering of mobile devices which can read and interpret 2D bar codes, and – on the other hand - possibility use standard hardware components, compatible software and standard protocols in manufacturing machines, make this idea capable of implementation. The following technologies and devices are applicable here:

- 2D codes (two dimensional bar codes)
- mobile devices – smart phones with GSM, 3G, EDGE, HSPDA, UMTS and WiFi modules
- UNIX systems with installed monitoring and front-end software
- LAN networks with wireless technology WiFi

## 2 Two dimensional bar codes

2D bar codes present data in a two dimensional matrix or grid pattern and are a kind of portable, small database in itself or an enter key to a larger database (opposite to regular - 1D bar codes, which are usually only an enter key to a database). 2D bar codes can be used to store all kinds of data, including URLs.

As of year 2011 - there are a number of two dimensional bar codes in use worldwide, invented or developed by different resources, such as [1]:

- Bumpy code - US Patent 5393967
- 3-DI - Lunn LTD - US Patent 5554841
- ArrayTag - US Patent 5202552
- Aztec Code - US Patent 5591956
- Codablock - US Patent 5235172
- Code One (Ted Williams, 1992)
- Code 16k (Ted Williams, 1989)
- Code 49 (David Allais, 1987), public domain
- Color Code (proprietary code Korea Yonsei University)
- CP code (proprietary code, CP Tron, Inc.)
- DataGlyphs (Xerox PARC, US Patent 5245165)
- Datamatrix (Robotic Vision, ISO)
- Datastrip Code (proprietary code, Datastrip Inc.)
- Dot Code A (Philips)
- HCCB High Capacity Color Bar Code (Microsoft Corporation)
- HueCode (Robot Design Accessories)
- ITACTA.Code (proprietary Code INTACTA)
- UPS Code/ MaxiCode/ Code6 (UPS)
- MiniCode (proprietary code Omnipapan Inc.)
- PDF417 and microPDF417 (Symbol Technologies)
- QR (Quick Response) Code (DensoWave, ISO)
- Snowflake Code (Electronic Automation Ltd)
- SuperCode (Ynjiun Wang) US Patent 5481103
- Ultracode (Zebra Technologies) US Patent 5481103

Most of the presented 2D codes are patented and subject to property rights of the respective owners, however some property owners have put their code in the public domain.

The following 2D codes are especially useful for utilization in a manufacturing environment, because they are in the public

domain and have technical features such as capacity and a high level of redundancy, for example:

- Datamatrix: - covered by several ISO Standards such as ISO/IEC 16022:2006 [2]. Datamatrix is capable to encoding up to 3116 characters. A sample Datamatrix code is presented in figure 1.



Fig. 1 – Sample Datamatrix Code  
Generator: qrcode.kaywa.com [3]

- QR Quick Response Code – a trademark registered to Denso Wave Corp. QR code is covered by ISO Standard ISO/IEC 18004 [4, 5]. QR Code is capable to encoding up to 7366 characters or 4466 Alpha numeric characters (with ability to encode domestic characters such as Japanese Kanji or Kana characters) A sample QR code is presented in figure 2.



Fig. 2 – Sample QR-Code (ULR - link)  
Generator: datamatrix.kaywa.com [6]

In monitoring applications presented here - 2D codes are used to mark machines, work centers and other inventory in the manufacturing area.

Each 2D code physically attached to a machine can point mobile devices used by service staff to a permanent web address with monitoring results or with any other information, such as maintenance or an emergency contact.

Example of machine marked with 2D-Code is shown in figure 3.



Fig. 3 - Manufacturing machine marked with 2D-Code

### 3 Mobile Devices and Operating Systems

Mobile technology vendors offer a large number of mobile devices on the market, which are equipped with a built-in camera and the ability to download software applications, including bar code readers, decoders and managers. Based on: IDC Worldwide Quarterly Mobile Phone Tracker [7], 5 largest vendors as of Q1 2011 are as listed below:

- Nokia
- Samsung
- LG
- Apple
- ZTE

Other vendors, such as HP, HTC, Motorola, Sony Ericsson, etc., also deliver this kind of mobile devices.

Depending on the vendor - different mobile operating systems can be found in mobile devices – as of 2011:

- Android
- BlackBerry
- iPhone
- Java/ J2ME
- Palm webOS (HP Palms)
- Symbian
- Windows Mobile/ WinCE

All of these allow user installed applications, including 2D code readers.

### 4 Web browsers and 2D Code Readers

To be able to browse web pages and review the results of monitoring provided by Unix systems – a regular mobile web browser can be used.

Mobile devices have least one web browser already installed in system, such as:

- Android web browser
- BlackBerry web browser
- Firefox for mobile
- Internet Explorer Mobile
- Opera Mini
- Safari
- WebOS browser

Depending on needs - other user-installable browsers can be additionally installed in mobile device by the user via the internet.

To be able to read and manage 2D codes - owners of mobile devices, depending on mobile operating system, can download and install respective (one or more) 2D barcode readers, which have been designed for a selected operating system, such as:

- I-nigma – for iPhone, Symbian, Windows mobile, Blackberry, Android and Java [8]
- NOKIA Barcode Reader – for Symbian/ Nokia Devices [9]
- Barcode Scanner – for Android and Java [10]
- QR deCODer – for webOS/ HP Palms [11]

Many other 2D code readers and other applications are available for download via:

- App Store – for iPhone [12]
- Android Market – for Android [13]
- Marketplace for WindowsPhone [14]
- OVI Store – for Nokia/ Sumbian [15]
- BlackBerry App World – for BlackBerry [16]
- iPAQ Appstore - for iPAQ/ HP [17]

## 5 Network Monitoring Tools

The Network Monitoring Tools listing maintained by Cottrell, SLAC Stanford University [18] shows a large selection of software for regular network monitoring, which can be also used for the monitoring of manufacturing resources.

One such monitoring tool is the Oetiker and Rand's Multi Router Traffic Grapher (MRTG) [19], which is primary designed to monitor the traffic load on routers or network interfaces and - every 5 minutes - generates images, which show a live, visual representation of monitored traffic.

The sample daily, weekly, monthly and yearly graphs achieved in a manufacturing system and displayed using a mobile device: Android examples are shown in figures 4-7.

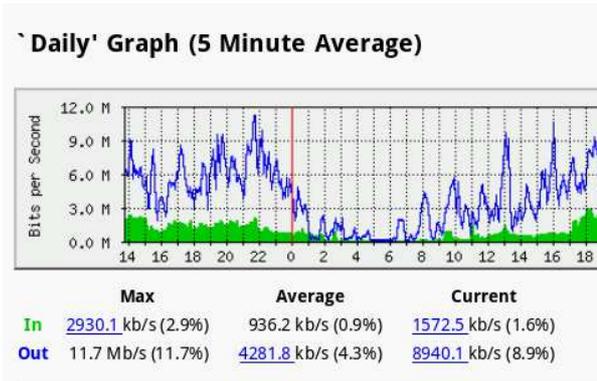


Fig. 4. Daily traffic, graph displayed on Android device

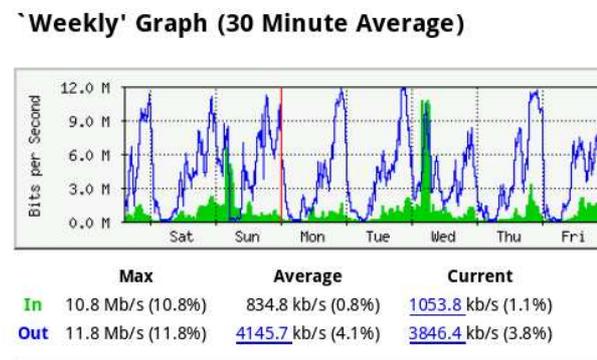


Fig. 5. Weekly traffic, graph displayed on Android device

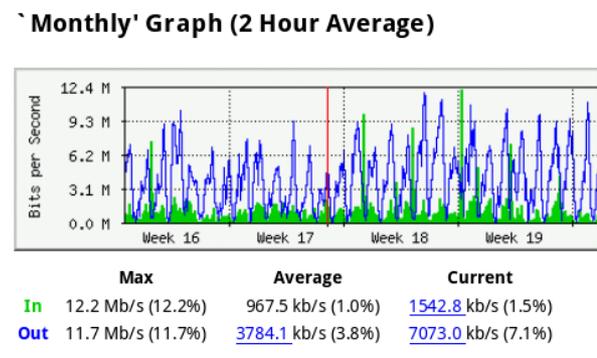


Fig. 6. Monthly traffic, displayed on Android device

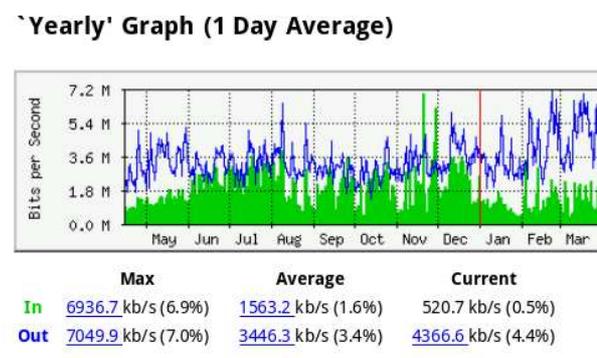


Fig. 7. Yearly traffic, displayed on Android device

## 6 Audits in Local Networks

The MRTG [18] allows the monitoring and display of any variable which is represented by a numeric value. Other useful software tools are: the Network Mapper and Security Scanner (mnap) [20] and Cron Daemon – a time-based job scheduler for Unix systems [21], which can be used for periodic audits inside local manufacturing networks. Using Cron Daemon, a system administrator can define any audit in a local network, repeated every defined (in crontab) time period. Therefore MRTG software can receive (from NMAP) representative numeric variables, which can be used for the monitoring of any networked machine, robot and/or other manufacturing resource. The status of each conducted audit for each machine (or status of each check point) can be acquired via the network and delivered to the MRTG at every 5 minutes time interval - for display on a local MRTG web page with monitoring results. Figure 8 shows a sample graph obtained while monitoring a single machine using MRTG and displayed using an iPhone 3GS with Safari browser.

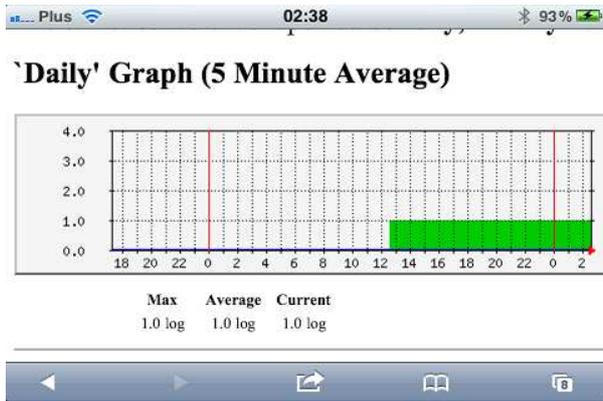


Fig. 8. Selected machine activity displayed on iPhone 3GS

NNAP can also check the status of more resources at once, just in one audit. The next figures 9-12 (below) shows sample graphs obtained while monitoring all machines and devices logged into the local manufacturing network.

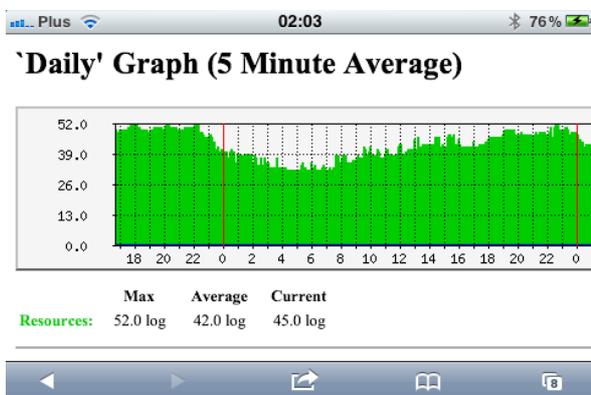


Fig. 9 Total number of machines logged into local network, daily graph displayed on iPhone 3GS

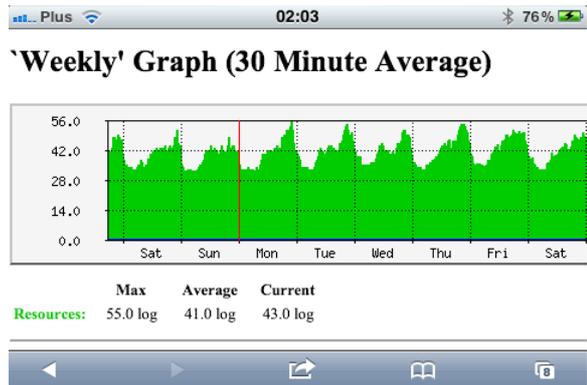


Fig. 10. Total number of machines logged into local network, weekly graph displayed on iPhone 3GS

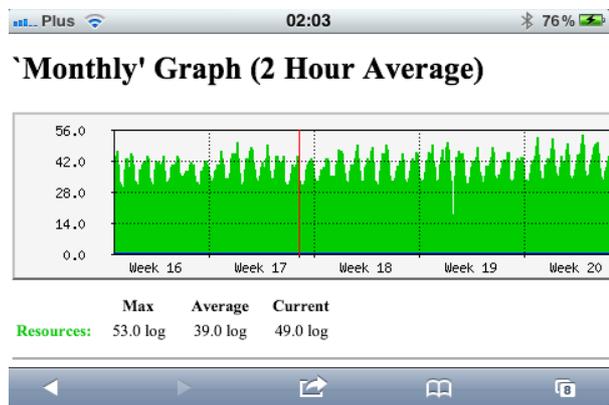


Fig. 11 Total number of machines logged into local network, Monthly graph displayed on iPhone 3GS

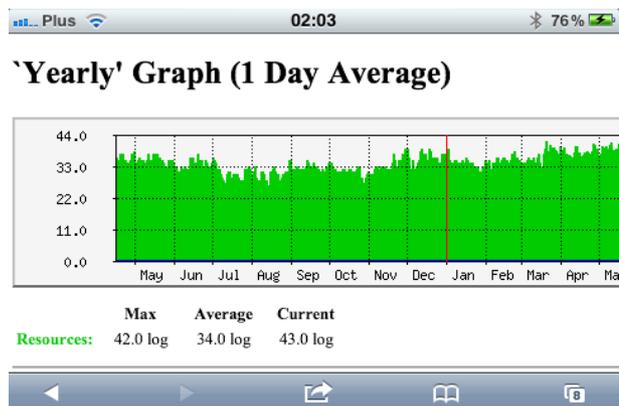


Fig. 12 Total number of machines logged into local network, yearly graph displayed on iPhone 3GS

The next graph shows the results of another audit: how continuous is the WiFi connection with the WISP (Wireless Internet Service Provider), using crontab and ping commands. The sample monitoring results of latency time and packets lost on the WiFi link is shown in figures 13-16.

'Daily' Graph (5 Minute Average)

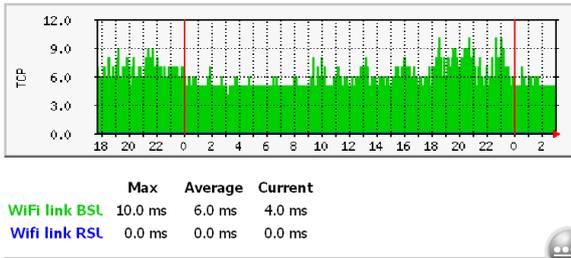


Fig. 13. Status of WiFi Link HTC HD2 – daily graph displayed on HTC HD2 with Internet Explorer Mobile

'Weekly' Graph (30 Minute Average)

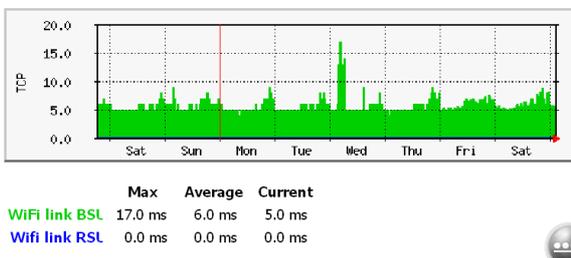


Fig. 14. Status of WiFi Link HTC HD2 - weekly graph displayed on HTC HD2 with Internet Explorer Mobile

'Monthly' Graph (2 Hour Average)

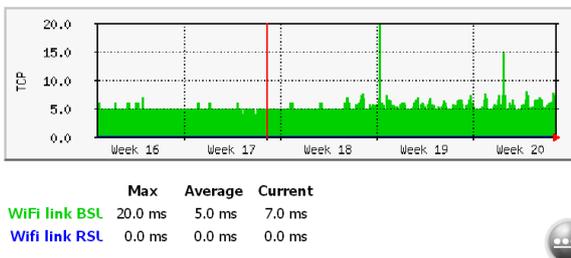


Fig. 15. Status of WiFi Link HTC HD2 - monthly graph displayed on HTC HD2 with Internet Explorer Mobile

'Yearly' Graph (1 Day Average)

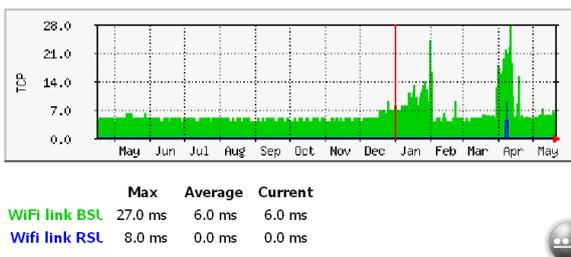


Fig. 16 Status of WiFi Link HTC HD2 – Yearly graph displayed on HTC HD2 with Internet Explorer Mobile

Last graph (figure 16) shows two issues with the monitored network, which was observed in January and in April that year (2011) in a tested local network. One issue was caused by a security problem, and the second - by raw technical problems.

## 7 Results and conclusions

Practical applicability of 2D codes and mobile devices to read Datamatrix and QR codes and to point those mobile devices to monitoring results and show those results provided by manufacturing monitoring software were conducted using the following selected mobile devices/ phones:

- iPhone 3GS with iPhone system v.4.3.1 and Safari web browser [22]
- Sony Ericsson Xperia X8 with Android 2.1-update1, and with downloaded Opera Mini web browser [23]
- HTC HD2 PocketPC T8585 with Windows Mobile 6.5 Professional CE OS 5.2 [24]
- Nokia N97 Mini with Symbian S60 and NG7.1.4 browser [25]

All of 4 aforementioned mobile systems did show their ability to read 2D codes, however in each case the 2D Code reader had to be downloaded from the proper Application Store or directly from a software vendor.

Each mobile device did log into standard WEP-Key or WPA/WPA2 secured WiFi local networks, however there were some unsolved problems with HTC HD2 phones based on Windows mobile system - to login into larger WiFi network, which was secured with: EAP-PEAP/ MSCHAPv2, where the CA CERT server certificate was required.

The Nokia N97 Mini logged into the same large WiFi network after some configuration changes were made by hand and after downloading the server certificate (using another 3G/ UMTS connection).

The iPhone 3GS logged into the same network without any problems. The iPhone found all the required security modules and protocols and downloaded the required server certificate automatically, after “one key” user confirmation of that certificate.

A quality reading process of 2D code depends very much on the quality of the camera built-into each mobile device, here iPhone was better and faster than other tested mobile devices.

After reading the 2D code each mobile device was able to display the web page with monitoring results of the selected machines or manufacturing network parameters.

The iPhone provided the most comfortable display and handling (scrolling/ zooming) of monitoring results primarily because of its ability for smooth zooming of the web browser screen. Other mentioned devices – as of today - do not have such smooth zooming ability; however zoomed pictures in all devices were good and readable.

The use of 2D codes, mobile devices and outlined monitoring software can provide help for technical staff to get prompt information on physical resources marked with 2D codes and therefore essentially accelerate the solving of network and technical issues in manufacturing systems.

## 8 Disclaimer

All remarks and references in this paper to any specific commercial product, service or resource by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by author or by ZUT, West Pomeranian University of Technology. The views or opinions of the author expressed herein do not necessarily state or reflect those of ZUT, West Pomeranian University of Technology, and shall NOT be used for any product or service endorsement purposes.

## 9 References

- [1] Adams Communications, BarCode 1, Specifications for popular 2D Bar Codes, [www.adams1.com/stack.html](http://www.adams1.com/stack.html) retrieved 2011.5.20
- [2] ISO/ IEC 16022:2006/ Cor2:2011 - Information Technology – Automatic identification and data capture techniques – Data Matrix bar code symbology specification, 2006, corrigenda 2011  
[http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=44230](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=44230) - retrieved 2011.5.20
- [3] Datamatrix Generator: <http://datamatrix.kaywa.com> retrieved 2011.5.20
- [4] ISO/IEC 18004:2000 – Information Technology – Automatic identification and data capture techniques – Bar code symbology – QR code, First edition: 2000  
[http://raidennet.net/files/datasheets/misc/qr\\_code.pdf](http://raidennet.net/files/datasheets/misc/qr_code.pdf) - retrieved 2011.5.20
- [5] ISO/IEC 18004:2006.Cor1:2009 – Information Technology – Automatic identification and data capture techniques – Bar code symbology – QR code, Corrigenda  
[http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=43655](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43655) - retrieved 2011.5.20
- [6] QR Code Generator: <http://qrcode.kaywa.com> retrieved 2011.5.20
- [7] IDC Corporate USA, Worldwide Quarterly Mobile Phone Tracker, 1Q 2010 – published 28 April 2011  
<http://idc.com/getdoc.jsp?containerId=prUS22808211> retrieved 2011.5.20
- [8] 3G Vision, I-Nigma, <http://i-nigma.mobi> - retrieved 2011.5.20
- [9] Nokia Inc., Nokia Beta Labs, Nokia Barcode Reader  
[http://nds1.nokia.com/NOKIA\\_COM\\_1/Microsites/BetaLabs/applications/apps/Nokia\\_Barcode\\_Reader\\_S60\\_32.sis](http://nds1.nokia.com/NOKIA_COM_1/Microsites/BetaLabs/applications/apps/Nokia_Barcode_Reader_S60_32.sis) retrieved 2011.5.20
- [10] Multi-format 1D/2D barcode image processing library with clients for Android, <http://code.google.com/p/zxing/> retrieved 2011.5.20
- [11] Live Runway, QR decoder from Palm Web OS, <http://www.juergentreml.de/programming/webos/qrdecoder> retrieved 2011.5.20
- [12] Apple Store, <http://store.apple.com> - retrieved: 5.2010
- [13] Android Market, <http://market.android.com> - retrieved 2011.5.20
- [14] Market Place, <http://marketplace.windeoshome.com> retrieved 2011.5.20
- [15] OVI Store, <http://store.ovi.com> - retrieved: 5.2010
- [16] BlackBerry App World, [appworld.blackberry.com](http://appworld.blackberry.com) - retrieved 2011.5.20
- [17] IPAQ Application Store, <http://www.ipaqchoice.com> retrieved 2011.5.20
- [18] Cottrell, L. (2010). Network Monitoring Tools, SLAC, Stanford University,  
<http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html> retrieved: 2011.5.20
- [19] Oetiker, Rand The Multi Router Traffic Grapher, <http://www.mrtg.org> - retrieved 2011.5.20
- [20] Lyon, NMAP Network Mapper and Security Scanner, <http://www.insecure.org/nmap/> - retrieved: 2011.5.20
- [21] Advanced Cronjob Tutorial – Online Cron Service & Cron Reference, <http://livecronjobs.com> - retrieved: 2011.5.20
- [22] Apple Inc., iPhone overview, <http://apple.com/iphone> - retrieved: 2011.5.20
- [23] Sony Ericsson, Xperia X8 Specification  
<http://www.sonyericsson.com/cws/corporate/products/phoneportfolio/specification/xperiox8> - retrieved: 2011.5.20
- [24] HTC Corporation, HTC HD2 overview, <http://www.htc.com/europe/product/hd2/overview.html> - retrieved 2011.5.20
- [25] Nokia Corporation, Nokia N97 Mini Overview, <http://www.nokia.pl/produkty/telefony/nokia-n97-mini> retrieved 2011.5.20

# Hashing Smartphone Serial Numbers

## An ASLR Approach to Preventing Malware Attacks

Mark Wilson and Lei Chen

Department of Computer Science, Sam Houston State University, Huntsville, TX, USA

**Abstract** – *The Internet and mobile devices today have merged seamlessly, giving smartphone users access to the World Wide Web, email and other network services and resources. Due to the increased popularity of smartphones they have become a very attractive target for malware. It is predicted that smartphone users will see a multitude of different malware attacks aimed at their mobile devices in the near future. This paper presents how malware can spread to smartphones, and possible routes to safeguard smartphones against attacks. Specific defensive tactics of the Symbian Operating System will be outlined and a variation of Address Space Layout Randomization (ASLR) specifically for smartphones will be presented to prevent the spread of malware from both the Internet and other smartphones.*

**Keywords:** Address Space Layout Randomization, attacks, hash, malware, Smartphone, Symbian

## 1 Introduction

A generic cellular phone, that is only offering a phone-calling feature, has become rare and is difficult to find in many cell phone service provider storefronts. The majority of cell phone users have switched to smartphones that include not only a phone-calling feature, but also wireless Internet access to check emails, update social networking website statuses, and much more. Smartphones today often include digital audio players, high mega-pixel cameras, and either external or onscreen QWERTY keyboards for text messaging and emails [8]. Arguably the most valued feature of a smartphone is also its most detrimental: Internet access. By users having constant connectivity to the World Wide Web, the possibility of malware intrusion becomes extremely likely. Malware is identified as a piece of code that affects the behavior of the operating system (OS) or other security sensitive applications without the user's consent and by a method making the alterations impossible to detect by usual means [6].

Smartphone operating systems have been attacked and infected by multiple pieces of malware, most notably the Cabir worm that infected the Symbian OS. Symbian has

employed a number of preventative measures to block the infection and spread of malware that aims to exploit weaknesses in the OS. Precautions that Symbian has installed to safeguard smartphones include a Trusted Computing Base (TCB), a Trusted Computing Environment (TCE), and Data Caging [1][7]. Though these methods hinder the infection of malware, an additional method could be introduced not only to the Symbian OS, but also to most other smartphone operating systems. A variation of Address Space Layout Randomization (ASLR) based on the hash value of each smartphone's serial number could not only prevent malware infection, but also allow telecom networks to track and trace the origin of any transmitted malicious code.

The rest of the paper is structured as follows. Next in Section 2, we survey the threats related to smartphone malware and the conventional solutions. Section 3 discusses the two major security measures, certificate signing and data caging, for Symbian Operating System. In section 4 we first briefly discuss Address Space Layout Randomization (ASLR), then introduce an ASLR based security enhancement solution to help prevent malware attacks by hashing smartphone serial number and renaming folders that need to be protected. Section 5 draws the conclusion and Section 6 proposes our future research.

## 2 Background

### 2.1 Smartphone malware threats

A number of factors contribute to weak malware protection on current smartphones. The first major reason is that a common OS is necessary for easy service creation. Unlike standard cell phones that relied solely on a proprietary OS that did not have to successfully communicate with another type of application or service, all smartphone operating systems provide the same basic foundation [2]. Powerful features such as: access to cellular networks and the Internet, multitasking for running several applications simultaneously, and data synchronization create a common ground. Unfortunately, this allows for

vast opportunities for security breaches and spread of malware infection. Most of software developers are eager to release new technologies or updated versions of current software but often neglect to properly or fully test them [10]. This failure to thoroughly harden the new software may lead to exploitation and ultimately corruption by malware. Finally, the users themselves are held accountable for some of their own habits. Similar to a PC, it is imperative that malware countermeasures such as firewalls and anti-virus monitoring be installed and working properly [4].

## 2.2 Solutions to safeguard smartphones

Though no single tactic is a failsafe to keep a smartphone malware-free, a combination of techniques can greatly reduce the likelihood of becoming infected. Installing and maintaining a secure firewall greatly limits the amount of traffic with internal or external peers. The user determines if another user is allowed access to a particular port, and the firewall will either grant or deny access to that port. Similar to PCs is the need for anti-virus software. By installing software to scan for malicious strings or patterns the user can be notified of a possible infection and can take proper measures in order to contain the malware and defend against any damage it may cause. Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) monitor the entire system for suspicious activity, including possible behaviors of malicious code. If abnormal behaviors are detected, closing ports or locking systems can result in order to prevent possible damage [5].

Smartphone hardening has been suggested in order to prevent the spread of malware to smartphone [4]. Simple actions, such as displaying the phone number of an incoming call and illuminating the LCD display when dialing, have already been employed by the smartphone to alert the user of suspicious activity. The smartphone's hardware itself can also play a role in reacting to malware infection. The smartphone's Subscriber Identity Module (SIM) card can be loaded with a clean, uninfected version of the smartphone OS. This tactic allows for the smartphone OS to be immediately reloaded with virtually no downtime. Finally, by turning off smartphone features that are not currently in use, such as Bluetooth or Wi-Fi, the chances of becoming infected may be reduced.

Due to smartphones having the ability to access the Internet and make calls, one smartphone has the potential of being screened by both the Internet and a Telecom network [4][5]. Smartphones being monitored by both of these types of networks can potentially protect the user from downloading or spreading malware. When many smartphones connect to the Internet, they are scanned in order to ensure that the latest security provisions are installed and that they will be shielded against possible threats. Internet access is denied if the smartphone is not

patched with the most current security patches. Telecom networks are able to provide protection against the spread of malware because suspicious activity is easily identified at telecom base stations. Examples of suspicious activity monitored by telecom networks include initiating a call and immediately aborting it, connecting calls without voice traffic, or prolonged data packet transmission to or from a single user. If the telecom network determines that a smartphone is behaving suspiciously, the base station can limit the smartphone's rate of calling or transferring data, employ call filtering, or block the smartphone completely. These two networks have the ability to work together by notifying each other of abnormal network behavior. If either side were to alert the other, precautions including call filtering or denial of Internet connections could be employed.

## 3 Security Measures of Symbian

The Symbian OS is the most widely used smartphone operating system in use today, with Android trailing closely behind [2]. Due to the popularity of Symbian, it becomes a large target for malware, and unfortunately regardless of how secure an OS might be, there is still a potential for the contamination and spread of malware. However, Symbian employs a number of methods that enhance its security architecture.

### 3.1 Certificate signing

The Symbian OS requires that all applications must be digitally signed in order run on the user's smartphone. Symbian provides a security platform that is divided into three separate levels in order to maintain a secure environment. The deepest level of security allows an application to access the Trusted Computing Base (TCB) set, consisting of the kernel, the file server, and the software installer and its registry; however, in order for the application to gain access to the TCB, it must be certified and formally verified by Symbian [7][2]. The Trusted Computing Environment (TCE) set governs applications that require access to a limited amount of sensitive system resources [2]. Finally, the third level of trust is associated with third party applications that require a digital certification. Certifications can be performed in one of two ways depending upon the access level required of the application. Applications that must access core OS files in order to run properly must be submitted to the Symbian Signed program for approval [2]. On the other hand, applications that do not require access to sensitive or confidential system files can be self-signed by the developer and are granted limited access.

The requirement of verifying and signing applications before granting proper access in the Symbian environment is a strong strategy for containing possible malware. By locking sensitive files away from certain applications, malicious code would be unable to access and share a user's

personal information with others across networks. The self-signed applications, though only granted limited access, still pose a potential threat to Symbian users. Applications that are self-signed are able to access such limited privileges as making phone calls, initiating network connects, and accessing device location data. Malicious applications running only those features could potentially initiate and connect to a network, send data, sensitive or otherwise, and spread to another network [10].

### 3.2 Data Caging

Data Caging refers to a security architecture that divides sensitive data into separate folders each with different restrictions. The Symbian OS creates four different directories under the root file system: `\sys`, `\resource`, `\private`, and `\(other)` [7]. The `\sys` directory and subdirectories are only accessible by the trusted kernel. By restricting the `\sys` folder, it guarantees that only the trusted kernel can create executable files or load them into memory [1]. The `\resource` folder is used to store read-only files that will not be modified after installation, e.g. fonts and help files. All processes are able to read the files stored in this folder, but only the trusted kernel has permission to write to these files if modification is necessary. Each installed application creates a subdirectory under the `\private` directory in order to store files pertaining only to that particular application. The subdirectories are named using the secure identifier (SID) that identifies a running process. Applications are able to access, read, and write to their own files, but are not granted permission to read nor write to any other application's subdirectory. Finally, the `\(other)` directory has no permission restrictions and is designated as public, allowing the user to read and write to files housed in this directory [7].

The concept of data caging to restrict the user's accessibility to sensitive system files is a strong preemptive strike against malware intrusion. By limiting access to system files, the OS is able to retain its user-friendly status along with its integrity. However, thousands of Symbian users have posted protests in online forums speaking out against the limited accessibility to the files on their phones. Those users suggest Symbian keep the current settings as default settings, but allow power users to change those settings to allow themselves full access. The Internet is currently riddled with tutorials of how to hack Symbian and gain access to its system files using different applications. Due to Symbian being highly compatible with both C++ and Python programming languages, many power users are able to compose short scripts enabling applications to crack their Symbian-based smartphones.

## 4 ASLR based serial number hashing

The Symbian OS is a very strong and security conscientious operating system with very few access points for malware to breach. Nevertheless, because Symbian is

ranked the most popular smartphone OS, it is likely that it will continue to be attacked. Symbian has already been attacked by such pieces of malware as the Cabir worm, CommWarrior, Skulls, and Doomboot among others. By examining methods employed by other operating systems and modifying them for smartphone usage, the digital world may be able to create not only a strong operating system that is nearly impervious to attack, but also a deterrent to those who want to employ smartphones as a means of attack against other networks.

### 4.1 Address Space Layout Randomization

Recently Microsoft Windows Vista and Macintosh OS X Snow Leopard edition began using a method of malware prevention called Address Space Layout Randomization (ASLR). ASLR is a security method that strengthens system security by increasing the variety of possible targets to attack [11]. This is effective because many viruses, worms, Trojan Horses and other types of malware search a computer system for a particular directory, file, or file type in order to properly infect the system. ASLR assigns random strings of characters to directories and files instead of their usual names in order to prevent possible threats from finding their target file. For example, a file contained inside of directory "`\system32`" is much more susceptible to infection than the same file being contained inside of a directory that has been assigned a random name. Programming a piece of malware to hunt for a particular file that has a different file path due to unpredictable folder names may prove to be a barrier that many malware authors simply do not want to invest the time into. Overcoming these obstacles will certainly slow an infection or attack, along with making it much more conspicuous [9].

In order to successfully employ an ASLR scheme and provide a distinguishable identifier of each smartphone, the serial number of the smartphone can be used. Each smartphone has its own unique serial number that is able to provide the vendor with the model number and technical specifications. By generating a hash value, using SHA-1, MD5, or a proprietary software format, from the smartphone's serial number a seemingly random number can replace common folder and file names that can still ultimately be linked to the smartphone itself.

### 4.2 Hashing serial number

For this research, a Nokia smartphone running the Symbian OS with the manufacturer-provided serial number "010082321439976/07951780736" will be illustrated. The smartphone's serial number generates an MD5 hash of "4abb107f8f8c4dc18482948081bdc18" as shown in Figure 1. The checksum of the smartphone serial number allows for a more secure string of characters than simply the serial number alone. It is presumable that serial numbers are not purely random and particular digits or sets of digits signify

the make, model, capacity, original installed OS version number, and such, and could be deciphered by those familiar with Nokia smartphones. By hashing this identifying information, the likelihood of such data being deciphered is doubtful. To further randomize this hash number, eight consecutive characters from within the hash value are used as the folder names for the main three folders in the data-caging scheme already in place. By producing a 32-character hash string, it allows the eight character selection a total of 25 possibilities (4abb107f, abb107f8, bb107f8f, etc.); that is more than enough to apply to the security of one smartphone using this scheme and also allow for many other folder titles in future versions in the event of expanding this theory. Longer checksum values could also be used in the event of needing additional eight-character string titles. Using these randomized character strings based on the smartphone hardware itself would not only drastically reduce the occurrence of malware infection, but also provide information about the smartphone itself if an infection were to occur. Figures 2 illustrates the names of folder SYS before and after data caging, hashing serial number, and renaming.

Current file MD5 checksum value:  
 4abb107f8f8c4dc18482948081bdcbl8

Figure 1. MD5 checksum of the smartphone serial number

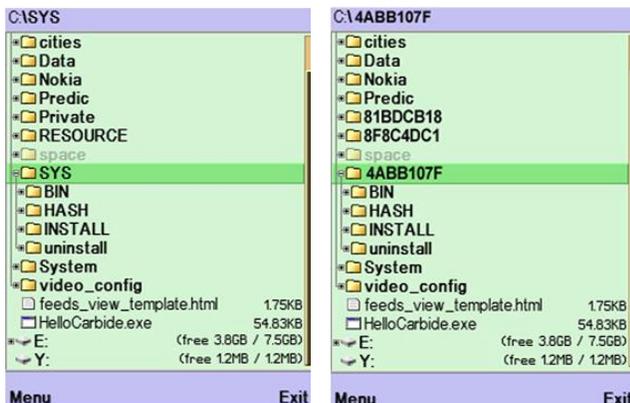


Figure 2. Folders before data caging (left) and Data caged folders renamed with eight-character hash string (right)

### 4.3 Security benefits

This variation of ASLR based on the smartphone hardware acts as a two-pronged attack against smartphone malware. First, the ASLR method compounds the difficulty of infecting the OS and ultimately prevents the smartphone from failing due to some type of malware attack. Second, because the ceasing of all intrusions, infections, and attacks, is unrealistic, the incorporation of the serial number and the information that is designated by it offers enough data about the smartphone to stop malware from spreading. By

planting the smartphone's serial number directly into the system directory, smartphone service providers would be able to associate a particular piece of malware to a specific smartphone. Possibly infected smartphones could be reported by the owner of the smartphone itself who suspects it may be infected, technical support staff diagnosing smartphone issues over the phone or in person at a service provider storefront, or by a telecom carrier itself who suspects an attack or is currently being attacked by a particular smartphone. In the case of a telecom network identifying suspicious behavior originating from a smartphone, that network would have the ability to notify an Internet network and to take necessary countermeasures as outlined in Section 2.2.

The necessary data needed by a smartphone provider would be the reverse algorithm to use the three eight-character folder titles to successfully decode the original 32-character hash value, and to search it within the database containing the hash values of serial numbers of all smartphones sold. After the infected smartphone has been properly identified, especially if it is involved in some type of attack, the service provider would deny service to that phone. This would effectively cease any attacks launched by that smartphone along with the possible spread of infection over networks.

Giving the smartphone service provider the ability to remotely access the user's infected smartphone by serial number would also allow the smartphone's hard drive to be collected and researched. This information would be imperative for smartphone antivirus development in the case of a zero day attack. Because there are so many variations of malware in the wild today, Antivirus research would also benefit by studying and creating a virus definition to harden the system against future malware.

In the most extreme cases, malware could be linked to particular smartphones owned by particular people. If that smartphone were to be engaged in multiple attacks against a telecom or Internet carrier, the source of a particular piece of malware that intentionally infected others smartphones, had a history of engaging in suspicious activity such as multiple short calls, the proper jurisdiction could prosecute that user. The majority of states in the U.S. now have laws regarding malware and the spread of malware included in their respective penal codes. Federal law 18 U.S. Code § 1030 criminalizes computer crimes such as hacking, computer fraud and the spreading of computer malware [3]. The federal law defines computer trespass as "to knowingly access a computer without authorization or by exceeding authorized access and thereby obtain information protected against disclosure," a starting point for the introduction of malware into a system. Depending upon the behavior and target of the attack, 18 U.S. Code § 1030(5)(a) would apply to smartphones and smartphone malware transmission if the user "knowingly caused the transmission of a program, information, code, or command, and as a result of such

conduct, intentionally caused damage without authorization, to a protected computer.”

## 5 Conclusion

In this paper a basic outline of smartphone malware was discussed along with selected security measures put into place by the popular smartphone OS, Symbian. Due to the increasing popularity of smartphones, malware specifically designed to infect, spread, and attack them is likely to be developed and put into operation. In order to successfully combat future threats of malware software developers need to remain diligent and continue to integrate counter measures into the operating system itself. In this paper, a variation of the standard ASLR scheme already employed by two major operating systems is proposed to further secure smartphones. To differentiate from the standard ASLR scheme, the hashing of the smartphone's serial number and the random selection of eight consecutive characters from that hash value allow for the smartphone's serial number to be retrieved directly by viewing the standard data caged folders in the Symbian OS. By obtaining the smartphone's serial number, it allows for possibilities of remotely ceasing attacks by disconnecting service, providing a sample of an infected smartphone for antivirus developers, and in the most extreme cases, gives law enforcement evidence to prosecute suspected malware developers, spammers, etc.

## 6 Future Work

Future work will include applying this model to other popular smartphone operating systems. Changes to convert this model to properly fit Windows-based, Android, or iPhone templates will vary depending upon the OS. However, because Windows and Mac OS X are already employing an ASLR technique on their desktop operating systems, a conversion to the smartphone OS employing this scheme may prove less intricate. In addition, data from Internet and telecom network providers must be gathered in order to properly test if a particular serial number can be linked to a specific person by their active account information. Furthermore, it must be researched to discover how crimes generated from smartphones are to be prosecuted in the event of a federal offense.

## 7 References

- [1] T. Badura and M. Becher, "Testing the Symbian OS platform security architecture," 2009 international conference on advanced information networking and applications, May 2009.
- [2] D. Barrera and P.C. van Oorschot, "Secure software installation on smartphones," IEEE security and privacy, December 2010.
- [3] S.W. Brenner, "U.S. cybercrime law: defining offenses," Information system frontiers, 2004.
- [4] C. Guo, H.J. Wang, and W. Zhu, "Smart-phone attacks and defenses," Third workshop on hot topics in networks, HotNets III, November 2004.
- [5] S. Khadem, "Security issues in smartphones and their effects on telecom networks," Chalmers university of technology, August 2010.
- [6] J. Rutkowska, "Introducing stealth malware taxonomy," White paper of COSEINC Advanced Malware Labs, November 2006.
- [7] A. Savoldi and P. Gubian, "Symbian forensics: an overview," In Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Harbin, China, Proc. IEEE, 15-17 August 2008.
- [8] A. Schmidt and S. Albayrak, "Malicious software for smartphones," Technische Universität Berlin, DAI-Labor, Tech. Rep. TUB-DAI 02/08-01, February 2008.
- [9] P. Szor, "The art of computer virus research and defense," Upper Saddle, NJ: Pearson Education, Inc., 2005.
- [10] L. Xie, X. Zhang, A. Chaugule, T. Jaeger, and S. Zhu, "Designing system-level defense against cellphone malware," 2009 28<sup>th</sup> IEEE international symposium on reliable distributed systems, December 2010.
- [11] O. Whitehouse, "An analysis of address space randomization on Windows Vista," Symantec advanced threat research, March 2007.

# Mobile Security Threats and Issues -- A Broad Overview of Mobile Device Security

Lei Zhang

Tian Jin University, Tian Jin, China

**Abstract** – *Mobile security draws more attention when mobile devices gain its popularity. Malware such as viruses, botnets, worms become a concern of using mobile devices since they leak sensitive information stored at or transmitted by mobile devices. This paper investigates malware in different platforms of mobile devices including Bluetooth, iPhone OS, and Blackberry. Countermeasures of vulnerability and attacks in mobile devices are also discussed to protect security and privacy of mobile devices.*

**Keywords:** mobile security, Bluetooth, blackberry, iPhone

## 1 An overview of mobile device security

In today's world, mobile devices are becoming more and more popular. As these devices have begun to spread, the demand for more and better functionality has come with them. However, more functionality leads to more complexity of the operating systems in various mobile devices. However, when involving in an operating system, the mobile devices are much more vulnerable to bugs, crashes, and security holes. When a system adapts to different functions, these functions might mess up with each other unexpectedly and cause it work strangely or improperly. With the plain fact that mobile devices are completely integrated into almost every aspect of our live, they leave a question, "is security an issue?" This question was answered by the first virus for a mobile computer, the cabir worm. Viruses, worms, and other malwares are always concerns since they can steal information and render devices useless. Since the mobile devices always access to the websites, wirelessly connected to different devices, many severe security issues have been raised.

To tackle the security issues, we have to understand different concepts of security. As defined by [1], malware is software designed to infiltrate a computer system without the owner's informed consent. The expression is a general term used by computer professionals to mean various forms of hostile, intrusive, or annoying software or program codes. When applying this term to mobile devices, it is in essence the same thing, but is even harder to tackle the serious problems caused by it. There are many different operating systems, and even more diverse functionality of each one, it is hard to have a powerful antivirus software that will run on all of the different operating systems and kill all kinds of viruses. It has been thought by the companies that the complexity a virus has to achieve makes it difficult to create a big number of viruses.

This misleading security ignorance creates fundamental security risks for the software systems. Just like people said "If we don't know a back door exist means we will not look for it". This idea is the foundation of many the problems in mobile security.

## 2 History of mobile malware

As mentioned in [3], Cabir, a computer worm developed in 2004 is designed to infect mobile phones running Symbian OS [2], which is an operating system designed for mobile devices and smartphones. It is believed to be the first worm that infected mobile phones. When a phone is infected by Cabir, the message "Caribe" is shown on the phone's display, and is appeared every time when the phone is turned on. The worm then attempts to spread out to other phones in the area using Bluetooth technology. The worm was not sent out into the wild, but sent directly to anti-virus firms, who believed Cabir in its current state is harmless. However, it does prove that mobile phones are also vulnerable to the viruses. Experts also believe that the worm was developed by a group who call themselves 29A, a group of international hackers. They created a "proof of concept" worm in order to catch world's attention. The worm can attack and replicate on Bluetooth enabled Series 60 phones. It tried to send itself to all Bluetooth enabled devices that support the "Object Push Profile". It can also infect non-Symbian phones, desktop computers and even printers. Cabir does not spread if the user does not accept the file-transfer or does not agree with the installation. Some older phones would keep on displaying popups. Cabir persistently re-sends itself and renders the User Interface until "yes" is clicked.

Even though the Cabir virus is credited as the first mobile device virus, it was only regarded as a concept virus. All the virus did was to show that a virus could be created based on the Symbian operating system. The codes were written to spur the development of operating system's creator, so that the security level of the operating system can be improved. However the source codes were leaked into the internet and modified, which made the virus more malicious than originally intended. About a month after the cabir worm struck, the next mobile virus, called "Duts" appeared. Duts was the first virus for the windows CE platform, and the first file infector for mobile devices. The duts virus would infect the executables in the root directory of the device if user permitted. Soon after duts, the brador virus came out. The Brador virus was the first backdoor virus for mobile devices. Backdoor is an open port

that waits for a remote host to connect to it. The viruses get into the system through the backdoor without being discovered [9].

After the brador virus, there were a large number of viruses for the Symbian Operating System, most of them are

Trojans. The reason these kinds of virus accomplished is because the operating system allowed games and other programs downloading. During the time, the codes were altered to include the virus that changes customizations on the phone and render it useless.

Table 1 Summary of Mobile Device Malware [9]

Name	Date detected	Operating system	Functionality	Infection Vector	Number of Variants
Worm.SymbOS.Cabir	Jwune 2004	Symbian	Propogation via Bluetooth	Bluetooth	11
Virus.WinCE.Duts	July 2004	Windows CE	File infector	— (File API)	1
Backdoor.WinCE.Brador	August 2004	Windows CE	Provides remote access via network	— (Network API)	1
Trojan.SymbOS.Mosquit	August 2004	Symbian	Sends SMS	SMS	1
Trojan.SymbOS.Skuller	November 2004	Symbian	Replaces icon file	OS 'vulnerability'	12
Worm.SymbOS.Lasco	January 2005	Symbian	Propagates via Bluetooth, file infector	Bluetooth, File API	1
Trojan.SymbOS.Locknut	February 2005	Symbian	Installs corrupted applications	OS 'vulnerability'	2
Trojan.SymbOS.Dampig	March 2005	Symbian	Replaces system applications	OS 'vulnerability'	1
Worm.SymbOS.Comwar	March 2005	Symbian	Propagates via Bluetooth, MMS	Bluetooth, MMS	2
Trojan.SymbOS.Drever	March 2005	Symbian	Replaces antivirus applications boot function	OS 'vulnerability'	3
Trojan.SymbOS.Fontal	April 2005	Symbian	Replaces font files	OS 'vulnerability'	2
Trojan.SymbOS.Hobble	April 2005	Symbian	Replaces system applications	OS 'vulnerability'	1
Trojan.SymbOS.Appdisabler	May 2005	Symbian	Replaces system applications	OS 'vulnerability'	2
Trojan.SymbOS.Doombot	June 2005	Symbian	Replaces system applications, installs Comwar	OS 'vulnerability'	1
Trojan.SymbOS.Blankfont	July 2005	Symbian	Replaces font files	OS 'vulnerability'	1

### 3 Vulnerabilities and threats of mobile devices

Mobile devices security is a relatively new technology because there is still not a large focus on it. Sadly enough, the only way that the security is going to develop is by the appearance of a large amount of mobile devices malwares which need to be dealt with immediately without further avoidance. This is not to say that the current devices do not have any form of security, sometimes users are uneducated

and render these measures ineffective [9]. Until people are properly taught what to do or what not to do, they will be more aware of security issues. Certain things like Bluetooth or Wi-Fi often time enabled by default on new mobile devices which are huge security risks. There are simple solutions for these problems; installing the newest firmware on devices, turning Bluetooth off when not in use, not connecting to unsecured wireless networks, not opening strange emails, and not running programs that you don't know what they do. These are the simple precautions people can take that will

eliminate the great majority of the mobile device vulnerabilities. This should be regarded as an extreme concern because of the nature of mobile devices. Often time triggered viruses are designed to make money off the ads or the other schemes. It is almost impossible to completely avoid the time triggered viruses if they are put onto a mobile device. This makes the mobile devices very attractive targets to the hackers. Most threats to mobile devices are in the form of worms, "a self-replicating virus". This is the biggest issue since mobile devices are designed to communicate with other devices. For this reason, the virus on the compromised mobile device spreads out, is now in leads to a possibly very devastating virus [9].

## 4 Security threats and countermeasures

While mobile phones are becoming more and more ubiquitous, they also have involved in more than just phones. They can be treated as a personal computer, video camera, portable media player, GPS, and more. This results in each mobile phone storing a lot of private information, which lead to the more frequent occurrence of the security issues.

### 4.1 How Bluetooth works

Today mobile phones usually come with an advanced built-in technology known as Bluetooth. Bluetooth is a wireless communication standard that allows up to eight Bluetooth enabled devices to communicate with each other within a range of 10 meters, creating a Personal Area Network (PAN). The Bluetooth protocol works at 2.4GHz frequency spectrum and uses low power mode. Bluetooth can handle device interferences, by using a frequency hopping technology where the "transmitters change frequencies 1,600 times every second (1)." Bluetooth technology can connect various devices such as a laptop computer, PDA, smart phone, not only two similar devices. Whatever the devices are, their connection setup can always be placed into two categories, a master-master connection and a master-slave connection. In master-master connections, both devices have input devices and can dynamically communicate with each other. In master-slave connections, one device does not have an input device while the other does. An example of this kind of connection would be a mobile phone and a wireless Bluetooth headset. The headset relies on preprogrammed instructions to complete setup and communication [3].

### 4.2 Discovery, pairing and binding

In order for two Bluetooth devices begin communicating, they first need to locate each other. This can be done through a process known as discovery. During the discovery process, one Bluetooth device scans for the other within its transmission range. Once the Bluetooth devices discover each other, the two devices will complete the next process known as pairing. Pairing is similar to networking TCP/IP handshaking. The devices exchange messages such as

address, version, and pairing code. The pairing code can be thought as a password. In a master-master connection, both device users have to enter the pairing code. In a master-slave connection, the slave device will automatically read the pairing code from its preprogrammed code. Once identical pairing codes are entered, a link key is generated. The link key is used for authentication. Based on the link key the two devices dynamically generate and share an encryption key. The encryption key is used in the final process known as binding. The key binding connection means no other device can interfere or snoop on the connection. Although these three processes can keep Bluetooth connections safer, not all Bluetooth communication channels require them [3].

### 4.3 Bluetooth security modes

Every Bluetooth device has three major security modes in which it can operate on. The first mode is known as non-secure security mode. In this mode, the features such as authentication, encryption, and pairing are not enforced. The second mode is known as the service-level security mode. In this mode, a central security manager restricts access to the device by performing authentication. The last mode is called the link-level security mode. In this mode, authorization and security procedures are enforced and implemented before an establishment of a communication channel. This mode typically involves in using the previously described processes of pairing and binding. Overall, Bluetooth has transformed wireless communication as it is widely implemented and supported. Unfortunately, like many protocols, it suffers from security threats and vulnerabilities [8].

### 4.4 Bluetooth attacks

One of the least serious and harmless Bluetooth attacks is called BlueJacking. This attack takes advantage of a small loophole in the messaging protocol and allows a Bluetooth device to send an anonymous message to a target Bluetooth device. When two Bluetooth devices wish to communicate with each other they must first perform an initial handshake process in which the initiating Bluetooth device must display its name on the target Bluetooth device. Instead, an attacker can send a user-defined field to the target device. BlueJacking takes advantage of this field in order to send the anonymous message [3].

A much more dangerous case, and one of the best known Bluetooth attacks, is BlueSnarfing. BlueSnarfing is the process in which the attacker connects to the victim's mobile phone through Bluetooth without the victim's attention. This attack is dangerous because the attacker can gain access to private information such as the address book, messages, personal photographs, etc. Furthermore, the attacker can initiate as well as forward phone calls. The attacker can complete this BlueSnarfing easily within 10 meters of the victim by using software tools such as Blooover, Redsnarf, and BlueSnarf [3].

## 4.5 Countermeasures

Even though mobile phones face security threats from Bluetooth attacks, there are still effective countermeasures that can be used for protection. The simplest action can be taken is to disable Bluetooth completely on the mobile phone. Alternately, the mobile phone's Bluetooth settings can be switched to an undiscoverable or hidden mode. It is important to be aware of Bluetooth attacks and take countermeasures, as Bluetooth attacks are one of the primary ways mobile phone data is compromised [8].

## 4.6 Mobile denial-of-service

Compared with Bluetooth attack, Mobile Denial-of-Service (MDoS) attacks can be the worst attacks on a mobile phone. One of the major ways the attack is completed is through a Bluetooth enabled device. An MDoS attack can render a mobile phone useless. MDoS attacks can congest available bandwidth causing all data transfers stop, leading the phone to freeze, crash, or even restart. While there are different types of MDoS attacks, they all usually follow a similar pattern on how the attack is implemented. The attacker first uses some sort of packet-generation software in order to create infinite and sometimes malicious packets. These packets can then be sent to the victim's mobile phone using a specified protocol. One reason these attacks are considered dangerous is that they are easy to be executed. MDoS ready-to-go tools can easily be found on the Internet and downloaded. These attacks are possible if there is a loophole found in Bluetooth communication. Bluetooth technology does not have a way to handle incoming packets, and therefore does not inspect them at all. Compared with a normal mobile phone user, the problem seems to be more serious to a business mobile phone user, since he or she who depends on the phone for work can be devastated during an MDoS attack. The attack could limit their ability to access important data, significantly slow down their connection speed, and could even cause entire disconnection. Mobile phone users need to be aware that MDoS attacks can and do happen [3].

## 4.7 Mobile denial-of-service attacks

BlueSmacking is a common type of MDoS attack. The basic idea behind the attack is to send oversized data packets to the mobile device. Mobile devices using Bluetooth have a size limit on the packets that they can receive. This size difference depends on the manufacturer and model of the phone. This means that the devices cannot handle packets that are greater than the size limit. The attacker takes advantage of this weakness and sends oversized data packets to the target device. The device will not be able to handle numerous, constant, oversized packets thus resulting in a denial-of-service [3].

The second MDoS attack, although not very popular, is called Jamming. As described earlier, Bluetooth works in the 2.4GHz frequency range and it handles interferences by frequency hopping. In a Jamming attack, the entire frequency band has to be jammed so that the Bluetooth device has no available frequency to use. The amount of work the attacker has to put in for a Jamming attack is not feasible resulting in the attack's unpopularity [3].

The third common MDoS attack is called a failed authentication attack. This attack prevents two Bluetooth devices from establishing a connection with each other. In order for the attacker to be successful, the hacker must flood the target device with spoofed packets while the target device is trying to connect with a desired device. In doing so, the target device's resource becomes congested and the target device is unable to make the connection with the desired device [3].

## 4.8 Countermeasures

Mobile phone users should be aware of MDoS attacks and also realize that there are countermeasures that are available in order to protect themselves from these attacks. One of the simplest things a user can do is to keep their phone up to date by downloading and installing the latest patches and upgrading their mobile phone. Another countermeasure is simply not to accept an unknown incoming message via Bluetooth. Users should only pair their mobile phone with known devices [2].

# 5 Mobile operating system

## 5.1 iPhone OS

The iPhone operating system has had several documented vulnerabilities so far; however they are generally fixed very quickly. The "app review" process is the main reason why there are not many documented cases of malware for the iPhone. All of the applications that have permission to run on the iPhone are very carefully inspected by apple and insured not have any viruses hidden inside or security risks. This is a double edged blade. With the very strict process, there is a much more limited base on what could be brought out for the phone if any application could be used on it [4].

The main security risk in the iPhone is when the system has its root password cracked by "jail breaking". The reason this is a problem as it gives the users root access to the phone with a username and password, but if people forget to change the username and password then it is easy to log in. With root access, it enables programs or processes to access any part of the system and modify them [4].

The world's first iPhone worm was found in early November 2009. The worm would replace the background on the iPhone with a picture of Rick Astley and the words "iKee is never gonna give you up". Once installed, the malware will

search the phone network for other vulnerable iPhones and infect them [11].

The worm is a breakthrough purely because it is the first worm for one of the world's most prominent cell phone, iPhone. Hopefully, it will force people to take more care of their phone and remember to change their passwords.

The second worm infecting iPhone takes advantage of the same security hole as the previous one. This worm will redirect customers' Dutch online bank to a phishing site that will capture their information [10].

## 5.2 Blackberry Architecture Overview

The Blackberry smart phone was developed by Research in Motion (RIM) and introduced to the public as a two-way pager in 1999. In 2002, RIM released the blackberry with updated feature like push e-mail, mobile telephone, text messaging, internet faxing, web browsing and other wireless information services. RIM developed a proprietary software platform named BlackBerry OS for its BlackBerry line of handhelds. BlackBerry OS provides multi-tasking and makes heavy use of the devices specialized input devices, particularly the trackball or touch screen [1].

BlackBerry OS uses the Java to provide an open platform for third-party wireless enterprise application development. Using BlackBerry MDS Studio and the BlackBerry Java Development Environment (JDE), the BlackBerry Enterprise Solution lets software developers create third-party Java applications for BlackBerry devices. After the application is written in Java, it is compiled into Blackberry proprietary .cod files. The Java byte code is "pre-verified" as valid on the PC side (in accordance with J2ME standards) before being compiled into a .cod file. It can then be transmitted to the BlackBerry for execution [1].

By default, unsigned applications have very limited access to this enhanced functionality. Applications must be signed by RIM in order to perform actions, which are deemed sensitive such as enumerating the Personal Information Manager or reading emails. Even signed applications may require user permission to carry out sensitive actions such as initiating phone calls. RIM provides a way for third party applications to gain full access to the Blackberry API by signing it with a hash function. For developers to obtain signatures for their applications they must first fill out an online form and pay a 100 USD fee to receive a developer key. RIM provides a signing tool that sends the SHA1 hash of the application to RIM. Once this hash is received by RIM they will in turn generate a signature. This signature is then sent back to the developer and appended to the application [1].

## 5.3 Blackberry Vulnerabilities

Since 2007, there were 11 known vulnerabilities that affected the blackberry Smartphone. Five of the vulnerabilities

were cause by an error within the PDF distiller (KB17118, KB17119, KB15770, KB15766 and KB18327). Three were caused by an error within ActiveX (KB16248, KB16469, KB13142). One vulnerability was caused by the Microsoft GDI component that BlackBerry products use (KB15506). Two Vulnerabilities exist in the Session Initiation Protocol (SIP) implemented on a BlackBerry 7270 Smartphone running BlackBerry Device Software 4.0 Service Pack 1 Bundle 83 and earlier (KB12700, KB12707).

- KB17118, KB15770, KB15766 and KB17119: the PDF distiller of some released versions of the BlackBerry Attachment Service. This vulnerability could enable a malicious individual to send an email message containing a specially crafted PDF file, when opened on a BlackBerry Smartphone, could cause memory corruption and possibly lead to arbitrary code execution on the computer that the BlackBerry Attachment Service runs on.
- KB18327: multiple security vulnerabilities exist in the PDF distiller of some released versions of the BlackBerry Attachment Service component of the BlackBerry Enterprise Server. These vulnerabilities could enable a malicious individual to send an email message containing a specially crafted PDF file, when opened on a BlackBerry Smartphone associated with a user account on a BlackBerry Enterprise Server, could cause memory corruption and possibly lead to arbitrary code execution on the computer that hosts the BlackBerry Attachment Service component of that BlackBerry Enterprise Server.
- KB16248: an exploitable buffer overflow exists in the BlackBerry Application Web Loader ActiveX control that Internet Explorer uses to install applications on BlackBerry devices.
- KB16469: A buffer overflow exists in the DWUpdateService ActiveX control that could potentially be exploited when a user visits a malicious web page that invokes this control.
- KB13142: When using Internet Explorer to view the BlackBerry Internet Service or T-Mobile My E-mail web sites that use the TeamOn Import Object ActiveX control, and when trying to install and run the ActiveX control, the ActiveX control introduces the vulnerability to the system.
- KB15506: These vulnerabilities expose the BlackBerry Attachment Service and the BlackBerry Desktop Manager to attacks that could allow a malicious user to cause arbitrary code to run on the computer on which the BlackBerry Attachment Service or the BlackBerry Desktop Manager is running.
  - If a BlackBerry Smartphone user is on the BlackBerry Enterprise Server or BlackBerry Professional Software is with BlackBerry Attachment Service running, and the user tries to use the BlackBerry Smartphone to open and view a WMF or EMF image attachment in a received email message sent by a user with malicious intent, the computer on which the

- BlackBerry Attachment Service is running could be compromised.
  - o If the BlackBerry Smartphone user uses BlackBerry Media Sync to synchronize an image created by a user with malicious intent, the computer on which BlackBerry Media Sync is running could be compromised.
- KB12700: The BlackBerry 7270 Smartphone user receives a malformed SIP INVITE message. When the BlackBerry Smartphone user tries to make a call using the Phone application, the following problems occur:
  - o An uncaught exception error message is displayed.
  - o When the BlackBerry Smartphone user tries to initiate a call, the following error message is displayed: Cannot connect. Call in progress
  - o The BlackBerry Smartphone cannot receive incoming calls. The BlackBerry Smartphone does not ring or display any indication of incoming calls.
- KB12707: A BlackBerry 7270 Smartphone receives a malformed SIP INVITE message. The following problems occur on the BlackBerry Smartphone:
  - o The BlackBerry Smartphone user cannot make a call using the Phone application
  - o The BlackBerry Smartphone may ring when it initially receives the malformed message, but does not receive incoming calls afterward (i.e. the BlackBerry Smartphone does not ring or display any indication of incoming calls).

- Spoofing: A situation where there is the opportunity to spoof information upon which the user will make a decision which may impact the security of the device.
- Data Interception or Access: A situation where data can be intercepted or accessed by malicious code that is on the device.
- Data Theft: A situation where data can be sent out of the device by malicious code that is on the device.
- Backdoor: A situation where malicious code resident on the device is able to offer functionality that would allow an attacker to gain access at will.
- Service Abuse: A situation where malicious code resident on the device is able to perform actions that will cause the user higher service cost.
- Availability: A situation where malicious code resident on the device is able to impact the availability or integrity of either the device or the data upon it.
- Network Access: A situation where malicious code resident on the device is able to use the device for one or more unauthorized network activities. This may include port scanning or alternatively using the device as a proxy for network communications.
- Wormable: A technology can be utilized by malicious code on the device to further help in its propagation in a semi-autonomous fashion [8].

The following table shows for each of the areas analyzed their susceptibility to these attacks, and how they may be mitigated:

## 6 Blackberry Attack Surface

There are multiple attack surfaces an attacker can exploit to compromise the confidentiality, integrity and availability of the blackberry smart phone.

Table 2 Vulnerability surfaces and misuses [1]

Sub- System	Spoofing	Data Interception /Access	Data Theft	Backdoor	Service Abuse	Availability	Network Access	Wormable
JAD Files	AI							
File System		AO						
SMS		FAI		FAI	FAI			FAI
Bluetooth			FAIO	FAIO				
Email		FAI		FAI				FAI
PIM			A			A		
TCP/IP				FAI			FAI	
HTTP			FAI	FAI			FAI	
Telephony		A	A		A			

**Legend:**

F: Firewall      A: Application Control/Permissions      I: IT Policy      O: Other Device Settings

The chart shows attacks requiring malicious code to be present on the device. The only way for malicious code to get into the device is through user interaction. Ignorant users may trigger action of malicious code through user interaction. These facts highlight the need for user education about safe computing practices when using all kinds of computing devices including mobile devices.

## 7 Future development

The largest problem with mobile security is there is not enough time dedicated to it when designing a mobile device. For the most part, the malware can only access if the user does something to make the system vulnerable in some way or fashion. Be it running a program that has the malware hidden in it, or cracking the system so that the built in security is removed. Many experts argue that the only thing that will make users more aware is a large amount of malware forcing people to become educated or else leave them unable to use their devices. The reason for this is because in the early 2000 there were a large number of viruses that completely debilitated networks. This in turn made people understand the importance of antivirus and their threats that they don't recognize. Since then, people have been much more careful with their computers. Due to this positive response, many people think this is the only way to make people pay attention to mobile devices security. In example, there have been many proofs of concept viruses that target phones just to show it can be done and explain it could have been even worse; however this generally is circulated through the technical world and never reaches the end users on a large scale.

## 8 References

[1] BlackBerry Internet Service. *Feature and Technical Overview*, 2009.

[2] Fadia, A.. *Hacking Mobile Phones*. Course Technology PTR, 2005.

[3] Franklin, C., & Layton, J. (n.d.). *How Bluetooth Works*. Retrieved December 1, 2009, from HowSuffWorks.com: <http://electronics.howstuffworks.com/bluetooth1.htm>

[4] Kabay, M. E. *iPhone security, Part 1*. Retrieved 2009, from Network World: <http://www.networkworld.com/newsletters/sec/2009/051809sec1.html?page=1>

[5] Kabay, M. E. *iPhone Security, Part 2*. Retrieved December 1, 2009, from Network World: <http://www.networkworld.com/newsletters/sec/2009/051809sec2.html?ry=gs>

[6] Kleidermacher, D. *The future of mobile devces*. Retrieved December 1, 2009, from Hearst Electronic Products:

[http://www2.electronicproducts.com/The\\_future\\_of\\_mobile\\_devices-article-facn\\_GREENHILLS\\_apr2009-html.aspx](http://www2.electronicproducts.com/The_future_of_mobile_devices-article-facn_GREENHILLS_apr2009-html.aspx)

[7] O'Connor, J. Attack Surface Analysis of BlackBerry Devices. *White Paper: Symantec Security Response*, 2007.

[8] Sanpronov, K. *Bluetooth, Bluetooth Security and New Year War-nibbling*. Retrieved December 1, 2009, from VirusList.com: <http://www.viruslist.com/en/analysis?pubid=181198286>

[9] Shevchenko, A. *An overview of mobile device security*. September 21, 2005. Retrieved December 1, 2009, from Viruslist.com: <http://www.viruslist.com/en/analysis?pubid=170773606>

[10] The Register. *iPhone Worm Infects Devices and Redirects Duth Online Bank*. Retrieved December 1, 2009, from CyberInsecure.com: <http://cyberinsecure.com/iphone-worm-infects-devices-and-redirects-dutch-online-bank-users-to-a-phishing-site/>

[11] The Register, Sophos. *World's First iPhone Worm Hits iPhone Owners In Australia*. Retrieved December 1, 2009, from CyberInsecure.com: <http://cyberinsecure.com/worlds-first-iphone-worm-hits-iphone-owners-in-australia/>

# Chaos-Based Symmetric Key Cryptosystems

Christopher A. Wood

Department of Computer Science, Rochester Institute of Technology, Rochester, New York, USA

**Abstract**—Chaos theory is the study of dynamical systems that are highly sensitive to initial conditions and exhibit seemingly random behavior. From the perspective of cryptography and information security, randomness generated from entirely deterministic systems is a very appealing property. As such, the application of chaos in modern cryptography has been a topic of much research and debate for over a decade.

This paper presents an overview of chaotic dynamics and their role in symmetric key chaos-based cryptosystems from both a theoretical and practical perspective. It is argued that chaos-based ciphers are not likely to succeed until a valid and accepted definition of discrete chaos in finite domains is established, the inefficiencies of chaos-based cipher implementations are improved, and thorough security analysis reveals them to be comparable to standardized cryptographic primitives.

**Keywords:** Chaos Theory, Cryptography, Dynamical Systems, Symmetric Key Cryptosystems

## 1. Introduction

Ever since the discovery of chaotic behavior in the mathematical models of weather systems by Edward Lorenz in the early 1960s [1], chaos theory has found its way into many different fields of science, including physics, economics, biology, and even philosophy. In recent years its influence has begun to spread into cryptography. The sensitivity to initial conditions and seemingly random behavior produced from deterministic equations caught the eye of cryptographers as they tried to incorporate these properties into cryptographic primitives, including symmetric key ciphers, hash functions, and pseudorandom number generators.

Although chaos-based symmetric key cryptosystems are very appealing at a theoretical level, they don't provide the same cryptographic assurances that come with standardized cryptosystems like the Advanced Encryption Standard (AES). Based on research efforts throughout recent years, chaos-based symmetric key cryptosystems are not likely to succeed and thrive into the future unless the following conditions are met:

- 1) A valid and accepted definition of discrete chaos in finite domains is established, likely stemming from the discrete Lyapunov exponent
- 2) The inefficiencies of cipher implementations that are based on real-value chaotic maps are improved

- 3) Thorough security analysis reveals that chaos-based symmetric key cryptosystems are comparable to standardized cryptosystems from a diffusion and confusion perspective

This paper explores the history of chaos-based symmetric key cryptosystems and discusses the shortcomings of past and present research efforts. We attempt to explain the properties that attribute to their lack of success and acceptance by commercial applications.

## 2. Chaos Dynamics

Chaos is best known as a sensitivity to initial conditions exhibited by dynamical systems described by differential equations or iterated mappings. In the case of chaos-based symmetric key cryptosystems, we will focus on chaotic systems that are defined by iterated mappings as they are the more likely candidates for actual implementation. It is important to note that these iterated mappings are really just recurrence equations derived from their differential equation counterparts.

For such systems, a sequence of points created by recursive iterations  $f^n(x)$  of some initial value  $x_0$  of the phase space, which is the domain of the map, is defined as a phase trajectory, or simply a trajectory. These systems must be sensitive to initial conditions, have a dense collection of points with periodic orbits, and be topologically mixing in order to be deemed chaotic [2].

Periodic orbits are recurring sequences of elements in trajectories produced by chaotic maps. In dynamical systems, a collection of points is dense if at any point  $x \in X$ , where  $X$  is the phase space,  $x$  either belongs to a subset  $A \subseteq X$  or is a limit point of  $A$ . One can see that if such a collection of points exists then all possible values in the phase space will be generated arbitrarily closely.

Topological mixing is a form of mixing that may be defined without appeal to a measure (or size) of the system. Formally, a system  $F$  possesses the mixing property if, for any two measurable sets  $A$  and  $B$ , there exists an integer  $N$  such that for all  $n > N$  the relationship in (1) is satisfied [3].

$$f^n(A) \cap B \neq \emptyset \quad (1)$$

Sensitivity to initial conditions is formally defined as a characteristic of dynamical systems where two significantly close points will rapidly diverge under  $f^n$  to produce very different trajectories as they are iterated by the map. This

characteristic is satisfied if the chaotic system has a positive Lyapunov exponent, which gauges the rate of separation of infinitesimally close initial trajectories. Specifically, two trajectories in a system's phase space with initial separation  $\delta X_0$  diverge according to:

$$|\delta X(t)| \approx e^{\lambda t} |\delta X_0| \quad (2)$$

$$\lambda = \lim_{t \rightarrow \infty, |\delta X_0| \rightarrow 0} \left( \frac{1}{t} \ln \frac{|\delta X(X_0, t)|}{|\delta X_0|} \right) \quad (3)$$

In other words, the difference between two initial trajectories will exponentially increase after a very short time depending on the magnitude of the Lyapunov exponent  $\lambda$ . Based on this rate of separation, a system is deemed chaotic if  $\lambda > 0$ . Conversely, the system is deemed "regular" if  $\lambda \leq 0$ .

One final important piece of chaotic dynamics is the notion of attractors and robust chaos. A chaotic attractor is a set of elements in the phase space towards which trajectories of the system evolve over time. The elements of trajectories produced by the system will remain in the bounds of the attractor as they are recursively generated. Furthermore, two arbitrarily close trajectories within an attractor will exhibit different and unrelated behavior within the bounds of the attractor as they are recursively iterated over time.

Robust chaotic systems are those that have an attractor even when parameters in the system undergo small changes. This is an ideal property for chaotic systems that are used for cryptography because any change in initial conditions will cause trajectories to remain within the same attractor, thus making it difficult to predict any outcome without knowing the initial conditions of the system and the iteration count.

### 3. Chaos Theory and Cryptography

At a theoretical level, chaotic systems have unique characteristics that have potential applications in cryptography. These characteristics can be related and subsequently mapped to properties of cryptographic primitives. For example, consider the mixing property of chaotic systems. By definition, chaotic maps with strong mixing properties will recursively generate regions of elements that will eventually cover the majority of the phase space and start to overlap as the system evolves over time. Cryptographic primitives possess a similar property known as diffusion, which is defined as the process by which the influence of a single plaintext digit is spread out over many ciphertext digits.

Another similarity between chaotic systems and cryptographic primitives lies in the relationship between discrete time-based system iterations and encryption rounds. Consider the following discrete time-based system:

$$x_{n+1} = rx_n(1 - (x_n)) \quad (4)$$

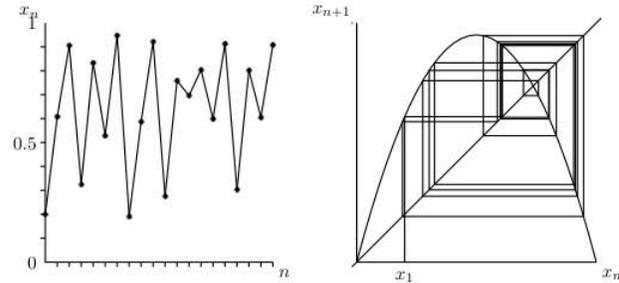


Fig. 1: The figure on the left shows a single trajectory of the logistic map as it is recursively iterated  $n$  times. The figure on the right shows a plot of the points  $(x_n, x_{n+1})$  for this same trajectory, which in turn depicts the chaotic attractor of the logistic map.

This is the recurrence relationship for the logistic map, which is derived from the differential form of the logistic equation ( $\frac{dx}{dt} = rx(1-x)$ ). The value  $r$  is a positive constant that is commonly referred to as the control parameter [4]. At first glance it might seem that this recursive map is very simple. However, after further analysis one can see that it is capable of very complicated behavior depending on the parameters of the system. Figure (1) shows one chaotic trajectory of the logistic map.

Each iteration of the logistic map produces new points in the phase space. Depending on the level of mixing within the chaotic map, this value will diverge and cover more elements in the phase space. These iterations are very similar to rounds in a cryptographic cipher, where each round serves to transform the internal state of the cipher towards the final ciphertext.

The last most significant similarity between chaotic systems and cryptographic primitives is the relationship that exists between system parameters and cryptographic keys. In a general sense, the system parameters for a chaotic map and a cryptographic key serve the same purpose, which is to determine the functional output of the system or cipher. Since chaotic maps and cryptographic ciphers are both deterministic, the system parameters and keys determine exactly what the output will be in such a way that it is statistically infeasible for an attacker to guess the output without knowledge of such values. It is typical in chaos-based ciphers for both the initial conditions and the iteration count to remain secret in order to maintain the security properties and pseudorandomness of the map.

Although the properties of chaotic maps and cryptographic primitives have similar characteristics that make them appealing to work together, there is one significant property of chaos to take into account when considering its application in symmetric key cryptosystems. Encryption schemes traditionally operate on finite sets of integers, whereas the chaotic principles discussed above usually occur on a (sub)set of real

numbers as chaotic behavior only truly exists in continuous domains. Therefore, the level of chaotic behavior exhibited by systems that operate on real numbers is closely tied to the amount of precision with which those numbers are represented.

#### 4. Chaos-Based Cipher Design

Modern symmetric key ciphers usually consist of the four operations that make up AES: key addition, S-box substitution, permutation, and linear mixing. Chaotic ciphers usually attempt to replicate these four elements using space-discretized versions of chaotic maps that approximate real-valued systems. The term discretized is very important here. Chaotic systems work in two dimensions: time and space. For example, the logistic map is discretized in the time dimension (i.e. values occur at fixed points in time or at fixed iteration counts). The phase space is still based on the field of real numbers, meaning that it is continuous in the space dimension.

One reason that these maps are space-discretized is that chaotic behavior is normally observed over the set of real numbers. However, even in real intervals such as  $[0, 1]$ , the number of elements  $x \in \mathcal{R}$  is uncountable. This implies that it is impossible to represent all of the values within this interval and, by induction, any interval on a continuum. Therefore, given the finite representation of modern computing devices we must limit the precision with which we represent elements in the phase space.

Another reason for further space-discretization of chaotic maps is for performance. By limiting the amount of precision for phase space elements for chaotic maps the amount of data that has to be stored and computed is reduced.

The approximation of a chaotic map is important in the design of chaos-based ciphers. Given the computational and memory limitations of modern computing systems there should be an efficient mapping scheme between elements on a continuum to elements of a finite set when discretizing chaos-based ciphers. The most common approach has been to partition the phase space of a continuum into a finite number of blocks [5]. The size of such blocks correlates to the degree of precision available for implementation. The more accurate the representation can be, the smaller such partition blocks can become, which in turns increases the size of the system and its phase space.

The confusion and diffusion properties of these chaotic maps must also be considered in terms of both the Euclidean geometry and Hamming distances [6]. This requirement goes back to the Lyapunov exponent measure of chaotic maps that gauges the rate of separation for trajectory elements. Approximated chaos for discrete systems must have a high rate of separation and input/output differences. Otherwise, information about the contents of the system parameters may be leaked if patterns begin to emerge through frequent periodic behavior exhibited by the map.

#### 5. Cipher Evaluation

Security, from a cryptography perspective, is measured from both the theoretical and practical levels of cryptographic primitives. At the theoretical level, cryptographic primitives are deemed secure if they possess "randomness increasing" and "computationally unpredictable" characteristics. A complete description of these properties is beyond the scope of this paper, but a brief discussion is warranted to make this paper self inclusive.

By definition, randomness increasing implies that the cryptographic primitive must increase the entropy of the system over which it operates. However, it is impossible in classical information theory for a deterministic function or map applied to a probability distribution  $P$  to increase entropy [5]. In practical implementations, however, where computational power and resources are limited, an increase in entropy may be possible. The reason for this is that given a mapping  $G : S_1 \rightarrow S_2$  ( $S_i$  are finite sets) that is applied to  $P$ , where  $P$  is a PDF for each set  $S_i$ , the result  $G(P)$  may be similar enough to approximate another distribution  $Q$ . Due to the limits of modern computing power it may be infeasible to differentiate  $Q$  from  $P$ , and if the entropy of  $Q$  is greater than that of  $P$ , then we can say that the mapping  $G$  is computationally randomness increasing. This loophole is exploited during the construction of modern chaos-based ciphers so as to hinder the application of cryptanalysis techniques to break the primitive.

One way to increase the entropy of chaos-based ciphers is to modify the order of the key and plaintext/ciphertext space. In such systems the size of these sets is directly proportional to the amount of entropy. Specifically, the entropy of a system with a key space of  $K$  keys is approximately  $\log_2 K$ . Clearly, as the order of the key space increases, then the entropy increases as well.

The initial conditions of a chaotic system also play a significant role in its entropy. Consider, for example, the bifurcations of the logistic map. As the value of  $r$  is varied the number of unpredictable trajectories of a given initial value  $x_0$  changes dramatically [4]. It is important to note that these initial parameters must be chosen such that the map both exhibits chaotic behavior. Furthermore, these should be chosen such that they have secure properties that allow it to avoid predictability and improve the pseudorandomness of trajectories.

The notion of being computationally unpredictable is a bit more sophisticated. Its roots lie in complexity theory, and the reader is referred to [5] for a more detailed discussion.

From a practical perspective, cryptographic primitives are deemed secure if they are resistant to known attacks. The two most common forms of cryptanalysis attacks are differential and linear cryptanalysis. Other forms of attacks specific to chaos-based ciphers include trajectory-based, loss of information, and memory attacks.

The probability of a successful differential or linear cryptanalysis attack depends on the statistical attributes of the cipher, or in this case, the internal chaotic map. If it is easy to predict values of the map after any iteration then it is obvious that these attacks will be simple to implement. However, as with any chaotic map, it is computationally difficult to perform such accurate predictions.

For example, consider the logistic map (4). If we partition the phase space of the region of the attractor into  $M$  equal subsets and calculate the number of times a trajectory visits each subset  $m_i$  for a large number of initial values and iterations we obtain a probability distribution of the map. The number of visits associated with each  $m_i$  is the probability  $p_i$  of that space in the phase space. It has been shown that the probability distribution of truly chaotic systems has no dependence on the system's initial value [7], which implies that the probability measure is unchanged by the dynamics of the system (i.e. invariant probability measure). If we build the logistic map with initial parameters such that its corresponding Lyapunov exponent  $\lambda = 4.0$  the probability distribution is given by equation (5). This is the ideal distribution for chaotic maps that are used in chaos-based cryptosystems, as the probability of each phase element occurring after an iteration of the map is the same as any other element. This property increases the difficulty of an effective differential or linear cryptanalysis attack.

$$P(X) = \frac{1}{\pi\sqrt{X(1-X)}} \quad (5)$$

The number of iterations of a chaotic map also impacts the security of chaos-based ciphers. Since chaotic maps are deterministic, the final value can be easily computed given the initial conditions. However, by making the number of iterations for the map unknown, determining the initial conditions based solely on the output trajectory element becomes more difficult.

One must also consider the size of the blocks of data encrypted and decrypted by chaos-based cryptosystems. Larger data blocks means the attacker will have a harder time sifting through the data to find patterns and correlations. However, this improved security comes at the cost of performance, especially when considering chaos-based cryptosystems. Given the complexity of floating point operations on traditional processors and the requirements for the cipher, it might not be feasible to support larger data blocks.

This leads to another aspect of chaotic maps to consider when implementing a chaos-based cipher: the set of elements in the phase space. A direct translation of chaotic systems over the set of real numbers to a running cipher results in the use of high precision floating point operations. This results in very inefficient code. In addition, different processor architectures might handle floating point operations differently depending on their capabilities, which makes them susceptible to reproducibility problems.

## 6. Case Studies

Many different chaos-based ciphers have been designed and proposed in recent years. This section is devoted to four of those cipher designs. Namely, the Simple and Advanced ciphers, Chaotic Feistel cipher, and Rabbit cipher. The internals for each of these cipher designs are discussed along with their relative security properties.

### 6.1 The Simple and Advanced Ciphers

The Simple and Advanced ciphers, proposed by Roskin and Casper [8], are two very basic applications of chaotic maps in block ciphers. They are based on the unpredictability of the logistic map (4). The general idea is to encrypt bytes of plaintext as the final trajectory elements obtained by a variable number of iterations of the logistic map. In this application, both the initial value and the number of iterations of the chaotic map vary.

The complete Simple cipher algorithm is outlined as Algorithm (1).  $f$  is the logistic map (4) with initial parameter  $r = 3.9$  that is used to generate trajectories of some initial value  $x_0$ .  $M_1$  is a mapping function between elements in the key space to the domain of elements in the logistic map (namely, the real interval  $[0, 1)$ ). Similarly,  $M_2$  is the inverse of  $M_1$  in that it maps elements in the domain of  $f$  to the set of integers between 0 and 255.

---

#### Algorithm 1 Simple cipher encryption

---

Generate the key schedule  $\{k_0, k_1, k_2, \dots, k_n\}$  from the 256-bit secret key  $K$

**for**  $i = 0$  to  $n - 1$ , where  $|P| = n - 1$  **do**

$x_0 \leftarrow M_1(k_i)$

$t \leftarrow k_{i+1} + 16$ , where  $t$  is the number of iterations

$x_t \leftarrow f^t(x_0)$ , where  $f$  is the logistic map with  $r = 3.9$

$c_i \leftarrow M_2(x_t) + p_i$

**end for**

---

The security of this cipher comes from the initialization of the chaotic map. Specifically, two successive values in the key schedule are used to generate the initial value for the map ( $x_0$ ) and the number of iterations. If an attacker were to obtain the key schedule, decryption would be simple. It seems that if one does not know the key, it would be difficult to reconstruct the original plaintext from the ciphertext.

To test the security of the cipher, the authors used it to encrypt image data so as to gather a visual measure of the amount of information leakage. They found that the cipher generated data in a periodic fashion. In other words, the pads that are produced by the cipher formed a series that created a pattern of displacement in the ciphertext. The reason for this is that the pad depends entirely upon the key, thus giving the cipher a period equal to the size of the key.

To avoid the periodic behavior of the Simple cipher, the authors implemented a feedback mechanism into its design

so as to vary the pad by both the key values and the output of the previous ciphertext byte. This created a feedback chaining model, as shown in figure (2), and gave the cipher very good statistical properties. It was shown that a change in a single bit in the encryption key changed, on average, 49.6% of the bits in the corresponding ciphertext. Ideally, a change in a single key bit will change 50% of the corresponding ciphertext, so this modification works well.

One problem with this cipher lies in the size of the set of all initial points for the logistic map  $S_I = \{x_0, x_1, x_2, \dots, x_n\}$ . Since the initial points for the logistic map are generated by the mapping  $M_1$  using the iteration keys and previous trajectory values (which are of size  $2^8$  and between the interval  $[0, 1)$ , respectively),  $S_I$  will have an order of  $2^8$ . Theoretically, this makes the initial points and trajectories of the logistic map susceptible to exhaustive search attacks. To work around this shortcoming the authors could have increased the size of the plaintext/ciphertext blocks and iteration keys to 256 bits to match their alleged key size, which would have increased the size of  $S_I$ . However, this also implies that the amount of precision at which these initial points were represented would need to increase. This modification would have an obvious impact on the performance of the cipher.

The authors made an attempt to work around this problem by introducing variability in the number of iterations of the logistic map. The current scheme is to use the sum of an iteration key, previous ciphertext value, and the constant 16 to generate the iteration count. One can deduce that the value 16 was chosen to provide a minimum number of iterations used to generate a pseudorandom value from the chaotic map. However, while it does introduce some pseudorandomness for the number of iterations, this number would provide more security assurances if it was generated from both successive iteration key values. For instance, the values of  $k_i$  and  $k_{i+1}$  could be XOR'd together and the resulting value could be incremented by 16 to produce the final iteration count. This new scheme should introduce more key-dependent variability which would strengthen the overall security of the cipher if the privacy of the key is maintained.

### 6.2 Chaotic Feistel Cipher

One recently proposed chaotic block cipher is the Chaotic Feistel cipher by Masuda et. al. [6]. The general structure of the cipher is shown in figure (3), where each round processes a 128-bit block of data.

For this cipher the authors propose a number of different options for the chaos-based mixing transformation, including a 1-D chaotic map, 2-D cat map, and even 4-D torus map [6]. For each chaotic map, the authors analyzed its security from both a dynamical systems and cryptographic perspective. Specifically, they focused their analysis on the Hamming distance between input and output values and the actual

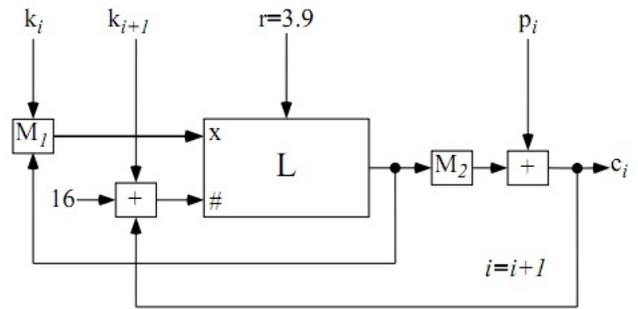


Fig. 2: Block diagram of the Advanced cipher which clearly shows the feedback mechanism used to further randomize the chaotic mappings produced by the logistic map [8].

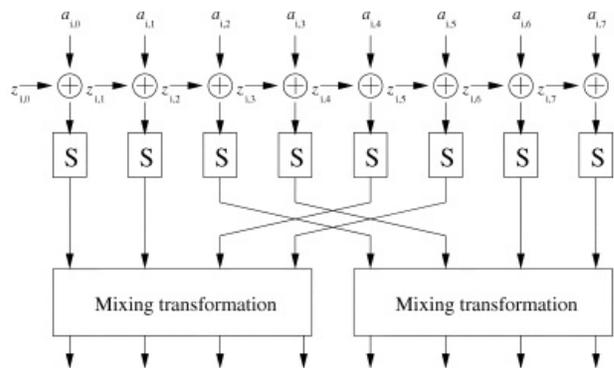


Fig. 3: Block diagram of the Chaotic Feistel cipher proposed in [6]. Each  $a_{i,k}, 0 \leq k \leq 7$  is a byte of plaintext that is fed into the cipher.

Euclidean distance between two elements generated by the chaotic maps.

The cipher also relies on chaos-based S-boxes for its non-linear round transformations, which are built from a custom discretized version of the skew tent map. This map was specifically crafted to guarantee small differential and linear probabilities, which are measures of the cipher's susceptibility to differential and linear cryptanalysis attacks, respectively. Each S-box is a one-to-one transformation defined as follows:

$$S_A(X) = F_A^n(X + 1) - 1 \text{ for } A \in K_S \quad (6)$$

where  $F_A^n$  is the discretized tent map for 256 elements built by the key  $A$  and consisting of  $n$  iterations.  $K_S$  is the set of keys available to build the S-box with sufficient security.

The problem with this S-box is that it is based on a discretized chaotic map that does not necessarily preserve the chaotic behavior of its real-valued counterpart. For this reason, the authors approached its analysis using various assumptions about its lack of algebraic structure and S-box input bytes. Through this analysis they were able to

numerically generate an approximate lower bound for the differential probability of the S-boxes, which effectively eliminated potential key values from  $K$  due to the high differential probability they produced. Specifically, the key space was reduced in size from its original length of 256 to 64, which resulted in the set  $K_S$ .

The small size of this key space is alarming from a security perspective, as block ciphers usually strive to maximize the order of this set. However, given that the differential probability of any possible key from  $K_S$  was less than  $2^{-4}$ , the S-boxes were deemed secure. The authors can argue that the variability in the S-box keys introduces randomness that improves its security, but their relatively small order may still make one wonder about their susceptibility to exhaustive search attacks.

Perhaps the most lacking part in their analysis of the S-boxes lies in their approach to measure the linear probability. The authors only state the use of numerical computation methods to determine the S-boxes' susceptibility to linear cryptanalysis attacks. Their work would have benefited from an algebraic analysis in order to determine the exact correlation between transformations of input and output bytes. Although the S-box substitution is non-linear by definition, poor construction of such a transformation could lead to potential linear correlation attacks on the entire cipher.

### 6.3 Rabbit Cipher

Rabbit is a relatively new stream cipher that was inspired by the random behavior of chaotic maps. Briefly speaking, it is constructed using a chaotic system of coupled non-linear maps that exhibits secure cryptographic properties in its discretized form. It is designed to work with 128-bit data blocks, as both the key and output data are 128 bits in length. Additionally, its internal data structure consists of eight state variables and eight counters. Its design is very similar to the counter mode of operation for traditional block ciphers in that the secret encrypted values can be precomputed and XOR'd with the plaintext for encryption.

The algorithm for the cipher can be broken down into the following four main components: key setup, next state (round) function, counter system, and extraction scheme [9]. The key setup scheme is responsible for initializing the eight individual state variables and counters of the cipher. The mapping between the state variables and counters is a one-to-one correspondence defined by splitting the bits of the key value into 8 individual partitions with some additional manipulations. In order to decrease any statistical correlation between the initial variables and the key, the system is iterated four times using the following next-state function:

$$x_{j,i+1} = g_{j,i} + (g_{j-1,i} \lll 16) + (g_{j-2,i} \lll 16) \quad (7)$$

where,  $g_{j,i}$  is defined as:

$$g_{j,i} = ((x_{j,i} + c_{j,i})^2 \oplus ((x_{j,i} + c_{j,i})^2 \ggg 32)) \quad (8)$$

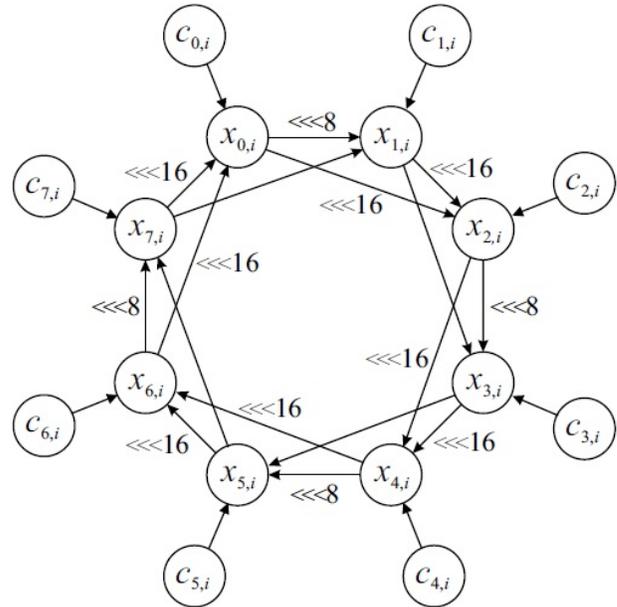


Fig. 4: The next-state function, comprised of eight coupled, non-linear, chaotic maps, used in the Rabbit cipher [9].

Note that all operations are done in modular arithmetic  $2^{32}$  and the index calculations are done in modular arithmetic 8 (since there are 8 state variables). This system of equations can be seen graphically in figure (4).

This map simulates chaotic behavior over the finite domain of integers. However, the cardinality of the field of elements is  $2^{32}$ , which is significantly larger than traditional block cipher fields (i.e.  $GF(2^8)$  used in AES). What is unique is that instead of operating over the domain of real numbers (meaning that implementation requires floating point operations), the domain is scaled up by  $2^{32}$  to translate real numbers into integers.

The counter dynamics are expressed using a system of equations similar to the state variable map. The phase space for the counters is the same as the state variables. The counter values are incremented before each system iteration using predefined constants such as  $0x4D34D34D$  and  $0xD34D34D3$  and carry-over bits from previous iterations. Just as with the next-state function, the addition done to increment the counters is modulo  $2^{32}$ .

After each iteration of the system, the bits of the internal state are extracted and XOR'd with the plaintext to encrypt (or ciphertext to decrypt) the data. This mode of operation is important because no inverse for the next-state function was defined, so it is treated as a one-way function as the cipher operates in a counter-mode.

A thorough security analysis of the operations that set up the secret key (key expansion, system iteration, and counter modification) shows that Rabbit is quite promising. Perhaps the most interesting security property is that the

next-state system iteration function ensures that after only two iterations all of the state bits are affected by all key bits with a probability of approximately 0.5. This value is ideal from a security perspective, but just to be safe the authors have chosen to give a safety margin of four iterations to increase this probability.

Additionally, the counter modification process is very difficult to invert without knowledge of the internal state variables. However, there is the possibility that counter values will be repeated for different keys (periodic behavior exhibiting a pattern), which is detrimental to the overall security of the cipher. Since the counter space is very large, predicting values of the counter is relatively easy since the increment function is based on simple addition operations. In particular, the least significant bit of each counter value has a probability of 1.0 to change, whereas the most significant bit has a probability of  $2^{-255}$  of changing. Despite these drawbacks, the fact that the counter bits carry over into subsequent increment operations means that each bit will have an equal period length.

Further analysis work revealed that the cipher held up well against algebraic and statistical attacks. Their algebraic analysis examined the Hamming distance for the  $g$  function. Through simple manipulation of the individual bytes for  $g$ , the authors were able to determine that each byte of  $y$  in  $g(y)$  has an entropy of approximately 7.99, meaning an acceptable level of diffusion was obtained. Unfortunately, the influence of the counter value in  $g$  was ignored. If these bytes were included in the byte-wise manipulation of the equation, the dependence results would have been slightly different and more complex, which would almost certainly lead to different results for the Hamming distance.

Also, the authors approached the security of Rabbit mainly from a cryptographic viewpoint, not a dynamical systems one. Simple numerical tests could have been performed to approximate the discrete Lyapunov exponent for the entire chaotic system, thus indicating the actual measure of trajectory divergence for the next-state function. However, given the fact that the chaotic system is comprised of multiple non-linear maps, analyzing the Euclidean distance for trajectories of the 1-D system becomes a matter of measuring the distance for all possible element pairs of the individual maps. Also, the size of the phase space ( $2^{32}$ ) makes measuring Euclidean divergence difficult, but still an important part of the analysis. While this would certainly increase the complexity of the security analysis, it might reveal characteristics about the system not touched upon by algebraic analysis.

## 7. Conclusion

It has been argued that a lack of definition for discrete chaos in finite domains based on a discretized version of the Lyapunov exponent plays a large role in the development and security of chaos-based symmetric key ciphers. Many

of the modern chaos-based symmetric key cryptosystems suffer from lack of truly chaotic behavior when their internal chaotic maps are discretized for implementation. Furthermore, the inefficiencies of these implementations have had a significant impact on both the performance and design of chaos-based ciphers. Manipulating elements in real-valued systems consists of expensive operations that significantly impact the overall efficiency of the cipher.

Chaos-based ciphers also suffer from a lack of thorough security analysis efforts that critique their design and implementation from both a dynamical systems and cryptographic perspective. It is not enough to consider one paradigm of security for these ciphers, as flaws in one may be enough to reveal a fundamental weakness in the other. Furthermore, analysis efforts should consist of both numerical and algebraic analysis techniques. Given the difficulty of implementing discrete chaos in cryptosystems, both forms of analysis are necessary to uncover potential weaknesses that may lead to successful differential or linear cryptanalysis attacks.

Overall, however, there are certainly elements of chaos theory that make it theoretically applicable to cryptography. For this reason, there has been and will probably continue to be significant research done in chaos-based symmetric key cryptosystems. However, given the loose connection between these two fields thus far, it is difficult to tell if these research efforts will be successful when compared to today's standardized cryptographic primitives and the emerging usage of elliptic curves and other number theoretical concepts in cryptography. Perhaps as chaos theory evolves this connection will become clearer and pave the way for more appropriate cryptography applications. Until then, however, traditional number theory cryptosystems will continue to lead the way into the future.

## References

- [1] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, pp. 130–141, 1963.
- [2] E. W. Weisstein. Chaos. From MathWorld, A Wolfram Web Resource. [Online]. Available: <http://mathworld.wolfram.com/Chaos.html>
- [3] I. P. Cornfeld, S. V. Fomin, and Y. G. Sinai, "Ergodic theory," *Springer*, 1982.
- [4] E. W. Weisstein. Logistic map. From MathWorld, A Wolfram Web Resource. [Online]. Available: <http://mathworld.wolfram.com/LogisticMap.html>
- [5] L. Kocarev, "Chaos-based cryptography: A brief overview," *Circuits and Systems Magazine, IEEE*, vol. 1, pp. 6 – 21, 2001.
- [6] N. Masuda, G. Jakimoski, K. Aihara, and L. Kocarev, "Chaotic block ciphers: From theory to practical algorithms," *IEEE: Transactions on Circuits and Systems*, vol. 53, pp. 1341 – 1352, 2006.
- [7] L. Shujun, M. Xuanqin, and C. Yuanlong, "Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography," in *Progress in Cryptology Ü INDOCRYPT 2001*, ser. Lecture Notes in Computer Science, C. Rangan and C. Ding, Eds. Springer Berlin / Heidelberg, 2001, vol. 2247, pp. 316–329.
- [8] K. Roskin and J. Casper, "From chaos to cryptography."
- [9] M. Boesgaard, M. Vesterager, T. Pedersen, J. Christiansen, and O. Scavenius, "Rabbit: A new high-performance stream cipher," *Proc. Fast Software Encryption*, vol. 2887, 2003.

# Secure Processing and Delivery of Medical Images for Patient Information Protection

Ming Yang<sup>1</sup>, Lei Chen<sup>2</sup>, Shengli Yuan<sup>3</sup>, and Wen-Chen Hu<sup>4</sup>

<sup>1</sup>School of Computing and Software Engineering, Southern Polytechnic State University, Marietta, GA, USA

<sup>2</sup>Department of Computer Science, Sam Houston State University, Huntsville, TX, USA

<sup>3</sup>Department of Computer and Mathematical Sciences, University of Houston Downtown, Houston, TX, USA

<sup>4</sup>Department of Computer Science, University of North Dakota, Grand Forks, ND, USA

**Abstract** - *In the delivery of medical imaging (such as X-ray, MRI) for remote diagnosis, the protection of the security and privacy of patient's information is extremely important. As conventional E-mail delivery is considered insecure, nowadays, people send medical images to a remote location using secure shared network storage space over IP protocol. While this is more reliable than traditional E-mail delivery, it introduces higher costs and dedicated devices. In this study, we propose a reliable and economical E-mail delivery approach which ensures the security and privacy of imaging contents and patient information. In the proposed methodology, patient information within the medical images (host image) is encrypted and embedded. Consequently, confidential data will not be visually available to unauthorized personnel. In order to further ensure the secure delivery of the medical images via E-mail over public network such as the Internet, the proposed system utilizes non-web-based Secure E-mail transmission using the Enigmail security extension installed on E-mail client software Mozilla Thunderbird. Thunderbird, Enigmail security extension, and Enigmail's essential component GNU Privacy Guard (GnuPG), are all open source and freely available online. With this system, any medical image that requires electronic transmission will have the patient's information protected, and will be readily available immediately upon the delivery at the destination. This system is an economical and reliable alternative to the IP-based delivery.*

**Keywords:** Encryption, Privacy, HIPAA, Information Hiding.

## 1. Introduction

The Health Insurance Portability and Accountability Act (HIPAA) [1] requires that medical providers and insurance companies implement procedures and policies to protect patient's medical information. Areas to be specifically

addressed include ensuring that confidential data is secured during electronic transmission, and that access is limited only to authorized personnel. Today, as remote diagnosis is becoming increasingly popular, medical images, such as X-ray or MRI, often need to be delivered from one location to another. This imposes new challenges that need to be faced:

- (1) The patient information is usually printed in the corner of the medical images for viewing. As a result, it is easily accessible to anyone and may be intercepted by a third party in the course of electronic transmission.
- (2) For scenarios such as medical imaging research, the patient information should not be accessible either.
- (3) Traditionally, medical images are delivered through printed films or in burned CDs. This is neither secure nor reliable.
- (4) Nowadays, people start to send medical images to a remote location over IP protocol, or use shared network storage space. This is more reliable than traditional approach, but it introduces higher costs and needs dedicated devices.
- (5) Traditional E-mail transmission of medical images is generally considered to be insecure.

In order to address these issues, we have proposed a reliable and inexpensive approach to ensure the electronic delivery of medical images while securing the confidentiality of imaging contents and patient information. We have developed an information hiding methodology that makes use of the RSA encryption algorithm and a Discrete Cosine Transform (DCT) based hiding technique. As a result, patient information will not be visually available to unauthorized personnel. To ensure the secure delivery of the medical images, our proposed system utilizes non-web-based Secure E-mail Transmission using Mozilla Thunderbird with its security extension Enigmail and the core security component GNU Privacy Guard (GnuPG), all of which are open source and freely available on the

Internet. Using this approach, any medical image that requires electronic transmission will have the patient's information protected, and will be readily available immediately upon the delivery at the destination. Secure E-mail delivery is also feasible through this approach.

## 2. System Overview

In the proposed system, patient information is not automatically visibly displayed in the corner of the medical image. Instead, this information is first encoded using ASCII character-encoding scheme and encrypted into a non-recognizable format using the RSA encryption algorithm.

Next the patient information is further secured by embedding it within a section of the image that is outside the Region-Of-Interest (ROI) [2]. This ensures that the encoded and encrypted information is embedded in a location that will not affect the image quality and further diagnosis. The area outside the ROI is located by using image segmentation techniques as discussed in Section III.A.

After the area outside of the ROI is located, the patient information (already encoded and encrypted) is embedded using a DCT domain methodology. This information hiding

algorithm is robust enough that further attacks (including cropping, noise, lossy compression, etc.) will not remove the embedded information. This methodology effectively and securely protects patient information in situations of electronic transmission and medical imaging research.

The security of medical images and patient information is further reinforced using the free open source Enigmail [7] security extension, with its core component GnuPG [8], installed on E-mail client software Mozilla Thunderbird. All these three software components and applications will ensure the secure delivery of the medical images through E-mail transmissions.

The image can be viewed in one of two forms. If the viewer does not have the authority to access the patient's personal information, for example a medical or computer researcher (or network hacker for that matter), the image is viewed with no data displayed in connection to the image. On the other hand, if the viewer, such as the patient's doctor, has the authority to access the confidential information, it can then be extracted, decrypted, decoded, and displayed upon the image with the input of the correct encryption/decryption key. The above procedure can also be combined with a fragile watermark to validate data integrity. This approach is illustrated in Figure 1.

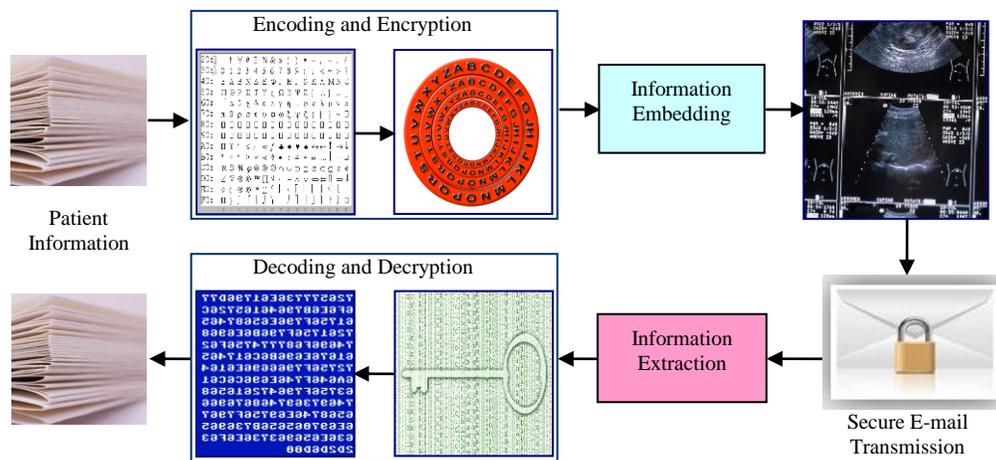


Figure 1. Flow Chart of Proposed Methodology

## 3. Implementation

In order to implement the proposed methodology, we first need to use image segmentation to identify the non-ROI region for information embedding, so that the embedded information will not affect the quality of the critical portion of the medical images. Next, patient information will be encoded, encrypted, and embedded for E-mail transmission. The patient information security system was implemented using MATLAB, a high-level programming language and

interactive numerical computing environment. MATLAB has a wide variety of image processing capabilities and can process DICOM, BMP, JPG as well as other image formats.

### 3.1 Image Segmentation

Image segmentation is the process of dividing an image into sections, regions, or parts [2]. This process has numerous applications, such as automated inspection. Gonzalez gives the example that in the automated

inspection of electronic assemblies, image segmentation is used to find defects, such as a missing or broken path [2]. In respect to our project, we aim to identify the ROI, or the location where the actual picture is on the medical image. For example, if we have an X-ray of an elbow, our region of interest would be the elbow, NOT the black space surrounding it. Clearly, image segmentation is a vital part of this project. This process identifies the region of interest of an image, and draws a boundary within which the patient information should not be placed. This was achieved using MATLAB's built-in contour functions, and later implemented using a Java program. In the embedding procedure a simple image segmentation algorithm was employed to identify the ROI. Figure 2 is an x-ray of a skull that has been analyzed using image segmentation and has only the contour lines shown.

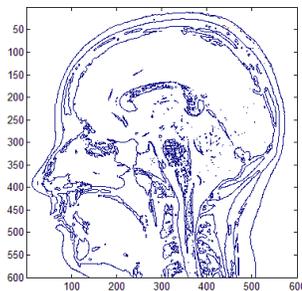


Figure 2. Image with Region of Interest (ROI) Boundaries

After image segmentation, the encoded/encrypted patient information is embedded in the portion of the background area that was determined outside of the ROI to preserve the quality of the host medical image [4].

### 3.2 Information Encryption

Patient data, delimited by commas and spaces, was read in from a test file for the first version of the system. In a later version the data was entered interactively by the user through a Graphical User Interface (GUI), written in Java and called by MATLAB.

The procedure to convert the text data to ASCII format in MATLAB resulted in seven-bit character strings instead of the expected eight bits. These strings then need to be broken apart and individually converted back into integer data types in order to perform the necessary mathematical operations for encryption and embedding.

The RSA encryption method was used to encrypt the patient's information. This particular method was chosen due to the simplicity of its algorithm. RSA is an asymmetric or public key algorithm, meaning it has both a public key and a private key [5]. The advantage of an asymmetric encryption lies in its higher level of security.

### 3.3 Information Embedding

Throughout the ages various methods have been devised to conceal information in transit. Tactics in previous times ranged from tattooing the message on a shaved head then waiting for the hair to re-grow before sending the message ([4]) to placing microfiche with the information under the postage stamp on a letter. With the creation of the Internet and other electronic data transmission mediums, steganography or the art of hiding information, has become even more important and commonplace.

Information can be hidden with success in text, image, audio/image, and protocol file formats. For image/video information hiding, there are two main groups of techniques: spatial domain algorithms and transform domain algorithms. Spatial domain algorithms generally involve manipulation of pixel intensity. Lossless image formats are most suited for spatial domain techniques [4]. The most well-known technique of information hiding in the image domain is Least Significant Bit (LSB) algorithm. Frequency domain algorithms try to modify the coefficients in the transform domain, which is more robust against transformation-based lossy compression [4].

### 3.4 Information Embedding and Extraction Algorithms

A high bitrate transform domain information hiding algorithm is designed to enable data embedding. In the proposed algorithm, a single bit is hidden within each 4x4 DCT coefficient block by means of vector quantization. Low-frequency coefficients are chosen for information hiding due to their relatively large amplitudes and the corresponding small step sizes in the quantization matrix [6].

The embedding algorithm is described in the following:

- (1) DCT (4x4) transform of the original image;
- (2) Scan the 4x4 DCT block along Zig-Zag scanning path;
- (3) Convert the 8 low-frequency coefficients to an 1-D vector;

$$(4) V: \text{ the 1-D vector } V = (c_0, c_1, c_2, \dots, c_6, c_7)$$

T: the step size for vector quantization

$|V|$ : the norm of vector V

$[\ ]$ : round-off operation

$$l = |V| = \sqrt{\sum_{i=0}^{15} c_i^2} : (\text{norm of } V)$$

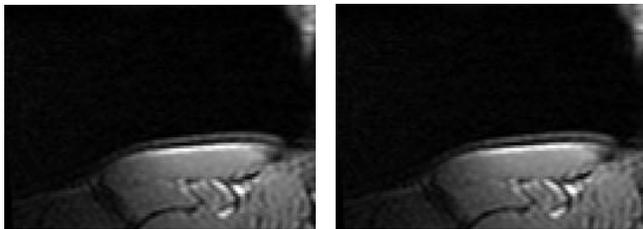
$$l_T = \left\lfloor \frac{|V|}{T} \right\rfloor = \left\lfloor \frac{\sqrt{\sum_{i=0}^{15} c_i^2}}{T} \right\rfloor : \text{(quantized norm of } V)$$

- (5) One single bit is embedded by modifying  $l_T$ :  
 $l_T' = l_T \pm 0.25$  (+0.25 to embed 1, -0.25 to embed 0);
- (6)  $l' = l_T' * T$ ,  $V' = \frac{l'}{l} * V$  ( $V'$  is the modified vector);
- (7) Place the vector  $V'$  back to its original location in the 4x4 DCT block;
- (8) Repeat the same operation for each 4x4 DCT block until all the information bits have been embedded.

The information retrieval algorithm is the following:

- (1) DCT transform the stego-image (image with embedded information);
- (2) For each 4x4 DCT block, scan the coefficients along Zig-Zag scanning path;
- (3) Pick up the 8 low-frequency coefficients and convert them to a 1-D vector  $V''$ ;
- (4) The norm of  $V''$  is:  $l'' = |V''|$
- (5) The quantized norm is:  $l_T'' = \frac{l''}{T} = \frac{|V''|}{T}$
- (6)  $I = l_T'' - \lfloor l_T'' \rfloor$
- (7) If  $I \geq 0$ , then 1 is extracted as the information bit;  
 else ( $I < 0$ ), then 0 is extracted as the information bit.
- (8) Repeat the same operation to each 4x4 DCT block until all the information bits have been extracted.

With the embedding algorithm, the quality of the host image will not be visually degraded (Figure 3). Also, the hidden information can be extracted without the presence of



original image. This feature is extremely important in many applications. The proposed algorithm is also very robust to lossy compression, according to the experimental results.

Figure 3. Comparison of Original Image (left) and Stego-Image (right)

After the image segmentation, encoding, encryption, embedding procedures are completed, the medical image is ready for transmission. The transmission of the medical image with a secure E-mail approach will be discussed in Section IV.

With the supply of the correct decryption key, the extraction, decryption, and decoding of the data are simply the reverse of the embedding/encryption/ encoding procedures, with the addition of the display of the patient data below the image in the receiver's GUI. A copy of the program was placed on a remote computer and the full procedure was tested. The data file was encrypted and embedded into the image and transmitted via E-mail. The image was retrieved at the second computer, extracted, and decoded.

#### 4. Secure E-mail Transmission

Not only is patient's personal information confidential and therefore prepared and embedded nicely using the proposed algorithm, the entire medical image also requires the guarantee of confidentiality and integrity on the path between sender and receiver. All of these security goals are achieved with the help from GNU Privacy Guard (GnuPG) [8], the core security component of Enigmail [7] extension installed on Thunderbird [11] E-mail client, as shown in Figure 4.



Figure 4. Securing Medical Images as E-mail Attachments using Thunderbird, Enigmail, and GnuPG

GnuPG, as described in OpenPGP standard [9] that it follows, makes use of both symmetric-key and public-key encryption to provide confidentiality. A unique random symmetric session key  $S$  created at the sender side, e.g. Dr. A, is used to encrypt the E-mail object content which consists of the message and medical images (with patient information embedded) as attachments. In order for the receiver, Dr. B, to be able to obtain key  $S$  in a secure manner and then decrypt the message and image attachments,  $S$  is encrypted at the sender using Dr. B's public key  $KB+$ , which can be posted on a public key server or sent directly to a sender via plaintext E-mails. To preserve the integrity of the E-mail message and the medical images, a message digest  $M$  is created using hash function SHA-1, then digitally signed by Dr. A with his private key  $KA-$  and attached to the E-mail object. As soon as the entire

E-mail arrives at Dr. B's computer, she can apply Dr. A's public key  $KA^+$  to what she has received. If the received message digest turns out to be the same as the SHA-1 hash value of the received E-mail object content, it proves that the E-mail message and all attachments have not been modified during transmission.

The stable version 1.0.4 version of GnuPG employed in our proposed system follows the Advanced Encryption Standard (AES) with 256-bit key. According to Bruce Schneier [10], current reported attacks can only break up to 11 rounds of the 14 rounds of AES-256. Therefore, the confidentiality of both E-mail messages and medical image attachments is ensured. Enigmail allows Thunderbird works seamlessly with GnuPG for protecting medical images in our system. The system uses Thunderbird for E-mails client software for several reasons: (1) it is free which adds no extra cost to the already expensive medical systems; (2) it supports most current operating systems; (3) it is open source that many plug-ins and extensions, especially those related to security, are freely available on the Internet.

## 5. Conclusion

In this study, we proposed a reliable and economical approach to ensure the electronic delivery of medical images while securing the security and privacy of image contents and patient information. The proposed methodology utilizes data encryption and high bitrate information hiding to ensure patient information security. It also makes use of secure E-mail transmission to ensure the secure delivery of medical images. It is a secure alternative of the existing medical imaging delivery approaches, such as server-to-server delivery through IP protocol. With this system, any medical image that requires electronic transmission will have the patient's information protected, and will be readily available immediately upon its delivery at the destination.

## 6. References

[1] "Health Insurance Portability and Accountability Act (HIPAA) and Its Impact on IT Security," Regulatory Compliance Series 3 of 6, Apani Networks White Paper Compliance Series. May 12, 2005. <http://www.apani.com>.

[2] R. C. Gonzalez, and R. E. Woods. "Digital Image Processing", Upper Saddle River: Prentice-Hall, 2002.

[3] D. Kundur, "Implications for high capacity data hiding in the presence of lossy compression", Proceeding of International Conference on Information Technology: Coding and Computing, March, 2000, pp. 16-21.

[4] T. Morkel, J.H.P. Eloff, and M.S. Olivier, "An Overview of Image Steganography," Proceedings of the Fifth Annual Information Security South Africa Conference. (ISSA2005), Sandton, South Africa, June/July 2005.

[5] M. Yang, S. Li, and N. Bourbakis, "Data-Image-Video Encryption", IEEE Potentials Magazine, Aug/Sept. 2004, pp.28-34.

[6] M. Yang, and N. Bourbakis, "A High Bitrate Multimedia Information Hiding Algorithm in DCT Domain", Proceeding of World Conference of Integrated Design and Process Technology (IDPT 2005), Beijing, China, June 13th-17th, 2005.

[7] "Enigmail Quickstart Guide", retrieved from <http://enigmail.mozdev.org/documentation/quickstart.php.html>, March 23, 2011

[8] "GnuPG", retrieved from <http://www.gnupg.org/documentation/index.en.html>, March 23, 2011

[9] "OpenPGP Message Format", RFC 4880, retrieved from <http://tools.ietf.org/html/rfc4880>, March 23, 2011

[10] "Schneier on Security", Bruce Schneier, retrieved from [http://www.schneier.com/blog/archives/2009/07/another\\_new\\_aes.html](http://www.schneier.com/blog/archives/2009/07/another_new_aes.html), March 23, 2011

[11] "Mozilla Thunderbird", retrieved from [http://en.wikipedia.org/wiki/Mozilla\\_Thunderbird#Cross-platform\\_support](http://en.wikipedia.org/wiki/Mozilla_Thunderbird#Cross-platform_support), March 23, 2011

# A Secure permutation routing protocol in multi-hop wireless sensor networks

Hicham Lakhlef, Jean Frédéric Myoupo

Université de Picardie-Jules Verne, UFR Sciences, 33 rue Saint Leu, 80039 Amiens France

{ lakhlef.hicham@yahoo.fr, jean-frederic.myoupo@u-picardie.fr }

*Abstract: A growing number of researches is done on the permutation routing problem in wireless networks, however, none of these studies do address the problems of security in the permutation routing in wireless sensor networks. The permutation routing problem in a military application is the fact that each soldier has items (information), that not concerned by him, and perhaps data which concerned, that is, in such applications and for confidential reasons, during the deployment, a soldier may hold items which are not necessary its own. The soldier to accomplish his task must receive its items from other soldiers in the network where it belongs. The necessity and the importance of secure permutation routing appear well when the permutation is a military application. The aspects of security that we deal with in this paper are not merely the authenticity, confidentiality, integrity, and non-repudiation, but we also show how we secure the partitioning into clusters and cliques in order to get consistency clusters and cliques.*

*Key Works:* Wireless Sensor-Actuator Networks,  
Permutation Routing, Security

## 1. INTRODUCTION

A sensor network is composed of a large number of sensor nodes which are densely deployed in a area. WSN can be deployed to provide continuous surveillance over an area of interest referred to as a sensor field [7, 21]. Wireless sensor nodes perform collaborative work [9] via wireless communication channels to retrieve information about targets that appear in the sensor field or to exchange some information. Higher-level decision making can then be carried out based on the information received from the sensor nodes. These networks can be deployed in inhospitable terrain or in hostile environments to provide continuous monitoring and information [9], or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants [1, 9, [21]note that in sensor network energy are limited and we must to minimize as possible the number of the broadcast in order to increase the lifetime of the system .

There are two types of wireless networks: Single hop wireless networks in which each station can transmit or communicate directly with any other station. All the stations use the same channel to communicate, and the message broadcast by one of the stations on the common channel is simultaneously heard by all other stations. In the multi-hop wireless networks intermediate nodes are used to route message from the source to the destination.

*Permutation Problem:* Consider a MANET  $(n, p)$  of  $p$  stations with  $n$  items saved on it. Each item has a unique destination which is one of the  $p$  stations. Each item has a unique destination which is one of the  $p$  stations. Each station has a local memory of size  $p/n$  in which  $n/p$  items are stored. It is important to note that in general, some of the  $n/p$  items stored in the station, say  $i$ , have not  $i$  as destination station. And even, it can happen that none of these  $n/p$  items belongs to it. In the other hand, the situation in which initially all items in  $i$  belong to  $i$  can also occur. The permutation routing problem is to route the items in such a way that for all  $i, 1 \leq i \leq p$ , station  $i$  contains all its own items.

A large variety of permutation routing protocols in a single-hop Network are known to day. These permutation routing protocols assume that the network are a single Hop Ad-Hoc Network, hence there is always a path connected by wireless links between a source and the destination. However, these varieties of methods are not adapted in the case of multi-hop Ad Hoc Networks. One way to solve this problem is to partition nodes into clusters where principal node in each cluster, called clusterhead, is responsible for routing items.

In reality, the change of information in wireless sensor networks is not secure, and the malicious node can do all tricks to prevent a normal run of permutation routing protocol. It can change (active attack) or intercept (passive attack) the information, and if a malicious node intercepts all information of a node say  $j$ , it can know the behavior of  $j$ , and thus the consequences in a military application for example will be very serious. Other behavior of the malicious node with the same consequences can occur with an active attack, when the malicious node modifies one or more information destined to node  $j$ . It can also make an attack to break the normal run of a clustering algorithm carried out by the sensors.

### 1.1. State of the art:

The first algorithm that treats the permutation routing in multi-hop wireless networks [3], needs

$$(k+1)n + O\left(\left|HUB_{\max}\right|\right) + k^2 + k$$

Broadcast rounds in the worse case. Where  $n$  is the number of the data items stored in the network,  $p$  is the number of sensors,  $|HUB_{\max}|$  is the number of sensors in the clique of maximum size and  $k$  is the number of cliques after the first

clustering. The number of broadcast rounds was improved to  $3n + 6 \log_2 k$  in protocol in [14].

The number of studies specifically targeted to permutation routing in single hop wireless networks has grown significantly. It is shown in [17] that the permutation routing of  $n$  items saved on wireless sensor network of  $p$  stations and  $k$  channels with  $k < p$ , can be carried out efficiently if  $k < (p)^{1/2}$ . Datta in [4] derived a fault tolerant permutation routing protocol of  $n$  items saved on mobile Ad-hoc network of  $p$  stations and  $k$  channels MANET( $n, p, k$ ) for short. He also assumed that in the presence of faulty stations some data items are lost. We came out with our work in [12] presenting a fault tolerant protocol which avoids the loss of items. The first energy-efficient permutation routing appeared in [18]. A more efficient energy-efficient permutation routing protocol was presented in [5]. In [23] Walls et al. propose an optimal permutation routing on mesh networks. Another approach as an application of an initialization algorithm appeared in [11]. All these approaches assume that the WSN is a single hop networks and none of these protocol is secure as in [13, 16].

**1.2. Our contribution:**

We consider a WSN ( $n, p$ ) with  $n$  items,  $p$  stations. We first propose to partition the network into single-hop clusters also named *cliques* with a secure algorithm. Secondly, we run a secure local permutation routing to broadcast items to their local destinations in each clique. Next we partition the cluster-heads of cliques with the hierarchical clustering technique but this hierarchical algorithm is not secure, we define the possible attacks on this clustering protocol. Next we propose the solutions to overcome these attacks. We show how the outgoing items can be routed securely to their final destination cliques.

The rest of this paper is organized as follows: section 2 we define the preliminaries and some definitions, section 3 we present an protocol of permutation routing in multi hops wireless sensors network with an single channel, after an protocol without collision and conflict in the channels, using the maximal capacity of the network this is obtained by defining an optimal colouring algorithm, in section 4 we present the experimental results, and in the section 5 we conclude and we define the future works.

**2. PELIMINARIES**

We assume that each station has a local clock which keeps synchronous time by interfacing with a Global Positioning System (GPS). Time is divided into slots and all packet transmissions take place at slot boundaries in WSN ( $n, p$ ).

As in [3, 4, 5, 13, 16, 17, 18], we suppose that the  $n$  items denoted  $a_1, a_2, \dots, a_n$  are saved on a WSN( $n, p$ ) such that for every  $i, 1 \leq i \leq p$ , station  $i$  stores the items. Each item has a unique destination station. It is important to note that hereafter a sensor knows the destination of items it holds. In fact the data item it holds is a couple  $S(s, d)$ , where  $s$  is the

real data item belonging to sensor source and  $d$  is the sensor destination. For every sensor  $h, 1 \leq h \leq p$ , let  $h_d$  be the set of items whose destination is sensor  $h$ .

The permutation routing problem is to route the items in such a way that for all  $h, 1 \leq h \leq p$ , sensor  $h$  contains all the items in  $h_d$ . Consequently, each  $h_d$  must contain exactly  $n/p$  items.

**1.1. A clustering scheme in cliques**

Our approach uses the secure clustering protocol from [22] to partition network into clusters (cliques). The figures 1. a and 1.b blow illustrate a network in which each clique is a single hop sub network. Each clique is a single hop network.

Figure 1. a: network with 11 sensors

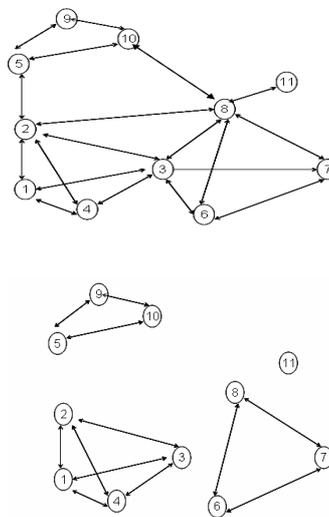
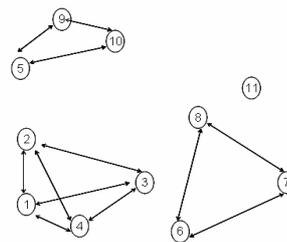


Figure 1. b: Resulting cluster formation in cliques



**2.2 Hierarchical Control Clustering**

Banerjee and Khuller [2] proposed a clustering algorithm for multi-hop sensor networks.

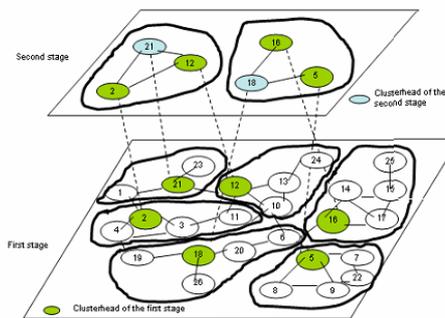


Figure 2: Hierarchical control clustering with k=3

Their clustering scheme is motivated by the need to generate an applicable hierarchy for multi-hop wireless environment.

Their method yields a multi-stage clustering. To reach their goal they construct a breath first search tree such that each level is composed of clusterheads of the immediate low level. These Clusters are by definition disjointed and the number of the nodes in a cluster remains between  $k$  and  $2k$  for some integer  $k$ . Figure 2 shows a hierarchical clustering of a network of 25 sensors with  $k=3$ .

### 1.3. Problem Statement

**Objective:** The objective of our paper is to secure the information sent and to secure the clustering algorithms partitioning a sensor network into mutually disjoint cliques or clusters, so that all nodes in the same cliques or the same clusters can communicate with each other. We denote the view of cluster for node  $i$  as  $C_i$ . For brevity, we call  $C_i$  as the cluster of node  $i$ . We call a node a *normal node* if it follows our protocol. Otherwise, it is a *malicious node*. We would like to guarantee that all normal nodes have consistent clusters, as reflected by the following cluster agreement property. *Cluster agreement* for a normal node  $i$  is defined as:

**Definition 1: (Cluster Agreement).** For each sensor  $j \in C_i$ ,  $C_j = C_i$ .

**Definition 1** implies that for each normal node  $j \notin C_i$ ,  $i \notin C_j$  must hold. That is, each normal node belongs to only one cluster. Cluster agreement is broken if *cluster Inconsistency* is detected. For node  $i$ , cluster inconsistency is defined as:

**Definition 2: (Cluster inconsistency).** There exists a node  $j \in C_i$  such that  $C_i \neq C_j$ .

It is desirable that each node find a cluster as large as possible. We do not consider trivial solutions with which each node forms a cluster that only includes itself.

#### Assumptions:

We assume that each node in  $WSN(n,p)$  share unique pairwise private keys used for digital signatures with other nodes, and unique private key shared by all nodes in  $WSN(n,p)$  this keys is used in phase 2 and phase 4, and All unicast messages exchanged between nodes are authenticated with the key shared between the two nodes. We use the low-end sensor nodes (e.g., MICA2 motes with 8-bit processors) defined in the recent investigations [8, 15] where it is shown that sensor can use public keys to perform cryptographic operations. Moreover, recent development of sensor platforms such as Intel motes uses more advanced hardware, and can perform public key cryptographic operations efficiently.

We use a combination of  $\mu$ TESLA [20] and digital signature to authenticate broadcast messages. We use digital signatures when non-repudiation is necessary and  $\mu$ TESLA for efficient broadcast authentication in other cases. We assume the clocks of the normal nodes are loosely synchronized, as required by

$\mu$ TESLA. We also assume the public keys used by the sensor nodes are properly authenticated. One approach to ensure this is to issue to each node a certificate for its public key so that other nodes can validate the node's public key by verifying this certificate. Moreover, the malicious nodes may launch Sybil attacks [6] or Wormhole attacks [10]. However, we assume these two kinds of attacks can be detected by using the techniques proposed in [19] and [10], respectively.

Now we propose to secure our algorithm phase by phase.

## 3. SECURE PERMUTATION ROUTING PROTOCOL

### 3.1 Phase 1:

We partition the sensors into cliques according to the secure protocol secure in [22], with this protocol we can't find Inconsistency Cliques. This algorithm gives  $k$  cliques, and the  $k$  clusterheads  $CH_{clique-i}$  (i.e. for each clique a clusterhead), the purpose of this phase is to obtain sub-WSNs single hop sub-WSNs( $n_i, p_i$ ),  $0 < i \leq k$ , that perform data exchange between them and permutation routing as in phase2.

After, we consider the clusterhead graph named  $G_{KH}$ , a link between sensor  $A$  and sensor  $B$  (cluster-heads  $A$  and  $B$ ) is possible if there is at least one direct link between sensor in clique of  $A$  and other in clique of  $B$ . The role of clusterhead  $CH_{clique-i}$  is to collect all data items whose destination sensors are not in clique  $i$ . We Note  $HUB(i)$  the clique  $i$ , and  $HUB_{max}$  the clique that contains the maximum number of sensors.

### 3.2 Phase2:

In this phase each item  $a_1, a_2, \dots, a_n$  in  $WSN(n, p)$  is a secure hash value signed with the with key shared in  $WSN(n, p)$ . We run in each clique the permutation routing protocol as the single-channel-routing in [17], i.e., for the wireless network single hop.

Once more, the idea of this phase is similar to the protocol single-channel-routing [17], where each sensor broadcasts its items one by one and every time unit. Clearly if in a slot  $t_0$  a sensor  $i$  broadcasts an item, the sensor, say  $j$ , whose identity matches the destination of the item being broadcasted copies it in its local memory. At  $t_0+1$   $j$  broadcasts an acknowledgment. If no sensor of the clique is the destination of the broadcasted data item then no action is taken and each sensor of the clique knows that the item is an outgoing item. Therefore each sensor counts the number of outgoing items it holds. Note that the clusterhead has the IDs of all the residents of its cluster. The broadcasts are carried out on cliques. So the clique with the great number of sensors should help to estimate the total broadcast rounds of this phase. At the end of this phase all data items that do not belong to the sensors of a clique are saved on the sensors of the clique. The goal now is to route these outgoing items to their final destinations.

This phase is carried in  $(n/p)|HUB_{max}|$ , because the phase processed in parallel and the clique that has the great number of sensor estimate the maximum number of

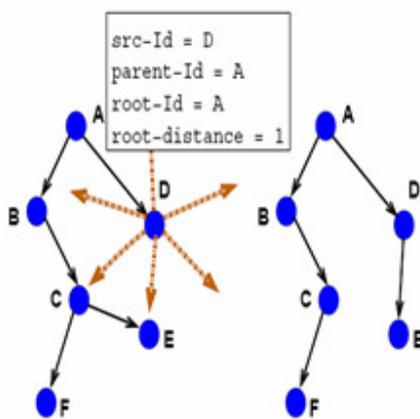
broadcast . Since  $HUB_{max} \leq p$ ,  $(n/p) \leq n$ . Therefore the number of broadcast rounds of this phase cannot exceed  $n$ .

**3.3 Phase 3:**

In this phase we use the clustering algorithm in [2], which is not a secure algorithm. We recall the principle of this algorithm and we define the possible attacks and we propose solutions to counter them.. It takes as parameters a graph and an integer  $C$ , and it generates clusters with size greater than  $C$  and lower than  $2C$ . It generates a single cluster if the size of the graph in terms of number of nodes is  $<2C$ .

The distributed version of this algorithm is decomposed into two steps: *Tree-discovery* and *Cluster-Formation*:

*Step 1:* Tree-discovery needs five information on each node  $\{src-id, parent-Id, root-Id, root-seq-no, root-distance\}$ . Each node sends a tree discovery beacon which indicates its shortest hop-distance to the root. On receiving this beacon, the node discovers a shorter path to the root. it updates its hop-distance to the root according to the information in the beacon , and updates its parent as in figure 3: node E is originally at distance 3 from the root A, E receives a beacon from node D, at distance 1 from the root and consequently E chooses D to be its new parent . This decreases the distance of E from the root to 2



**Figure 3. Tree-discovery of [2]**

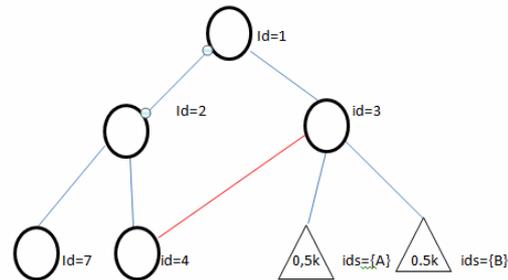
*Step 2:* In *cluster formation* starting with the leaves each node sends the number (counting himself) of sensor nodes that exists in the sub-tree rooted at him. It also sends this corresponding sub tree in the same message  $\{ subtree-size, node-adjacency\}$ . If this number is greater than  $C$  it initializes a cluster. If the size of the tree is  $<2C$  the algorithm gives a single cluster.

**Example:** Set  $C<3$ . In figure 4  $<8>$  and  $<9>$  are sub-trees rooted à 8 and 9 respectively.  $<8>$  and  $<9>$  contains  $0.5C$  nodes each. This figure shows an example of formation of clusters: Node 3 sends to its parent (Node 1) the message  $\{subtree-size, node-adjacency\}$  (i.e.,  $\{C+1, node-adjacency\}$ ). Node 1 notes that the number of nodes in its

*subtree-size equals  $C+1$ . So it creates the cluster  $C_1= \{<8>, <9>, 3, 1\}$ . The number of the remaining nodes (3 nodes) is less than  $C$ . Hence the cluster  $C_2= \{2, 4, 7\}$  is created.*

**3.4 Attacks and solution:**

**a. Attacks:** a malicious node can launch an attack with a lack of information sent to its neighbor in the step *tree-discovery*. The attack is in the construction of the tree. The malicious node which has one parent sends another beacon to other neighbor to take him as another parent. So the malicious node has two parents and it creates a cycle in the tree. The impact of this cycle is in the step *cluster-formation*: the malicious node sends to its both parents the message  $\{ subtree-size, node-adjacency\}$ , which initiate a cluster-formation at the same time, and the malicious node participates in the two initiations to provide inconsistency clusters. For example in figure 4 if we consider the cycle, created with the malicious node of identity 3, it sends the information  $(C+1=0,5C+0,5C+1)$  to 1 and 4 that initiate *Cluster-Formation*. The resulting clusters are  $C_1=\{1,3,<8>,<9>\}$  and  $C_2=\{4,3,<8>,<9>\}$  that are inconsistency clusters and the cluster agreement is broken. The malicious node may also modify this information  $(C+1)$  with an active attack in *Cluster-Formation* to eliminate the involvement of other nodes in *Cluster-Formation*.



**Figure 4. Tree-discovery and cluster formation**

**b. Solutions:**

**i. Secure Tree-discovery:** Our solution for the secure tree discovery is *controller based*. The root controls this step. Each node  $N$  that wants to take neighbor  $N_v$  as a parent calculates a secure hash value over the message  $M= \{src-id, N_v-id, parent-Id, root-Id, root-distance\}$ , and signs this hash value with the key shared with the root (controller). Next it sends it to the root  $N_r$ . The root checks if there is no cycle in the tree. To reach this goal it calculates two secures hash values over  $M_r=\{M, accord\}$  one with the key shared with  $N_v$  and the second key shared with the node  $N$ , and signs this hash value. It sends  $M_r$  as a response to  $N$  and  $N_v$ . However if there is a cycle or other problem the message of the root will be  $M_r=\{M, not agree\}$ .  $N_v$  verifies the information using the key shared with the root and takes it as a parent if there is an accord and if the root-distance is minimum.

**ii. Secure Cluster-Formation:** After the application of algorithm [22] in phase I each node knows the neighbors of its neighbors.

Starting from the leaves in the tree created in Secure Tree-discovery, the node  $N$  sends a secure hash value of the messages  $M=\{m=\{ subtree-size, node-adjacency\}, m_p\}$  to its parent  $p$ .  $M$  is signed and hashed with the key shared with  $p$ , and  $m_p$  is signed and hashed with the key shared with the parent of  $p$ . And  $N$  requests a secure feedback from the parent of  $p$  because  $p$  may do an active attack on the information.  $p$  checks if the size of the sub-tree is  $>C$  in order to initiate a cluster formation, but it cannot attack the information sent by its children, because its parent can detect this attack.

**3.5 Phase 4:**

**Remark:** Before we give the details of this phase, note that a node of figure 5 below is a clusterhead of a of a level in the hierarchical control clustering. Thus a node of figure 5 is in the cluster of at least  $C$  sensors and at most  $2C+1$ sensors.

In this phase each item  $a_1, a_2, \dots, a_n$  in  $WSN(n, p)$  is a secure hash value signed with the with key shared in  $WSN(n, p)$ . We use the tree created and used in phase 3. In the tree a node broadcasts only to its parent or its children. The goal of this phase is to route the outgoing items saved to their final destination. To route these elements without collision and without conflict we use the optimal coloring algorithm defined in [14]. Figure 5 show an example of this coloring procedure.

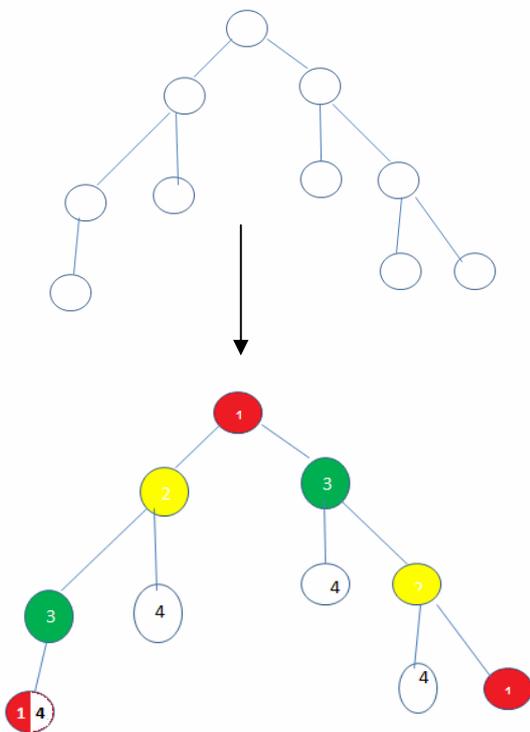


Figure 5. Optimal tree coloring

The goal of this sub-phase is to route the element to the clusterhead of its sensor destination. We use the following coloring algorithm.

**Algorithm routing\_of\_outgoing\_items**

```

COL is the number of colors, in our example
Begin
  If (( $\exists color\_j$  in set_color_i ) and (color_j mod COL =0))
  then
    Broadcast the item;
    for each color of j color_j= color_j+1;
  else
    for each color of j color_j= color_j+1;
  Other nodes r copies the item in their own local memory if
  the destination match with one of the identity in HUB(r);
end
    
```

*a. External broadcasts*

The algorithm works as follows: in each time slot the node (here clusterhead) checks if this slot is the appropriate time to broadcast. The appropriate time to broadcast is the time when the broadcast is taken without collision and without conflict, with instruction  $color\_i \bmod C = 0$ . The first group that has the color  $C$  broadcast their outgoing items in the first slot to their neighbors. But in the second slot, the round is for the group that has the color  $C-1$ , and so on. Clearly in a slot a clusterhead (node of figure 5) invites its resident sensors to broadcast one by one their outgoing data items to him. In the sub-slot that follows it broadcasts the received item to its neighbors. However this broadcasting process carried out by a clusterhead takes one slot time.

*b. Internal broadcasts*

On receiving an item whose destination matches with the ID of a sensor in *clique-i* the clusterhead of clique-i forwards it to its destination sensor in *clique\_i*. It selects a sensor that has to save this item otherwise (this is possible in virtue of the precedent remark)

**4. CONCLUSION**

One of the most challenges on permutation routing in wireless network multi-hop or single hop is the security. In this paper we proposed a secure permutation routing in multi-hop wireless sensors network which the objective is to secure the information exchanged by the sensors.

However some open problems remain. The derivation of a fault tolerant algorithm from the multi hop protocol of this paper which guarantees the delivery of data items to non faulty nodes is to be investigated. Also, the construction of an energy-efficient permutation routing protocol for multi-hop ad hoc network is a challenge.

## REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks Elsevier Journal*, Vol. 38, No. 4, pp. 393-422, March 2002.
- [2] S. Banerjee, S. Khuller, *et al.*, "A Clustering Scheme for Hierarchical Control in Multi-Hop Wireless Networks," *Proceedings of the 20th IEEE International Conference on Computer Communications*, Vol. 3, 2001, pp. 1028-1037.
- [3] A. Bomgni, J. F. Myoupo, "A Deterministic Protocol for Permutation Routing in Dense Multi-Hop Sensor Networks". *Wireless Sensor Network* vol.2 pp. 293-299, 2010.
- [4] A. Datta, "Fault-Tolerant and Energy-efficient Permutation Routing Protocol for Wireless Networks," *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, Nice, 2003, pp. 22-26.
- [5] A. Datta and A. Y. Zomaya, "An Energy-Efficient Permutation Routing Protocol for Single-Hop Radio Networks". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 4, 2004, pp. 331-338.
- [6] J. R. Douceur. The sybil attack. In First International Workshop on Peer-to-Peer Systems (IPTPS'02), Mar 2002.
- [7] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, Seattle, 1999, pp. 263-270.
- [8] N. Gura, A. Patel, and A. Wander. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *Proceedings of the 2004 Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2004.
- [9] T. Haenselmann (2006-04-05), *Sensor networks*, GFDL Wireless Sensor Network textbook, retrieved 2006-08-29.
- [10] Y. Hu, A. Perrig, and D. Johnson. Packet leases: A defense against wormhole attacks in wireless ad hoc networks. In *INFOCOM*, April 2003.
- [11] D. Karimou and J. F. Myoupo, "An Application of an Initialization Protocol to Permutation Routing in a Single-hop Mobile Ad-Hoc Networks," *Journal of Super-computing*, Vol. 31, No. 3, 2005, pp. 215-226.
- [12] D. Karimou and J. F. Myoupo, "A Fault Tolerant Permutation Routing Algorithm in Mobile Ad Hoc Networks," *International Conference on Networks-Part II*, 2005, pp. 107-115.
- [13] D. Karimou and J. F. Myoupo, "Randomized Permutation Routing in Multi-hop Ad Hoc Networks with Unknown destinations," *IFIP International Federation of Information Processing*, Vol. 212, 2006, pp. 47-59.
- [14] H. Lakhlef, J.F.Myoupo "An efficient permutation routing protocol in multi- hop wireless sensor network", manuscript, 2011.
- [15] D. J. Malan, M.Welsh, and M. D. Smith. A public-key infrastructure for key distribution in tinysec based on elliptic curve cryptography. In *SECON*, October 2004.
- [16] F. Myoupo, "Concurrent Broadcasts-Based Permutation Routing Algorithms in Radio Networks," *IEEE Symposium on Computers and Communications*, 2003, pp. 1272-1278.
- [17] K. Nakano, S. Olariu and J. L. Schwing, "Broadcast-Efficient Protocols for Mobile Radio Networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 10, No. 12, 1999, pp. 1276-1289.
- [18] K. Nakano, S. Olariu and A. Y. Zomaya, "Energy-Efficient Permutation Routing in Radio Networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 12, No. 6, 2001, pp. 544-557.
- [19] B. Parno, A. Perrig, and V. Gligor. Distributed detection of node replication attacks in sensor networks. In *IEEE Symposium on Security and Privacy*, May 2005.
- [20] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar. SPINS: Security protocols for sensor networks. In *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks*, July 2001.
- [21] K. Römer; Friedemann Mattern (December 2004), "The Design Space of Wireless Sensor Networks", *IEEE Wireless Communications* **11** (6): 54–61.
- [22] K. Sun, P. Peng and P. Ning, "Secure Distributed Cluster Formation in Wireless Sensor Networks," *22nd Annual Computer Security Applications Conference*, Las Vegas, 2006, pp. 131-140.
- [23] I. S. Walls and J. Žerovnik, "Optimal Permutation Routing on Mesh Networks," *International Network Optimization Conference*, Belgium, 22-25 April 2008.

# Software Security Engineering

## Monitoring and Control

Esmiralda Moradian, Anne Håkansson  
Department of Communication Systems, The Royal Institute of Technology, KTH,  
Forum 100, 164 40 Kista, Stockholm, Sweden  
[moradian@kth.se](mailto:moradian@kth.se), [annehak@kth.se](mailto:annehak@kth.se)

**Abstract** - *Poorly constructed software can induce security weaknesses and defects, which can be exploited by attackers. Despite many security standards and mechanisms, a vast amount of software systems have security vulnerabilities. The security problems induce the necessity of monitoring and controlling software development and maintenance. In this paper, we propose a multi-agent system that supports security in development of new systems and modification of existing systems. Thus, the multi-agent system verifies and validates the goals and requirements during different phases of development lifecycle. For the verification and validation, searching for information and mapping are needed. Searching for information about the project and security documents such as, risks, list of threats and vulnerabilities is performed by software agents. Comparisons and analyzes of requirements and use cases as well as mapping of those to attack patterns is performed by meta-agents. The proposed multi-agent system supports confidentiality, integrity, availability, accountability, and non-repudiation.*

**Keywords:** Software security engineering, monitoring and control, multi-agent system, verification and validation, security requirements, checklists.

## 1 Introduction

Software is the necessary component of the global infrastructure and a building block of every system. [9] For example, Service Oriented Architecture (SOA) implemented through web services, Software as a Service, and other internet-based systems are built of software. Software systems become interconnected and complex because of the growing interactivity between the organizations. However, the interconnected and distributed environments increase security problems [9]. The reason is that security, if considered, comes too late in the software engineering; often during design phase or implementation phase. In addition, software developers are not security experts, and most software requirements are incomplete. Moreover, software is often used in contexts for which it was not designed, and, most probably, the software is often changed during the development and its lifetime. Consequently, poorly constructed software induces security weaknesses and defects, which result in vulnerabilities. Malicious users exploit vulnerabilities in software in order to compromise the software's security properties or the

dependability of a component or between several interacting components [4]. Vulnerabilities, exploited by malicious users, can have serious consequences for the humanity, such as cause death. Furthermore, security breaches can be costly both in terms of efforts to fix problems and damages to organizations.

Therefore, no matter what architecture or model is used, software security is essential for the organizations and businesses success. Distributed software systems must satisfy security properties in order to avoid unwanted behavior. Software lifecycle should involve security from the very beginning, i. e., the requirements phase. Then the software needs to be controlled and security monitored at each stage of the development and the lifecycle process. Development progress should be logged. Verification of fulfillment of security goals and validation of security requirements, during different phases of development, is essential for stable and secure system state. Offered services are usually developed by different service providers that can increase the risks in the system.

There are many international standards that support development and management of software systems in a secure way, among others, Common Criteria (CC), ISO/IEC 27000 series, and ISO/IEC 12207. However, standards are seldom used during the development process due to evaluation of the software according to any of the standards is resource and time demanding [1]. Hence, many organizations do not work in line of supporting standards and, consequently, standards are often ignored by developers. Therefore, an automated process is required. In this paper, we present a multi-agent system that enhances secure development and management of software in an automated way and supports confidentiality, integrity and availability, as well as, accountability and non-repudiation. The proposed system authenticates human agents (HA) and logs actions of HA but also logs events in the system. Authentication and authorization of the users in development team can prevent unauthorized and malicious users (insiders) from accessing and modifying the software. Agents are able to perform, for example, searching, checking and matching tasks. Security requirements are verified and validated by meta-agents against goals, policies, risks and standards. Every step is documented by the meta-agents. The process generates recommendations in a form of checklists.

## 2 Related Work

This research extends the research in [10, 11] where Controlled Security Engineering Process was presented. Security requirements engineering was discussed. The importance of security policies and goals as well as risk oriented software engineering was pointed out. The architecture of the multi-agent system was presented.

Mouratidis et al. [13] propose Tropos, which is an agent oriented software methodology. Secure Tropos is a security oriented extension that defines security constraints and secure dependencies. Secure Tropos process allows model validation and design validation. Authors use principles of authorization, access control, and availability. In our research, we present a semi-autonomous method that controls development new systems or modifies existing systems against security needs. Hence, we provide a multi-agent system that verifies security goals satisfaction. The system enhances confidentiality, integrity, availability, accountability, and non-repudiation and also supports security in software lifecycle.

Bode et al. [3] propose a method that integrates software engineering and security engineering. Authors [3] describe how to get from security goals to the solutions in the software architecture. In our work, we use multi-agent system throughout engineering process in order to enhance security and support developers in every phase of software lifecycle, from requirements to maintenance. The proposed system provides authentication and authorization of the users (called human agents (HA) hereafter), as well as, monitors and records actions and changes for the purposes of integrity.

## 3 Software Security Lifecycle

Software engineering process lacks continuity and visibility, which makes it difficult to see progress in software construction [18]. Gathering and capturing requirements is the critical part of development that affects the development process during all other phases. Requirements express behavior of the system, the system states and object states and the transition from one state to another. However, it is common that security requirements of software system are not identified. [10] Requirements engineering often suffers from problems, such as, incomplete requirements specification or requirements that are specified without any analysis or analysis restricted to functional end-user requirements; and incapable of being validated. [2] Often, security is added after the system is developed. Security requirements are constraints on the functions of the system that operationalize one or more security goals [5].

Secure design involves logical, physical and component security architecture. [17] Logical security architecture concerns with specifying logical security services, such as, confidentiality and integrity protection; entities; domains; and security processing cycle, such as registration, login, and session management [17]. Physical security architecture concerns with specifying data model and structures (i.e.,

tables, messages, signatures), rules (i.e., conditions, procedures, actions), security mechanisms (i.e., access control, encryption, virus scanning), security technology infrastructure, and time dependency (i.e., events, time intervals) [17]. Integration of the component into the system can affect security of the overall system. Thus, security analyses of the components, as well as, security evaluation of each identified component and all components together are necessary.

Development involves writing code, i.e., converting design into executable programs. However, developers and programmers are usually not security experts. Implementation issues result from insecure coding. Howard and LeBlanc [7] point out that a set of coding guidelines should be defined for the development team. Developers should choose tools that enforce secure implementation. Programming languages that is difficult to use securely should be avoided. [4] Secure code is prerequisite for software security. Thus, the consistency, simplicity (make it easier to analyze and verify correctness and security), traceability, and minimal interdependency of components are important features to be considered. [ibid]

Code review is a way to detect security flaws and should be performed by security specialists in the area. Identified and managed security flaws, secure the code. All found flaws should be referred to the threats and logged in a database [7]. Fault handling should be implemented in order to prevent software from entering an insecure state. [4]

Testing is an important phase of software development process. Testing ensures that the implementation of each requirement is tested and that the system is ready for delivery. [11] However, missing security requirements, security flaws in architecture and design will produce unreliable test results. [10] Tests should be created so they correspond to the earlier identified security risks. [9] Thus, security testing should be a part of development life cycle.

Security testing is performed to verify that the design and code can withstand attack. [7] Security testing starts with component testing (also known as module or unit testing). Component tests focus on one component and verify that the component functions as expected. Component testing is performed in a controlled testing environment. [14] Earlier performed risk analysis should be taken into account and risks mitigated. [9] Integration testing verifies that collection of components work together as described in the specifications. [14]

Penetration testing is a simulation of the actions of a malicious user, i.e., attacking the system. There are different automatic tools that testers usually use. Penetration tests are performed in order to identify "intracomponent failures and assess security risks inherent at the design level". [9] Tests need to be documented for further analysis.

Analysis on how changes can affect the overall system should be made. However, authors in [4] point out that most people involved in software development lack knowledge

when it comes to software security engineering. Thus, they do not recognize how the mistakes, made during development, can result in weaknesses and vulnerabilities when software becomes operational. [4] Therefore, a multi-agent system that supports developers by providing verification and validation of security requirements and enforcing risk-driven checks and controls during all phases of lifecycle process is highly needed.

Software operates in dynamic environment. Software anomalies that affect security, often, are results from uncertain environment (where software operates) [4]. Despite of uncertainties, software must be able to respond to changes in its execution environment to remain in a secure state. Intelligent agents can be used in order to identify deviations in the environment and react to changes.

## 4 Multi-Agent Systems

The more complex a system is the more difficult is to control it and consequently predict its behavior. The use of agent systems for security monitoring, control and management is increasing due to growing complexity of distributed systems. For example, a multi-agent system can consist of a network of agents that interact with each other in order to solve security problems. [12]

We use a multi-agent system that supports security throughout software engineering process. The proposed multi-agent system operates both internally (within organization) and externally in distributed environment. Agents can work on different layers, for example, Identity layer, Web layer, and Web Services layer.

A single agent does not have complete information and is not capable to solve the entire problem on its own. Therefore, we use a team of agents. Our system consists of two levels of agents, i.e. ground-level software agents and meta-level agents. The agents are communicative, mobile, cooperative, goal-oriented, autonomous, adaptive, and reactive [15]. The agents are communicative in the sense that they are able to communicate with users (HA), other agents in the system, and other systems. [11] The agents are mobile, and can move between different locations over the networks while searching for components and services. Agents, in our system, cooperate with each other by exchanging messages. The tasks have to be performed by the agents without any external guidance, and, thus, must be autonomous. Autonomous agents can operate without human intervention. The meta-agents, in our system, can for example, compare, analyze and combine information received from software agents. The meta-agents are also autonomous, i.e., meta-agents are able to act autonomously, reason and take decision in order to satisfy their objectives. Thus, the meta-agents are intelligent agents.

The agents, in our multi-agent system, are goal-oriented information agents, since they perform search for information according to specified goals. Moreover, agents are also able to find, filter and classify information. Search can be performed

within organization network and on the Internet. Meta-agents, on the other hand, are reactive and adaptive, since, they can react to events in the environment according to predetermined rules, as well as, learn, respond and adapt to changed environments.

The information agents work in a deterministic environment, which means that next state of the environment is determined by the current state and the action that is being executed by an agent. [16] The environment is dynamic, since it can change during execution. Therefore, meta-agents are used to keep track on changes in the environment while it is deciding on an action [8].

The information agents perform one task at the time in an episodic environment. The task is divided into many simple tasks due to task complexity. The proposed system is able to deal with tasks such as: authentication, authorization, access control, searching, security monitoring and tracking actions and changes for the purposes of evidence. Moreover, the system is able to analyze and compare information that is necessary to achieve the goal, verify requirements, make decisions, identify security properties of the components and services, propose recommendations and create and update checklists.

## 5 Security Monitoring and Control of Software Engineering with Multi-Agent System

Our multi-agent system supports security in software engineering process. Starting from the requirement phase, the development process is security oriented and controlled by the agents that verify and validate security requirements throughout lifecycle. Developers can intentionally or unintentionally, induce weaknesses in software. Moreover, many changes occur during development of software. Misunderstanding of changes and lack of communication among developers can result in software insecurity. Most weaknesses can be prevented by applying necessary security activities. [4] Though, the more software grows in size and complexity, it becomes more difficult to control. Thus, security monitoring and control, as well as, security checkpoints are needed throughout the lifecycle.

The proposed multi-agent system provides monitoring and controls the software lifecycle. Moreover, to minimize malicious actions of "insiders" and "outsiders", our system supports following security properties: confidentiality, integrity, availability, accountability, and non-repudiation. To provide confidentiality, only securely identified HA are allowed to login in the system. Only authorized human agents are permitted to access information. Access rights can be changed only by the security manager. To prevent improper or unauthorized change of information (to provide integrity), in our system, only legitimate users are allowed to make changes in accordance to the permissions they have. Principle of least privilege is applied here. Human agents, that have been

securely authenticated, must be able to access information or resource they have authorization for in order to be able to perform the task(s). By tracking actions of HA, the proposed multi-agent system is able to provide accountability. All activities are logged. Every single action can be tracked to the HA that made the change. Thus, if HA made, for example, modification to design specification (or any other action) it cannot deny the action later, which is non-repudiation.

In our system, we define two categories of the agents: software agents and meta-agents. The ground level software agents and meta-level agents have different tasks. The ground-level software agents search for information and deliver results and meta-agents have a supervisory role of these software agents. The meta-agents are Authenticator/Interface agent(s), Coordination/Management agent, Control agent and Match agent while searching agents are ground level software agents. Meta-agent controls software agents by keeping track on them. For the amount of databases to be searched the number of meta-agents and software agents are expanded. [11] Some tasks of the agents are depicted in Figure 1.

The human agent (HA) communicates with the system via AuthenticatorAgent. The AuthenticatorAgent authenticates HA and checks access rights of HA. If identification of HA fails then access to the system is denied. An AuthenticatorAgent also checks if human agent has access rights to work with the specific project. Furthermore, access control list is checked in order to prevent unauthorized users from accessing the information they have no permissions to access. Moreover, AuthenticatorAgent logs activities of HA. Additionally, AuthenticatorAgent monitors events, records the results, and send message to the Coordination/Management agent. The sequence is as follows:

1. User (HA) submit user credentials to multi-agent system (MAS)
2. AuthenticatorAgent verifies LoginContext
3. AuthenticatorAgent returns success/failure
4. AuthenticatorAgent authorizes the request (in case of success)
5. AuthenticatorAgent validates the request
6. AuthenticatorAgent logs the request

AuthenticatorAgent logs events.

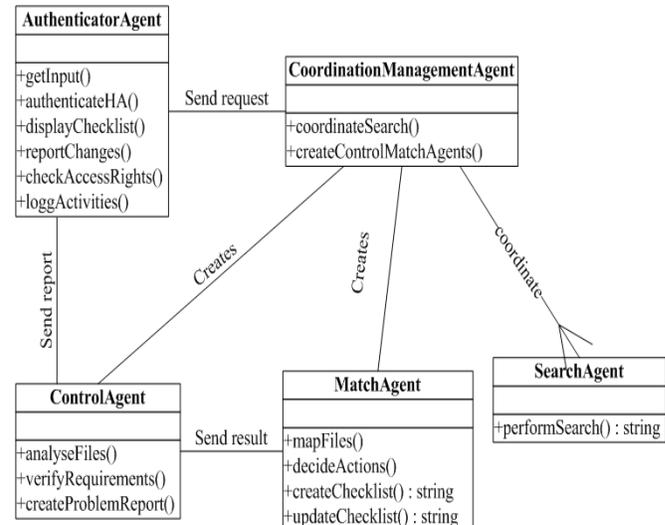


Fig. 1 Tasks managed by agents

When HA is authenticated successfully and authorized by the system, HA makes a request. Request consists of ProjectID, protection profile (PP) and other key parameters that are important for fulfilling the request. The software agents execute request. Software agents are the search agents that search for information, such as project documents (such as requirements, risk analyses, specifications etc.), policies, standards, documented knowledge from earlier projects, components and services according to received commands from the meta-agent. All documents should be tagged. The software agents start to execute and a meta-agent is following along to the next node. Search is executed in parallel in multiple databases, knowledge bases and registries to minimize search time.

Meta-agents are the management agents, coordination agents, matching agents and checking agents. The meta-agents, in our system, can compare, analyze and combine information received from software agents. The meta-agents are autonomous, i.e., meta-agents are able to act autonomously, reason and take decision in order to satisfy goals.

Coordination/Management agent is an intelligent meta-agent responsible for coordinating search and assigning tasks to search agents. This agent makes decisions based on input facts and observations. Depending on task complexity, agents can create new agents by cloning themselves. The replication allows performing tasks in parallel manner. [11] Coordination/Management agent creates control and match agents by cloning itself. Software agents verify links between documents and retrieve documents to map.

Control agent is responsible for analyzing goals and verification of requirements. Meta agents analyze requirements to identify errors: conflicting, missing and/or incomplete business and security requirements. In response to errors a problem report (faults) is generated and the process is halted until new request or task is received. If no errors are found the analyses result is send to match agent that provides

mapping. These meta-agents maps documents to the requirements, find where requirements fits in policies (security and organizational), goals (security and business) and security standards. Mapping to relevant expert knowledge stored in knowledge base is performed in order to retrieve success/failure scenarios. Requirements are also mapped to the identified risks, design specifications, threats and attack patterns to determine which attacks can target the software. The simplified sequence diagram shows in Figure 2 partial messages exchange between the agents.

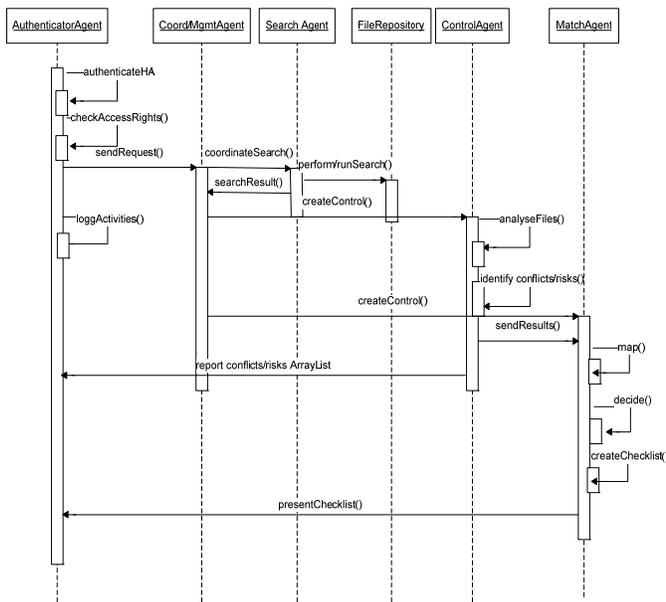


Fig.2 Message exchange between agents

Software agents also perform external search in component pools and service registries to find available requested components and web services. Since neither firewalls nor security technologies based on PKI are useful for Web Services security [6] it is necessary to analyze component(s) security properties in order to identify security risks. This task is performed by the meta-agent(s). Web Services allows corporate resources to be accessible to outsiders that can result catastrophic system failures and data loss. [6] Thus, security properties of components or services should be identified. Security testing as well as penetration testing should be performed. Risks must be identified, categorize, prioritize and managed. Vulnerability reports are generated by meta-agent. Reports are sent to a DB and presented for HA.

During operation phase, software must be able to respond to changes in environment to remain in a secure state. However, environment uncertainties can complicate the identification of the point at which the software entered an insecure state and can obstruct the determination of actions that returns software in a secure state [4]. The proposed multi-agent system can be used to facilitate identification of

insecure state point (apply for development and operational phases). Agents in our system can react to and record changes in environment as well as generate report with necessary information. This task is performed by meta-agents due to their ability to learn, respond and adapt to changed environments. The meta-agents act as follows:

1. Observe operational environment
2. React and capture events
3. Identify insecure state point
4. Monitor software behavior
5. Identify success of potential attacks
6. Identify and prioritize attack patterns
7. Identify and prioritize vulnerabilities
8. Create report and recommendations (checklists)
9. Present reports and checklists for HA

Meta-agents generate recommendations in a form of checklists. Reports are generated as PDF-files and consist of texts, tables and relations visualized with graphics. The multi-agent system provides the holistic view of software development and operation, which makes it easier to see progress in different phases of lifecycle. The proposed system handles security monitoring as well as input and change control which minimize security issues and improve software security. Logged information (documents, reports, analyses, etc) facilitate maintenance of the software.

## 6 Conclusion and Further Work

In this paper, we presented a multi-agent system that supports security in software engineering process. The proposed system monitors and controls development and operation of new systems and modification of existing systems. The multi-agent system supports verification and validation of security goals and requirements during different phases of software lifecycle. Moreover, the system provides recommendations to the developers as a result of mapping activities and performed analyses. Furthermore, due to ability of the agents to operate in dynamic environment and react to changes in the environment our multi-agent system can make it easier for developers to identify the point when software enters an insecure state, and thus, enhance security. The proposed multi-agent system supports security properties such as confidentiality, integrity, availability, accountability, and non-repudiation.

## 7 References

- [1] Abbas, H. Yngström, L., Hemani, A. Option Based Evaluation: Security Evaluation of IT Products Based on Options Theory. First IEEE Eastern European Conference on the Engineering of Computer Based Systems, pp.134-141, October 2009.

- [2] Allen, H.J., Barnum, S., Ellison, R.J., McGraw, G., Mead, N.R. *Software Security Engineering A Guide for Project Managers*. Pearson Ed., ISBN: 0-321-50917-X, 2008.
- [3] Bode, S., Fischer, A., Kühnhauser, W., Riebisch, M. *Software Architectural Design meets Security Engineering*. 16<sup>th</sup> Annual IEEE International Conference and Workshop on the Engineering of Computer Based System (ECBS 2009), San Francisco, CA, USA, April 13-16, pp. 109-118, IEEE Computer Society, 2009
- [4] Goertzel, K.M., Winograd, T., McKinley, H. L., Holley, P., Hamilton, B. A. *SECURITY IN THE SOFTWARE LIFECYCLE Making Software Development Processes—and Software Produced by Them—More Secure* DRAFT Version 1.2 (August 2006)
- [5] Haley, C.B., Moffett, J.D., Laney, R., Nuseibeh, B. A Framework to security requirements engineering. *SESS'06*, May 20–21, 2006, Copyright 2006 ACM 1-59593- 085-X/06/0005, Shanghai, China.
- [6] Hartman, B. Flinn, Donald J. Beznosov, K. Kawamoto, S. *Mastering Web Services Security*. Wiley (2003)
- [7] Howard, M., LeBlanc, D. *Writing Secure Code*. 2nd Ed. ISBN 0-7356-1722-8, Microsoft Press (2003)
- [8] Håkansson, A., and Hartung, R.: Calculating optimal decision using Meta-level agents for Multi-Agents in Networks. 11th International Conference, KES2007, LNCS/LNAI, Springer-Verlag, Heidelberg New York, ISBN:3-540-74817-2, LNAI4692-I, pp. 180-188 (2007:b)
- [9] McGraw, G. *Software Security Building Security In*. ISBN 0-321-35670-5, Pearson Ed. Printed in US in Crawfordsville, Indiana, 2006.
- [10] Moradian, E. *System Engineering Security*. Springer Berlin/Heidelberg. Volume 5712/2009, pp.821-828. ISBN: 978-3-642-04591-2, (September 2009)
- [11] Moradian, E., Håkansson, A. Controlling Security of Software Development with Multi-Agent System. In Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems: Part IV Springer Berlin/Heidelberg. Volume 6279, pp. 98-107, (September 2010)
- [12] Moradian, E. 'Secure transmission and processing of information in organisations systems', *Int. J. Intelligent Defence Support Systems*, Vol. 2, No. 1, pp.58–71, (2009)
- [13] Mouratidis, H., Giorgini, P., Manson, G. *Integrating Security and System Engineering: Towards the Modelling of Secure Information Systems*. Volume 2681/2003 pp. 63-78. Springer Berlin/Heidelberg (2003)
- [14] Pfleeger, S. L., *Software Engineering Theory and Practice*. Prentice-Hall, Inc. 2 Ed. 2001 ISBN 0-13-029049-1
- [15] Phillips-Wren, G. *Assisting Human Decision Making with Intelligent Technologies*. Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part 1, Zagreb, Croatia, Vol. 5177, pp. 1-10, (2008)
- [16] Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*. Prentice-Hall. 1995. ISBN: 0-13-103805-2
- [17] Sherwood, J., Clark, A., Lynas, D.: *Enterprise Security Architecture A Business-Driven Approach*. CMP Books, 2005, ISBN 1-57820318-X
- [18] Van Vliet, H.: *Software Engineering Principles and Practice*. 2nd Ed. JohnWiley and sons, 2004. ISBN 0-471-97508-7

# A Novel approach as Multi-place Watermarking for Security in Database

**Brijesh B. Mehta, Udai Pratap Rao**

Dept. of Computer Engineering, S. V. National Institute of Technology, Surat, Gujarat, INDIA-395007

**Abstract-** *Digital multimedia watermarking technology had suggested in the last decade to embed copyright information in digital objects such as images, audio and video. However, the increasing use of relational database systems in many real-life applications created an ever-increasing need for watermarking database systems. As a result, watermarking relational database systems is now merging as a research area that deals with the legal issue of copyright protection of database systems. The main goal of database watermarking is to generate robust and impersistant watermark for database. In this paper we propose a method, based on image as watermark and this watermark is embedded over the database at two different attribute of tuple, one in the numeric attribute of tuple and another in the date attribute's time (seconds) field. Our approach can be applied for numerical and categorical database.*

**Keywords:** Database Security, Database Watermarking, Multi-place Watermarking.

## 1 Introduction

Watermarking is firstly, introduced for image processing and then it extended to security of text and multimedia data. Now days, it also used for database and software. There is so much work done so far by many researches in the watermarking multimedia data [1] [2] [3]. Most of this method were initially developed for images [4] and later extended to video [5] and audio data [6][7]. Software watermarking techniques [8][9][10][11], also been introduced but it did not get much success because they are easily detectable in code. Due to differences between multimedia and database we cannot directly use any of the technique as it is for database, which developed for multimedia data. These differences include [12][13]:

- A multimedia object consists of a large number of bits, with considerable redundancy. Therefore, the watermark has more space to hide where as a database relation consists of tuples, each of which represents a separate object. So the watermark needs to be spread over these separate objects.
- The relative spatial/temporal positioning of various pieces of a multimedia object typically does not

change. Whereas, tuples may changes with updates in database.

- Portions of a multimedia object cannot be dropped or replaced arbitrarily without causing perceptual changes in the object. Whereas, tuples may simply be dropped by delete operation in database.

In this paper, we have proposed a new approach for robust database watermarking.[15] When we are talking about watermarking as copyright protection robustness is a very important issue as to prove ownership of data, user have to detect their watermark in data without any damage or say defect this defines robustness of watermark. Means the rate of correctly detection of watermark is also called robustness of the watermark. There are two main stages when we apply watermarking to data, (1) Watermark Embedding, and (2) Watermark Detection. In Watermark Embedding or say watermark insertion, we are applying or inserting a watermark in to the object or data, which, we want to protect. In watermark detection or watermark extraction, we try to extract the watermark from data or object or just check for the presence of watermark in data in some cases.

Rest of the paper is organized as follows: in Sec. 2, we discuss related work in this area. In Sec. 3, we discuss overview of our novel approach. In Sec. 4, we discuss algorithm for our approach. In Sec. 5, we evaluated the performance of our algorithm with reference to different attacks. Then conclusion and future work in Sec. 6 and references in Sec. 7.

## 2 Related Work

Watermarking relational databases is a relatively new research area that deals with the legal issue of copyright protection of relational databases. Therefore, literature in this area has been very limited, and focused on embedding short strings of binary bits in numerical databases [18]. In year 2000, S. Khanna et al. proposed the novel idea of controlling the security of database with digital watermark [14], which arouses the researchers' interest in watermarking database. So mainly there are two paths come out for database watermarking.

Firstly Agrawal et al.[12] presented a scheme and implemented it. This algorithm assumes that numeric attribute

can tolerate modifications of some least significant bits. So, Tuples selected first for watermark embedding. Then certain bits of some attributes of the selected tuples modified to embed watermark bits.

Second scheme of Sion et al.[15] in which, all tuples securely divided into non-intersecting subsets. A single watermark bit embedded into tuples of a subset by modifying the distribution of tuples values. The same watermark bit embedded repeatedly across several subsets and the majority voting technique employed to detect the watermark.

Some other authors had also tried to improve above two approaches with their own ideas to make them more secure and robust. From these we have studied some of the papers which are:

- Watermark based copyright protection of outsourced database by ZHU Qin et al.[16] Which, explains a database watermarking method based on first approach we discussed above with chaotic random number generator.
- A Speech based algorithm for watermarking relational databases by Haiqing Wang et al. [17] which, explains a database watermarking methods same as above but here they have used voice as a watermark.
- Watermarking Relational Database Systems by Ashraf Odeh et al.[18] which, explains a watermarking method based on first approach but here image is used as watermark and it is embedded in Date attribute's time field. We have also used this algorithm in our approach as one of the algorithm to embed watermark.
- Robust and Blind Watermarking of Relational Database Systems by Ali Al-Haj et al.[19] which, explains image based watermark embedding method in the non-numeric field of database as spaces and double space for encoding the watermark bit.
- One of the latest paper we go through is about a new relational watermarking scheme resilient to additive attacks by Nagarjuna Settipalli & R. Manjula.[20] As title suggest they just taken care of additive attacks applied on database watermarking.

### 3 Proposed Approach

Our proposed approach is based on the modification of two algorithms given by Agrawal et al. [13] And Ashraf Odeh et al. [18]. In earlier approach[13][18] author used a single attribute of a tuple to embed a watermark but we are embedding the same watermark in two attributes using our proposed algorithms. Therefore, it will be difficult for attacker to remove both watermarks from the database, based on extracted bits from both algorithms we can generate original watermark very easily, and we can prove ownership of database. In our method binary image is used as watermark. The whole procedure of embedding and extraction of watermark is performing in two phases, in first phase we insert

watermark in the numeric field of the database and in second phase we insert a watermark in the seconds field of database same way at the time of extraction we follow the reverse order of above phases.

## 4 Algorithms

Main purpose of writing this algorithm is to give more robust database watermarking technique. In earlier techniques they insert watermark at one place only so we found that, it can be removed by some of the database update operation where is in our approach its little more difficult.

Notations used in this algorithm are:

- $n$  - Number of tuples in the relation
- $v$  - Number of attributes in the relation available for marking
- $k_1(5\text{-bits})$ - Key used to determine the place where watermark can be inserted
- $k_2(4\text{-bits})$ - Key used to generate watermark for second phase

### 4.1 Watermark insertion / embedding

Suppose that the scheme of a database relation is  $R(P, A_0, \dots, A_{v-1})$ , where  $P$  is primary key attribute, If there is no primary key, auto-increasing attribute will be added to act as primary key. Assume that all  $v$  attribute are numeric and are candidates for marking.

A very important assumption regarding database watermarking is that small changes in LSB of a numeric attribute are tolerable within certain precision range. To get a copyright protection data owner should pay this price. In fact, it is noteworthy that the publisher of books of mathematical tables has been introducing small errors in their tables for centuries to identify pirated copies [3].

Here we are going to use two keys for the watermark insertion  $k_1$  and  $k_2$ . Both  $k_1$  and  $k_2$  are known by database owner only.  $k_1$  is used to select the tuple in which we need to insert a watermark where as  $k_2$  is a 4 bit key which is used to convert single bit watermark into 5 bit watermark to easily embedding it into time field of date attribute. The parameters  $v, n, k_1, k_2$  are known to the owner only. We are using binary image as watermark.

In first phase we embed the watermark using the modified algorithm of Agrawal et al.[13] we have removed a keyed hash function from the algorithm provided by Agrawal et al. and made it simple to insert and detect though we compromise with the security but as we are inserting watermark at two places, we can take this chances.

1. For each tuple  $r \in R$  do
2. If  $(F(r.P) \bmod k_1 \text{ equals } 0)$  then

3. For each attribute of tuple
4. If(attribute  $\in v$ ) then
5. Find LSB of that attribute and replace it with watermark bit
6. End if
7. End loop
8. End if
9. End loop

Therefore, by this way we have inserted watermark at one place now it's time to insert the same watermark at other place and that other place is the time field of date attribute. Hiding the binary information in the seconds field (SS) should have the least effect on the usability of the database. A major advantage of using the time attribute is the large bit-capacity available for hiding the watermark, and thus large watermarks can be easily hide, if required. Therefore, in second phase we use algorithm proposed by Ashraf et al. [18] with slight modification as we are taking MM (Minuit) field to decide that is it possible to insert a watermark or not. Whereas in algorithm given by Ashraf et al., they are using SS field and directly inserting 5 bits of binary image into the SS field whereas we are inserting only 1 bit of image which is concatenated with 4 bit of key. As we are embedding single bit in attribute but this algorithm uses 5-bit watermark so we first do concatenation operation between key  $k_2$  and watermark bit to get 5 bit new watermark and then uses the algorithm to insert a watermark.

1. Concatenation of the value of watermark bit and  $k_2$
2. Find the decimal equivalent of the string
3. Embed the decimal number in tuples selected by the pre-defined key  $k_1$  as follows:
  - 3.1. For each selected tuple do
  - 3.2. For each selected Time attribute do
  - 3.3. If the 'MM' field of the 'Time' mode  $k_1 = 0$
  - 3.4. Embed the decimal number in SS field
  - 3.5. Else Next attribute
  - 3.6. End if
  - 3.7. End loop
  - 3.8. End loop

Therefore, by this way we have completed multi-place watermark insertion in the database but now the difficult part is to extract the watermark from two places and then comparing them to check for the original watermark

## 4.2 Watermark extraction

Watermark Detection is the procedure to check whether watermark is present in data where as in watermark extraction watermark is been carried out and regenerated.—Algorithm designed by Agrawal et al.[13] was for detection of the watermark not to extract it because it calculates total count and match count, so we need to slightly modify it to extract the watermark. So, as we mention earlier in insertion that we have removed the keyed hash function from the algorithm and we

are actually taking values from the data not only checking that whether watermark is there or not.

1. For each tuple  $s \in S$  do
2. If((s.P) mod  $k_1$  equals 0) then
3. For each attribute of tuple
4. If(attribute  $\in v$ ) then
5. Find LSB of that attribute and extract it as our watermark
6. End if
7. End loop
8. End if
9. End loop

Now, in above algorithm if we find any bit that has changed after insertion, then we are putting it as zero and all detected bits as it is. So, if we find 1 in the watermark then we can say that they are the correct values but for zero we can't say anything yet but after second phase we have more clear idea about the original watermark.

In second phase, we are using Ashraf Odeh et al.'s algorithm with slight modification that instead of generating binary image from that binary equivalent of the extracted watermark. We are taking only LSB of the binary data.

1. Extract the decimal number in tuple selected by the pre-defined key  $k_1$  as follows:
  - 1.1. For each selected tuple do
  - 1.2. For each selected 'Time' attribute do
  - 1.3. If the 'MM' field of the 'time' mode  $k_1 = 0$
  - 1.4. Extract the decimal number from SS field
  - 1.5. Else Next attribute
  - 1.6. End if
  - 1.7. End loop
  - 1.8. End loop
2. Find the binary equivalent of the extracted decimal number
3. Extract last bit (LSB) from it which indicate our watermark.

Now, we have two watermarks, which may be changed or not changed. Now, we will compare these two watermarks with original watermark. Therefore, if any one of them will match we will consider it true and by this way we can get the correctly extracted watermark in percentage by the following algorithm.

```

Matchcount=0, totalcount=0
For each bit of watermark
  If( WM = WM1 or WM=WM2) then
    matchcount=matchcount+1
  End if
  Totalcount=totalcount+1
End for

```

## 5 Performance Evaluation

Proposed algorithm has been tested and evaluated on an experimental data set consists of approximately 5000 tuples. We are taking care of robustness in our performance evaluation. We have completed all our programming in C language. We have considered few of the major attacks applied on Database Watermarking like:

### 5.1 Subset addition attack

In this type of attack, the attacker adds a set of tuples to the original database.[18] This kind of attack does not have any effect on our algorithm coz new tuples added have no watermark embedded in them so at the time of extraction they are simply ignored so we get 100% extraction after adding 100% new tuples.

### 5.2 Subset deletion attack

In this type of attack, the attacker may delete a subset of the tuples of the watermarked database hoping that the watermark will be removed.[18] Graph in Figure 1 shows that watermark can completely only be removed by all the tuples because if 5% tuples are there then also we can extract from the database this shows that our algorithm is very robust against this kind of attack.

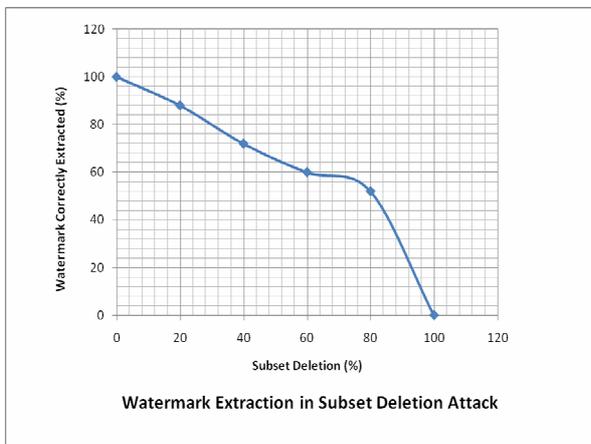


Fig. 1. Robustness Results due to the Subset Deletion Attack

### 5.3 Subset alteration attack

In this type of attack, the attacker alters the tuples of the database through operations such as linear transformation. [18] By doing this attacker thinks that he/she will be get success in removing the watermark from database but graph in figure 2 shows that even altering all the tuples watermark can still be extracted.

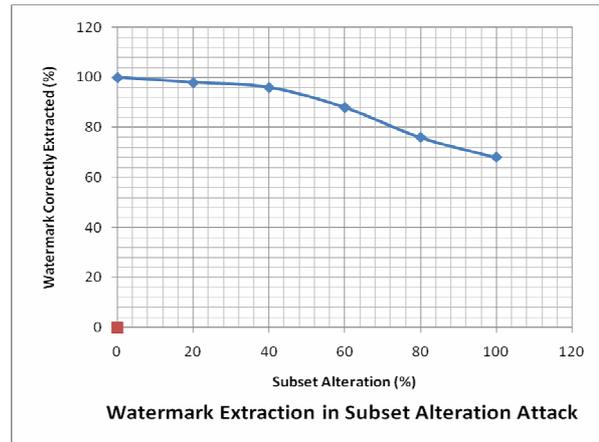


Fig. 2. Robustness Results due to the Subset Alteration Attack

### 5.4 Subset selection attack

In this type of attack, the attacker randomly selects a subset of the original database that might still provide value for its intended purpose. [18] By doing this attacker thinks that the small he selects has no watermark embedded in it but as our algorithm embeds watermark at two places even he selects 10% part of dataset it contains a watermark in it and it can be proved by the graph in the figure 3.

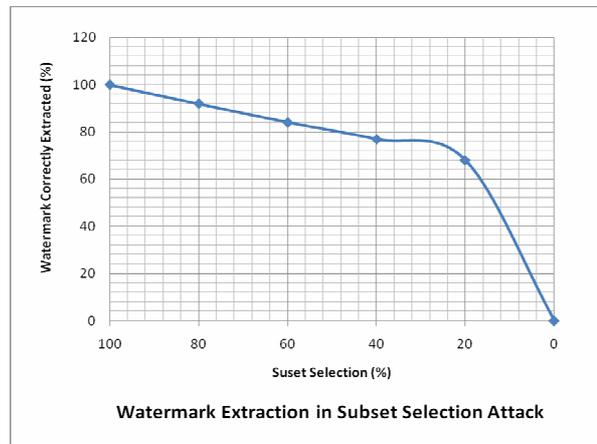


Fig. 3. Robustness Results due to the Subset selection Attack

## 6 Conclusions and Future Work

We have proposed a novel approach for robust database watermarking which is very useful in copyright protection of database. As we are inserting same watermark at different places so, there is a less chances of it to get attacked and if so, it is comparatively easy to extract the original watermark because of the watermark is embedded at two places. We can have one correctly extracted watermark from that two places. Though it may be more costlier in a context of data correctness as it changes many attributes of a tuple in database

but if we are thinking about robust copyright protection then we need to pay this price.

As in our proposed approach we are not only detecting the watermark but also extracting it. Therefore, instead of binary image we can insert some kind of biometrics data like, speech. For eg. Speech has a unique characteristic means almost everyone's voice is different than the other person.

## 7 References

- [1] J. Cox, M. L. Miller, "A review of watermarking and the importance of perceptual modelling," In proc. of electronic imaging, 1997.
- [2] N. F. Johnson, Z. Duric, S. Jajodia, "Information Hiding: Steganography and watermarking-Attacks and Countermeasures," Kluwer Academic Publishers, 2000.
- [3] S. Katzenbeisser, F. A. Petitcolas, editors, "Information Hiding Techniques for Steganography and Digital Watermarking," Artech House, 2000.
- [4] Joseph J. K. O Ruanaidh, W. J. Dowling, F. M. Bolend. "Watermarking digital images for copyright protection," IEEE proceedings on vision, Signal and Image Processing, 143(4), pp. 250-256, 1996.
- [5] F. Hartung B. Girod, "Watermarking of uncompressed and compressed video," Signal Processing, 66(3), pp. 283-301, 1998.
- [6] L. Boney, A. H. Tewfik, K. N. Hamdy, "Digital watermarks for audio signals," In International Conference on Multimedia Computing and Systems, Hiroshima, Japan, 1996.
- [7] S. Czerwinski, "Digital music distribution and audio watermarking," Available from <http://citeseer.nj.nec.com>
- [8] C. S. Collberg, C. Thomborson, "Watermarking, Tamper-Proofing, and Obfuscation-Tools for Software Protection," Technical Report 2000-03, University of Arizona, 2000.
- [9] C. Collberg, E. Carter S. Debray, A. Huntwork, C. Linn M. Stepp, "Dynamic Path Based Software Watermarking," University of Arizona, 2003.
- [10] Patrick Cousot, Radhia Cousot, "An Abstract Interpretation based framework for Software Watermarking," POPL' 04, Venice, Italy. ACM 2004.
- [11] Gaurav Gupta, Josef Pieprzyk, "Software watermarking Resilient to Debugging Attacks," Journal of multimedia, vol. 2, no. 2. Academy publisher, 2007
- [12] R. Agrawal, J. Kiernan, "Watermarking relational databases," Proceedings of the 28<sup>th</sup> International Conferences on VLDB, pp. 155-166, 2002.
- [13] R. Agrawal, J. Kiernan, "Watermarking relational data: Framework, Algorithms and Analysis," VLDB Journal, pp. 155-166, 2003.
- [14] Sanjeev Khanna, Francis Zane, "Watermarking maps: hiding information in structured data," Int'l Conf. SODA 2000, San Francisco, California, USA, pp. 596-605. 2000.
- [15] Radu Sion, Mikhail Atallah, Sunil Prabhakar, "Rights protection for relational data," Int'l Conf. 2003 ACM SIGMOD International Conference, Madison, Wisconsin, USA 2003.
- [16] ZHU Qin, YANG Ying, LE Jia-jin, LUO Yishu "Watermark based Copyright Protection of Outsourced Database," IEEE, IDEAS, pp. 1-5, 2006.
- [17] Haiquig Wang, Xinachin Cui, Zaihi Cao, "A speech based Algorithm for Watermarking Relational Database," IEEE, ISIP, pp. 603-606, 2008.
- [18] Ashraf Odeh, Ali Al-Haj, "Watermarking Relational Database Systems," IEEE, pp. 270-274, 2008.
- [19] Ashraf Odeh, Ali Al-Haj, "Robust and blind watermarking of Relational Database System," Journal of Computer Science 4 (12), pp. 1024-1029, 2008.
- [20] Nagarjuna Settipalli, Prof. R. Manjula, "A new Relational Watermarking Scheme Resilient to Additive Attacks," International Journal of Computer Application (0975-8887), Volume 10-No. 5, 2010.

# Quantifying the Role of Access Control in End-to-End Network Security

A. Usama Ahmed, B. Ammar Masood, C. Liaquat Ali Khan  
Security and Cryptology Cell, Air University, Islamabad, Pakistan

**Abstract** - Modern day networks consist of a mix and match of technologies with varying capabilities. Securing such networks is a tedious task and demands a lot of contribution from security professionals. However, failing to conduct an in depth and standardized analysis may result in an imperfect network security design. ITU-T provides recommendation for end-to-end network security in its standard X.805 which precisely lays down the foundation for the assessment of network security. Keeping X.805 requirements and the current network in view, the next thing that security professionals face is the correlation of both and its practical implementation. This paper contributes to answer the same and gives a comprehensive methodology for practical implementation of X.805 architecture. Also this paper takes into account the current trend of Network Access Control solution and its quantitative contribution towards achieving the desired goals set by X.805 standard.

**Keywords:** X.805; Network Access Control; end-to-end security; Compliance; ISO-18028;

## 1 Introduction

Entities participating in the information flow from one end to the other are distinct in their characteristics and so are vulnerable in their own perspective. Strengthening the security of individual entities often leave gaps that are overlooked and that becomes the cause of denial, destruction and even loss of information vital to the communicating parties. Planning, implementing and maintaining the security of a network need precise efforts. Security professionals had always faced a usual challenge of enhancing the security capabilities of the network to meet the need of the day.

Conventional methods of network security had been simple and undemanding. There had been simple networks with unsophisticated applications and so had their vulnerabilities. The best possible way had been to have a perimeter defense with a border firewall and a user provisioning system for access control. As the networks grew, and enhanced capabilities were introduced in the equipments as well as the applications, the job of a security professional meant more than planning a perimeter defense.

The fast paced growth in the capabilities of the network and the variance in technologies let the professionals deal with

security in a more elaborate way than ever. For the same the security professionals has followed best practices for the information security as per their requirements and needs of network. These requirements revolve around the overall objective of organization's information security policy. Adhering to the best practices may suffice but still leave the security professionals with a question in mind. What were the objectives set at the start of the day? What have been achieved and what has been left out?

### 1.1 ITU-T X.805

A standard and definite approach is, therefore, required to be used so that a quantifiable effect of security endeavors can be realized. The well known and distinguished set of recommendations is given by ITU-T as X.805 standard [1] for end-to-end security of a network infrastructure. It aims to achieve an overall safe and secure network by answering the most common security concerns that often arise when one talks about end-to-end security.

The type of threats, respective protections required, the elements and activities within a network that need protection are the primary focus of X.805 standard. This standard, in itself fulfills the needs of a network's security and plays a crucial role in achieving the overall objectives of Information Security Management System (ISMS) of the organization.

### 1.2 Information Security Management System

International Standard Organization (ISO) has finalized a security management standard ISO 27001[2], and its expanded details "Code of Practice for ISMS" in ISO 27002[3]. Information Security Management System (ISMS) focuses on detailed aspects of entire information security within an organization.

Network security is one of the main concerns of the entire cycle of ISMS. The clause for network security expands on a detailed architecture standard ISO-18028 (IT Network Security), which is further divided into five parts. The first part, ISO-18028/1 elaborates the best practices used for managing network security. This document bridges the gap which arises between the administrators and the management personnel responsible for network security. ISO 18028-2 provides recommendation for the planning, design and implementation of the network security architecture and is in fact the ITU-T X.805 standard adopted by ISO.

The Fig 1 below illustrates the mapping between the ISO 27001, ISO 27002 and ISO 18028 standards. Both X.805 and ISO 18028-2 are used in this paper interchangeably to represent the standard security architecture.

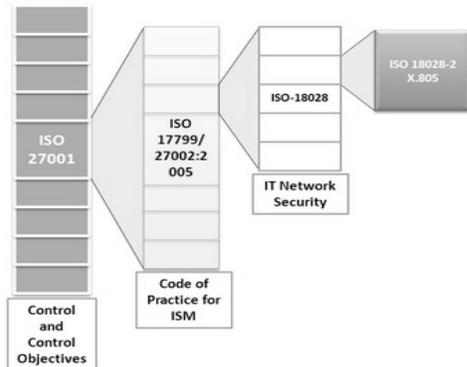


Figure 2. Mapping of 18028-2 to ISO 27001

ITU-T X.805 covers eight dimensions of security namely Authentication, Access Control, Non-repudiation, Data Confidentiality, Communication Security, Integrity, Availability and Privacy. These dimensions are then applied to a hierarchy of network elements and facilities spread across three layers namely Infrastructure, Services and Application layer. The activities that take place within the network are divided into three planes namely Management, Control and End User planes and lie within each of the three security layers described above. The above mentioned eight dimensions are applied to each intersection of these three planes across each of the three layers. The details of each dimension as described by the standard specify the requirements to be fulfilled by the network for compliance.

Despite all the deliberations made on security in the X.805 standard, the question that security professionals face is “*how to best implement X.805 standard?*” Although X.805 sets the requirements; yet, it does not describe the way to link and practically implement it on a network infrastructure for end-to-end security. This paper not only addresses the practical aspects in implementation of the ITU-T X.805 standard, but also exemplifies the process by taking into account the solution for Network Access Control.

### 1.3 Network Access Control

The current trend in the network security products indicates a strong inclination towards the development of integrated products. The trend offers an advantage of ease of management and lesser interoperability issues. The Network Access Control (NAC) solution is one such integrated product which is being increasingly used in support of authentication, access control and non-repudiation. It is a revolutionary security strategy that aims to ensure a safe network environment by unifying end point security technologies like antivirus, anti-spyware, security patches and updates along with end-user authentication strictly based

on compliance to an end point security policy. NAC allows network access based on assessment of behaviour to only compliant and trusted endpoint devices (PC or non-PC devices), and restricts the access of noncompliant devices until they become compliant.

In legacy network architecture, end users and devices are authorized based on as to who or what they are. NAC offers a more elaborate mechanism to make sure that only those end points become the part of network whose identity as well as status of health has been verified. A precise definition of NAC technology is somewhat challenging due to the evolving nature of this product; however, the principle objectives of the NAC can be viewed as [4]

- Endpoints that lack up-to-date antivirus, security patches or firewall softwares are denied network access to avoid contaminating the entire network.
- Network operators can define policies, regarding user roles, level of access and areas of network where those roles are permitted access.
- Identity management and user provisioning system acting as a part of NAC to ensure seamless implementation.

There are various versions of NAC available in the market, each from different vendor but all working toward the goals as defined above. Based on the functionality; however, there are two main categories of NAC. One is the software based, with the liberty to use any recommended type of underlying hardware and operating system, and the other is the hardware (Appliance) based, that avail a dedicated underlying hardware with an operating system desired for the same. Both the solutions have their pros and cons, and deciding the best to be implemented in the network depends upon the need and requirement of the network in question. The wide range of products available in the market makes it difficult to compare the products on point to point basis; yet, on the other hand, the same variety and wide range of products help to easily identify the one precisely suitable for the needs of an organization [4].

In this paper an attempt has been made to consider the practical implementation of X.805 standard across an organizational network while quantifying the security posture resulting through the addition of a NAC solution. The novelty of approach lies in quantification of the security posture. This quantification aims to bridge the gap that is often always present between the management and technical team dealing with network security within an organization.

## 2 Related Work

Although the standardized and systematic method of planning and assessment of network security is precisely defined in ITU-T X.805 standard, but not much work has

been reported in regards to issues related to practical implementation of this architecture in a real network. In [5] Richard, Ahmad and Kiseon have worked on the assessment of security natively offered by IEEE 802.15.4 standard for Low Rate Wireless Personal Area Networks in the light of X.805 architecture. Their assessment is more theoretical than practical and that too belongs to a different domain of network. Application of X.805 framework on a model of banking network has been attempted by Ramirez in [6]. This approach has only elaborated the standard specifically in terms of banking network requirements. It fails to address the methods of meeting those requirements and in fact also lacks the methodology for assessment of said network after the requirements are fulfilled. A similar theoretical concept is given in [6] explaining the effectiveness of X.805 architecture in terms of Return on Investment (ROI) for any enterprise. Both the works only subjectively address the application of X.805 standard as compared to the detailed objective analysis attempted by us. Moreover, our research has presented a generic method for planning, assessment and quantification of network security. It is equally applicable to any kind of network, including banking, education, health care and defense.

### 3 Proposed Work

In this section we discuss in detail our proposed approach to plan and assess the security of a network in the light of ITU-T X.805 architecture. The practical realization of X.805 architecture is described and also the possible quantifiable outcome in case of an example scenario when NAC is introduced in the network. The approach used in this analysis is to precisely break down the network into entities, and then fit those entities into each of the modules offered by X.805 and then assess the capability of each entity gained through NAC against the required capability as per X.805 specification. The deficiency, therefore leads to the introduction of external components that may be software or hardware, entity based or network wide to meet the standard requirements.

The quantitative results from this analysis are presented later in the section and are based on a legend in which a control can be in any of the four states as explained in Table I. NAC offers satisfactory compliance for a control, when it provides the required feature solely on the basis of its configuration. Partial compliance is offered when the control objectives are not fully met by the NAC solution and there is a requirement of additional effort to be put into it. This requirement may be hardware or a software upgrade to the end user device. The scope of this research does not precisely deal with the method of fulfilling this requirement. Whenever there is no need for an upgrade and merely policy enforcement can offer the compliance by forcing the element to pass through NAC, it is mentioned as Implicit Compliance. If NAC is totally unable to meet control objectives, it is marked as Not Applicable.

TABLE I. LEGEND AND DESCRIPTION

Key	Description
F	Satisfactory Compliance
P	Partial Compliance
I	Implicitly Compliance by virtue of network design
×	Not Applicable

In the perspective of NAC, the main focus is on end user devices. In today's network, most common end user devices are PCs, Laptops, and PDAs, non-PC devices like IP Phones and printers, which constitute the infrastructure layer of X.805 framework. Similarly, the services layer consists of those services running on any of the end terminal devices. Most common services used by a host PC are likely DHCP, POP3, SMTP, HTTP(S), NAC Agent, Antivirus, Firewall, Security Center and IPSec to name a few. The Application layer consists of any and all the applications present at the user terminal. The main focus is on the applications that directly interact with the lower layers without any need of other applications such as email clients, web browser or any other custom built.

TABLE II. NAC COVERAGE OF ACCESS CONTROL DIMENSION

Access Control Security Dimension									
Security Planes	Security Layers								
	Infrastructure			Services			Application		
	PC, Laptop	IP Phone	Printer	POP3, HTTP(S), NAC Agent, Antivirus, Firewall, Security Center	SIP-RTP, SRTP, SIP	LPD/LPR, IPP	Email Client, Web browser, custom Built.	Call Manager	Print Manager
End User	F	×	F	F	×	F	P	×	F
Control	F	×	F	F	×	F	P	×	F
Management	F	I	F	F	I	F	I	I	F

Table II shows the coverage of NAC across all layers and planes with respect to the Access Control security dimension. This dimension deals with the protection against unauthorized access to network resources. One method of achieving this objective is the role based access to elements of network infrastructure, like devices, services, application etc. The contribution of NAC for access control of each entity participating in our network is precisely laid down in table II.

NAC shows limited contribution towards Access Control of IP Phone and its services. IP Phones may or may not offer functionality of restricting access to its End User plane as a built-in feature but it's a fact that they lack contribution from NAC. Similarly the control plane containing all signaling and call control information may have an access control mechanism on part of the protocol or the IP Phone itself but

eventually it lacks contribution from NAC. Management activities performed on an IP Phone can however be implicitly forced to pass through NAC. In this scenario the personnel or devices originating management commands have to be authenticated and compliant to network policies. The same functionality of NAC can be perceived for the other entities in the network. NAC offers full coverage for Access control in case of PCs and printers with the exception of being partially supportive for applications running on the PCs. The applications depend on NAC for authorized access but defining levels of access within the application is entirely on the discretion of application itself. Access control levels for PCs, printers and their services can be defined in the user provisioning subsystem of the NAC solution. NAC fully ensures that only authorized personnel and devices have access to the End User, Control and Management Plane of the PCs, Printers and their services.

It is evident from the table II that NAC does not show much contribution to access control of IP Phones and its services, which might need device level software upgrades. The same approach has been followed for the analysis of other security dimensions described throughout this section.

TABLE III. NAC COVERAGE OF AUTHENTICATION DIMENSION

Authentication Security Dimension									
Security Planes	Security Layers								
	Infrastructure			Services			Application		
	PC, Laptop	IP Phone	Printer	POP3, HTTP(S), NAC Agent, Antivirus, Firewall, Security Center	SIP, RTP, SRTP, SIPS	LPD/LPR, IPP	Email Client, Web browser, custom Built.	Call Manager	Print Manager
End User	F	I	F	F	×	F	F	I	F
Control	F	P	F	F	P	F	F	×	F
Management	F	I	F	F	I	F	F	I	F

Table III explains the NAC coverage with respect to Authentication security dimension. In this regard X.805 specifies the requirement for confirming the validity of the claimed identities of the communicating entities. Table III shows the strength of a NAC solution in fulfilling the overall objectives of Authentication security dimension. NAC shows a great amount of contribution for authentication security dimension; however, for IP Phone and its protocols, it still offers a limited support. There is need of software level upgrade at device level for full compliance to X.805 requirements.

Table IV highlights the NAC coverage for Non-repudiation dimension. Preventing an individual or entity from denying a committed action is the key objective of this control.

TABLE IV. NAC COVERAGE OF NON-REPUDIATION DIMENSION

Non-Repudiation Security Dimension									
Security Planes	Security Layers								
	Infrastructure			Services			Application		
	PC, Laptop	IP Phone	Printer	POP3, HTTP(S), NAC Agent, Antivirus, Firewall, Security Center	SIP, RTP, SRTP, SIPS	LPD/LPR, IPP	Email Client, Web browser, custom Built.	Call Manager	Print Manager
End User	P	×	P	P	P	P	P	×	P
Control	P	×	P	P	P	P	P	×	P
Management	P	I	P	P	I	P	P	I	P

The overall impact drawn from a quick analysis of the table is that NAC only offers a part of non-repudiation and that is related to network joining, leaving, and connection establishment between two peers and their status at the said time. For full compliance there might be a need of a software upgrade at device or at network level.

Although X.805 also considers Data Confidentiality and Communication security dimensions; yet, they are not included in the analysis as NAC does not offer any capability to address these dimensions across any of the layers or planes of X.805 architecture. The services and applications may have their inherent capability to offer data confidentiality but this does not count for increasing the score of NAC solution. Communication security is also not addressed by the NAC solution. Intermediate network elements like switches, routers and gateways etc offers the capacity to ensure communication security and can be achieved by appropriate configuration and integration of these network elements.

TABLE V. NAC COVERAGE OF DATA INTEGRITY DIMENSION

Data Integrity Security Dimension									
Security Planes	Security Layers								
	Infrastructure			Services			Application		
	PC, Laptop	IP Phone	Printer	POP3, HTTP(S), NAC Agent, Antivirus, Firewall, Security Center	SIP, RTP, SRTP, SIPS	LPD/LPR, IPP	Email Client, Web browser, custom Built.	Call Manager	Print Manager
End User	P	×	×	P	×	×	P	×	P
Control	P	×	P	P	×	P	P	×	P
Management	P	×	P	P	×	P	P	×	P

Table V explains the data integrity dimension of X.805. NAC solution offers limited data integrity and that too is offered by

its capability of ensuring up-to-date and active anti-malware application and security updates. NAC solution currently lacks capability to ensure data integrity for IP Phone. Keeping in view the current trends in NAC solution, it can be foresighted that this capacity may be incorporated soon.

ITU-T X.805 deals with the constant availability of resources to the authorized users in Availability security dimension and specifies that there must be no denial of authorized access to the network resources whether it is infrastructure, service, application or stored information. NAC ensures partial availability for whole infrastructure as shown in Table VI. To ensure full compliance to availability security dimension there might be a need for network level upgrade like Intrusion Detection/Prevention systems, firewall appliances etc.

TABLE VI. NAC COVERAGE OF AVAILABILITY SECURITY DIMENSION

Availability Security Dimension									
Security Planes	Security Layers								
	Infrastructure			Services			Application		
	PC, Laptop	IP Phone	Printer	POP3, HTTP(S), NAC Agent, Antivirus, Firewall, Security Center	SIP, RTP, SRTP, SIPS	LPD/LPR, IPP	Email Client, Web browser, custom Built.	Call Manager	Print Manager
	End User	P	P	P	P	P	P	P	P
Control	P	P	P	P	P	P	P	P	P
Management	P	P	P	P	P	P	P	P	P

The privacy dimension of ITU-T X.805 states that the Information related to user or device activities must not be viewable to unauthorized entities and they must not be able to deduce the scope of activity performed on the element in question. Authentication and Access Control security dimensions are helpful in providing compliance to the Privacy security dimension and therefore yields the same insight as for Authentication and Access Control dimensions.

Table VII summarizes our analysis. Data Confidentiality, Communication Security and Privacy are not covered by NAC and hence are not included in the summary. The table provides a quantitative analysis of the overall coverage of X.805 dimensions by NAC and the same is depicted as percentage overall score of NAC solution in the Fig.2.

TABLE VII. ANALYSIS SUMMARY

Summary of NAC coverage of X.805 architecture				
End User Device	Covered	Partially Covered	Implicitly covered	Not Covered
PC, Laptop	15	29	1	0
IP Phones	0	13	11	21
Printers	18	25	0	2

It is evident from Table VII that the NAC solution offers good support for PCs, Laptops and Printers either by full compliance or by partially helping the entities achieve the said status. It is also a matter of fact that the contemporary NAC solutions single handedly provide a good amount of contribution towards achieving X.805 compliance. However, NAC shows limited support for IP Phones, which are rapidly proliferating the existing networks.

### Overall Score for NAC coverage

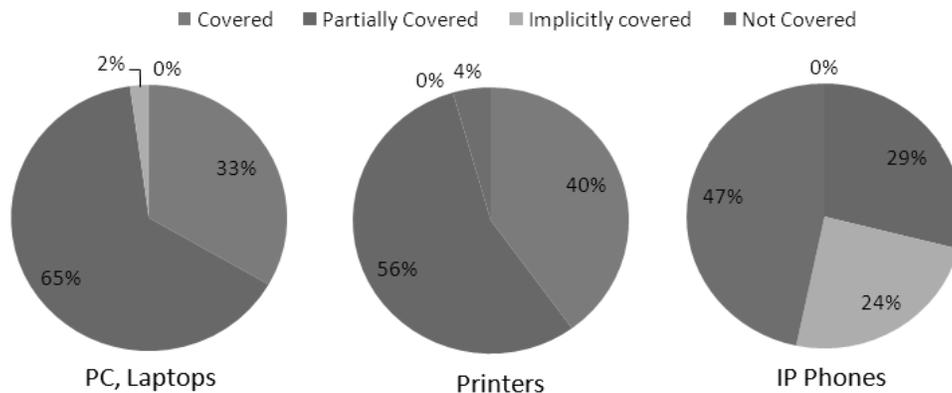


Figure 2. Overall Percentage score for NAC coverage of X.805

## 4 Conclusion

In this paper we have presented an approach for practical implementation of ITU-T X.805 standard for achieving end to end network security, while also considering the impact of a generic Network Access Control solution. The objective analysis as covered in Tables II – VI and summarized in Table VII quantifies the impact of NAC solution on overall network security and helps in developing better understanding of the assessment of a network's security. The analysis indicates that NAC contributes tremendously towards achieving X.805 compliance for any network. The remaining requirements where it lacks contribution can be satisfied by introducing other solutions and their contribution can be adjudged on the same pattern before bringing them into the network. By contributing a quantified security posture assessment approach, the proposed work is expected to open new horizons in network security management.

## 5 References

- [1] "ITU-T Recommendation X.805", <http://www.itu.int/itudoc/itu-t/aap/sg17aap/history/x805/x805.html>, [Dec. 11, 2009]
- [2] "Information security management systems – Requirements", [http://www.iso.org/iso/catalogue\\_detail?csnumber=42103](http://www.iso.org/iso/catalogue_detail?csnumber=42103), [Nov. 29, 2009]
- [3] "Code of practice for information security management", [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=50297](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50297) [Nov. 29, 2009 ]
- [4] J. Edwards. "The Essential Guide to NAC":, <http://www.itsecurity.com/features/essential-guide-nac-062308/>, Jun. 23, 2008 [Apr. 05, 2010].
- [5] A. O. Richard, A. Ahmad, K. Kiseon, "Security assessments of IEEE 802.15.4 standard based on X.805 framework". *Int. J. Secur. Netw.* 5, 2/3, 188-197 (Mar. 2010). DOI=<http://dx.doi.org/10.1504/IJSN.2010.032217>
- [6] D. Ramirez, . "Case study: ITU-T recommendation X.805 applied to an enterprise environment—banking" *Bell Labs Technical Journal*, Vol., Iss., 12-3, pp 55-64, Sep. 2007, DOI= <http://dx.doi.org/10.1002/bltj.v12:3>
- [7] A.R. McGee, U. Chandrashekhar, S.H. Richman, "Using ITU-T X.805 for comprehensive network security assessment and planning", in *Proc. Telecommunications Network Strategy and Planning Symposium. NETWORKS 2004*, 11th International, Vol., Iss., 13-16, pp273- 278, June 2004.

# A Framework for Online Document Attestation Using Encryption and Digital Watermarking

Mohammed A. El-Affendi

Department of Computer Science

College of Computer & Information Sciences

Prince Sultan University, Riyadh, Saudi Arabia

Associate Member of the Center of Excellence for Information Assurance, KSU

Muhammed Khurram Khan

Center of Excellence in Information Assurance

King Saud University

Riyadh, Saudi Arabia

**Abstract-***Document attestation is an important prerequisite in many transactions and processes. A good example is the attestation of university diplomas and school certificates, which is an important requirement for admission and employment processes. Despite the progress made in process and service automation, document attestation remains a completely manual operation, or partially automated at best. This paper proposes a simple framework for the full automation of the document authentication (attestation) process. The proposed approach is based on digital watermarking and aims to maintain the look and feel of stamps and signatures used in the process, while ensuring the authenticity of the stamped document. A summary of the main features of the authenticated document plus a carefully computed hash are first encrypted in the private key of the authenticating agency, and then embedded in the image of the associated stamp or signature. The main challenge, of course, is that watermark should be robust to scan-print and similar attacks.*

## 1 Introduction:

Document attestation is usually performed by authorized organizations or agencies that vary from one country to another. The common denominator between these organizations is that they are authorized and internationally recognized. The type of attestation depends on whether the country is a member of the Hague Convention or not [1]. Members of the Hague Convention use a legalization document called the "Apostille". The Apostille is a recognized document that certifies the authenticity of the document and the identity of the signatory. Non-members of the Hague Convention apply embassy legalization schemes which involve a chain of endorsements.

Usually the service provider verifies the validity of the certificate manually and then stamps the document using an authentic stamp recognized by all relevant stakeholders. In

many countries this process is still fully manual. The customer starts by paying the fees, then presents the certificate to an officer who verifies the stamps on the document and endorses them by placing a seal on the document. In some countries the process has been partially automated. At least the fees may be paid using some sort of e-Payment system.

This paper proposes a simple approach to automate the process of document authentication and attestation in a manner that emulates as much as possible the look and feel of manually authenticated documents. The proposed approach is based on cryptography and digital watermarking technology and requires the design of a secure protocol for online authentication and attestation of documents.

## 2 Related Work:

Digital watermarking is now a relatively mature field with many applications in a variety of areas such as copy right protection, document authentication, information hiding ...etc. [15]-[17]. Recently, some work has been done to develop watermarking schemes that are robust to scan-print processes and other types of attacks [2]-[14]. The main motivation behind these schemes is to apply digital watermarking to hard copies of documents and cards. The success of these approaches paved the way for many applications such as card authentication, license verifications..etc. The approach proposed in this paper relies on the findings of some researchers in this area, particularly the class of parameter invariant watermarking schemes [4].

### 3 The Problem

Current approaches for document attestation appear to be out of date and out of phase with current directions to automate all services and render them completely available online. The main obstacle is the requirement that some sort of recognizable stamp (Seal) and signature should appear on the document. For example, the experience in KSA is that all interested people should drive to a central office in the Ministry of Foreign affairs and queue for the service, not to mention the parking problem.

### 4 A Proposed Solution:

This paper proposes an automated solution that makes possible for individuals and agencies to authenticate documents using an online service. According to this solution, a user can submit a document for authentication over a secure connection using a client-side agent. On receiving the document, the authenticating agency checks the validity and authenticity of the document and the user (owner) using a variety of measures. If both the document and owner are authentic, the agency creates a summary of the key features of the document, generates a hash based on this summary, encrypts the summary plus the hash using its private key, embeds the encrypted digest in the stamp (seal) image, and inserts the watermarked image on a carefully generated image of the original document. The stamped image is then sent back to the user over a secure link.

The main assumptions and prerequisites may be summarized as follows:

- Attestation is performed by a an internationally recognized National Agency
- The Authenticating Agency is digitally certified with a known public key
- The document to be authenticated is issued by a known national agency (the source agency) that maintains digital copies of all issued documents
- The authenticating agency has full access to the digital database maintained by the source agency
- The submitting user must be registered with a valid national ID

Based on these assumptions, there are two possible scenarios

- 1- Optimistic: The authenticating agency has full online access to all relevant databases owned by third party organizations – The system will be able to validate submitted documents online, and in some cases generate a soft version of the authenticated document from the respective databases.
- 2- Conservative: No assumptions are made regarding online access for third party databases. In this case

the authentication officers and employees are responsible for validating submitted documents (manual intervention) and generating the authenticated document.

### 5 System Design and components:

The proposed system consists of three major components:

#### 5.1 The Document Submission Agent

- Provides an interface through which a client can submit a document for authentication
- The client fills a form and uploads an image for the document
- The document image should conform to the specifications required by the system
- Using the form, the client provides the following information:
  - Owner ID
  - Owner Name
  - Document ID
  - Document Type
  - Document Source
  - Document Issue Date
- On Clicking the submit button, the Document Submission Agent performs the following
  - Verify the validity of the document using appropriate databases
  - Verify the identity of the owner
  - Log the submission transaction
  - Create a database entry for the submitted document
  - Notify the client that his submission is successful, that his request is being processed and provide a system generated unique reference ID for follow up purposes
  - Notify the appropriate Authentication Officer
- Receive operation fees from the client using an appropriate payment system

#### 5.2 The Attestation Agent

On notification, the authentication officer performs the following through the Authentication Agent:

- Generate a cryptographic hash for the contents of the submitted document
- Create a digital watermark consisting of the following elements:
  - The document hash
  - The owner ID
  - Owner Name
  - The document ID

- Time stamp
- The word “Authentic”
- Authenticator domain name (URL)
- A Pass code
- Encrypt the watermark using the Private Key of the authentication organization
- Embed the encrypted watermark in the image of the Authentication Organization Stamp
- Generate a new image for the submitted document
- Place the watermarked stamp on the image of the submitted document
- Update the database entry for the submitted document and mark it as “authenticated”
- Send a completion message to the client and ask him to download the authenticated document

### 5.3 The Verification Agent:

The receiving agency can check the authenticity of a submitted document using a verification agent. The verification agent performs the following:

- Compute a hash for the major contents of the submitted document
- Scan the watermarked stamp
- Extract the watermark from the stamp
- Decrypt the watermark using the public key of the authenticating agent.
- Compare the recovered hash with the computed hash
- Verify that the recovered document ID and owner information is identical to the values shown by the document

## 6 The Marking Process

For the marking process, we employed a simple algorithm that hides watermark bits using the first moment of the lower diagonal of the DCT blocks. The host image is a plain BMP image. The Discrete Cosine Transform of the image is divided into 8X8 blocks. If the watermark bit=1, all elements of the lower diagonal of the corresponding 8x8 block are set to zero. If the watermark bit=0, all elements of the lower diagonal of the corresponding block are set to -100.

This ensures that the first moment of an embedding block is quite distinct from moments of normal blocks. Since, the effect of lower diagonal blocks on perceptual features of the image are less observable than other components, the embedding process will not affect the appearance of the image in an obvious way.

The embedding process may be described as follows:

```

if ((BlockCount > 30) && (w > 20) && (w < 200))
{
    if (kk < WaterMark.Length)
    {
        if (WaterMark[kk] == 1)
        {
            for (int i = 0; i < 8; i++)
            {
                for (int j = 0; j < 8; j++)
                {
                    if ((i + j) >= 7)
                    {
                        BDCT[i, j] = 0.0;
                    }
                }
            }
            done = true;
            BitCount++;
        }
        else
        {
            for (int i = 0; i < 8; i++)
            {
                for (int j = 0; j < 8; j++)
                {
                    if ((i + j) >= 7)
                    {
                        BDCT[i, j] = -100;
                    }
                }
            }
            done = true;
            BitCount++;
        }
    }
}
if (done) kk++;
}
}

```

The Digital Watermark Extraction process may be described as follows:

```

if ((BlockCount > 30) && (w > 20) && (w < 200))
{
    if (km < WaterMark.Length)
    {
        if ((LDAvg < 3) && (LDAvg > -1))
        {
            RMark = 1;
            RecovHash[km] = RMark;
            if (RMark == WaterMark[km])
                Success++;
            else FalsePos++;
            km++;
        }
        else
        {
            if (LDSum < -100)
            {
                RMark = 0;
                RecovHash[km] = RMark;
                if (RMark == WaterMark[km])
                {
                    Success++;
                }
                else FalsePos++;
            }
            km++;
        }
    }
}
}

```

## 7 Results

The watermarking process has been applied to 10 BMP images that have some resemblance to some seal images. The recovery rate from most images has been 100% for watermark consisting of 748 bits. In some few cases the recovery rate has been as low as 88%. Clearly, the recovery process depends on the richness and texture of the image. The results are still being analyzed to identify the conditions under which the watermarking process will be robust and reliable.

## 8 Conclusion

A simple framework for online authentication of documents has been proposed. The proposed approach is based on encryption and digital watermarking and aims to automate the document authentication process, while maintaining the look and feel of seals and stamps placed on original documents. The authenticated document is linked to the stamp placed on it using an encrypted watermark that summarizes the features of the document and includes a hash that may be used to detect any modifications in the document. The initial results are very encouraging. Further work is needed to increase the reliability of the process and tune it to fits real environments.

## Acknowledgement

The authors would like to thank the Center of Excellence in Information Insurance (Coeia) of King Saud University, Riyadh, KSA for funding the research that lead to the publication of this paper.

## 9 References

- [1] <http://perryvisa.com/webPages/legalDefine.php> "Legalization Tutorial"
- [2] Arya, Dhruv "A survey of Frequency & Wavelet Digital Watermarking Techniques", International Journal of Scientific & Engineering Research", Volume 1, issue 2, Nov 2010.
- [3] A.T. S. Ho and F. Shu "A Print-and-scan Resilient Digital Watermark for Card Authentication" In: Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the Joint, 2003.
- [4] X. Kang, J. Huang, W. Zeng. "Efficient General Print-Scanning Resilient Data Hiding Based on Uniform Log-Polar Mapping" IEEE Transactions on Information Forensics and Security, Vol. 5, No. 1, March 2010.
- [5] D. Cheng, X. Li, W. Qi, B. Yang "A Statistics-Based Watermarking Scheme Robust to Print-and-Scan" in the Proceedings of the IEEE International Symposium on Electronic Commerce and Security, pp. 894-898, 2008.
- [6] C. Y. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, and Y. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Trans. Image Process.*, vol. 10, no. 5, pp. 767–782, May 2001.
- [7] D. He and Q. Sun, "A practical print-scan resilient watermarking scheme," in *IEEE Int. Conf. Image Processing*, vol. 1, pp. 257–260, 2005.
- [8] K. Solanki, U. Madhow, B. S. Manjunth, S. Chandrasekaran, and I. El-Khalil, "Print-scan' resilient data hiding in images," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 4, pp. 464–478, Dec. 2006.
- [9] D. Zheng, J. Zhao, and A. E. Saddik, "RST-invariant digital image watermarking based on log-polar mapping and phase correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 753–765, Aug. 2003.
- [10] D. Zheng, Y. Liu, J. Zhao, and A. E. Saddik, "A survey of RST invariant image watermarking algorithms," *ACM Computing Surveys*, vol. 39, no. 2, Jun. 2007.
- [11] D. Zheng, Y. Liu, and J. Zhao, "A survey of RST invariant image watermarking algorithms," in *Proc. IEEE Canadian Conf. Electrical and Computer Engineering*, Ottawa, ON, Canada, May 7–10, pp. 2051–2054, 2006.
- [12] D. He and Q. Sun, "A RST resilient object-based video watermarking scheme," in *Proc. IEEE Int. Conf. Image Processing*, pp. 737–740, 2004.
- [13] J.-L. Dugelay, S. Roche, C. Rey, and D. Doerr, "Still-image watermarking robust to local geometric distortions," *IEEE Trans. Image Process.*, vol. 15, no. 9, pp. 2831–2842, Sep. 2006.
- [14] M. Alghoniemy and A. H. Tewfik, "Geometric invariance in image watermarking," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 145–153, Feb. 2004
- [15] Y. Govindharajan, S. Dakshinamurthi, Copyright Protection Protocols for Copyright Protection issues, *WSEAS Transactions on Computer Research*, Vol. 3, No. 4, , pp. 242-251, 2008.
- [16] V. Saxena, J. P. Gupta, A Novel Watermarking Scheme for JPEG Images, *WSEAS transactions on Signal Processing*, Vol. 5, No. 2, , pp. 74-84, 2009.
- [17] C. Atupelage, K. Harada, Perceptible Content Retrieval in DCT Domain and Semi-Fragile Watermarking Technique for Perceptible Content Authentication, *WSEAS Transactions on Signal Processing*, Vol. 4, No. 11, , pp. 627-636, 2008.

# Two-Argument Operations for Cryptographic Purposes

K. Bucholc

Institute of Control and Information Engineering, Poznan University of Technology, Poznan, Poland

**Abstract** - Two-argument operations are heavily used in data processing. In cryptography we either use simple operations, available in every computer instruction set, like addition, xor or more sophisticated operations like Galois field multiplication. In this paper we consider operations which are easy to implement for any bit vector length. Both linear and non-linear operations are taken into account. The operations are either reversible or semi-reversible. We focus on reversible operations to find the most interesting for hardware implementation.

**Keywords:** instruction set, operations, cryptography, semi-reversible operation

## 1 Introduction

Cryptography plays important role in many computer applications. Such operations as encryption, decryption, hash value computing are widely used.

Therefore any architectural improvement, which speeds up cryptographic application, influences the overall computer performance.

For example we can notice that, in server applications, switching on the encryption may substantially decrease performance.

The problem may be solved by including support for cryptographic operations in the computer hardware. There are two approaches possible: we can either extend the instruction set buy including support for concrete algorithm (e.g. AES), or supply more general instructions, which may be used in different cryptographic algorithms. In this paper we will focus on the second approach.

## 2 Two-argument operations

Two-argument operations are heavily used in data processing. Let us consider a block cipher implemented as permutation-substitution network (SPN). There are 3 operations in each round: subkey mixing, substitution, and permutation - fig.1.

Subkey mixing is a two-argument operation performed on the data block bits and the round key bits. Two-argument operations are also often used to compute permutation. Some simple operations available in any computer instruction set

may be used for this purpose. Another approach is to use more sophisticated operation.

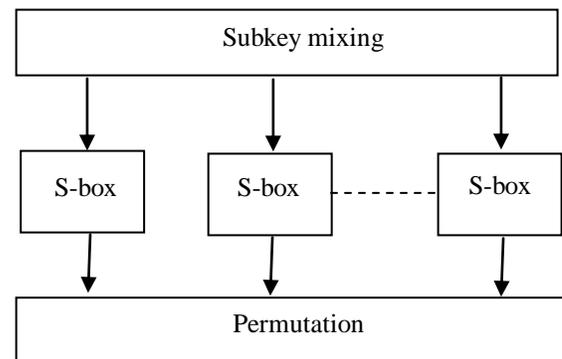


Fig.1. One round of a SPN block cipher

For example DES [3] algorithm uses xor operation as a subkey mixing operation, SAFER [4] uses xor and addition, whereas PP-1 [1] uses xor, addition, and subtraction.

The AES [2] algorithm may serve as an example of the second approach. It uses xor for key mixing, but more complicated Galois field multiplication to achieve permutation.

Let us consider a two-argument operation. It processes two  $m$ -bit arguments and produces one  $m$ -bit result. The operation may be presented as a table containing  $2^{2m}$   $m$ -bit vectors.

There are  $2^{m2^{2m}}$  different such tables. It means that we can define  $2^{m2^{2m}}$  different operations on  $m$ -bit arguments. Only some of these operations are useful. For example for encryption we need reversible (or at least semi-reversible) operations. Otherwise decryption would be impossible.

For small  $m$  (e.g. 8-bit) operations may be presented as a table. But for bigger  $m$  it is impractical or even impossible.

Usually for encryption and decryption we use **reversible operation** on finite field. It means that if  $z=f(x,y)$  there exist  $g$  and  $h$  such that  $x=g(z,y)$ ,  $y=h(z,x)$ .

To perform encryption and decryption the used operation not necessarily must be reversible. It is enough if the operation satisfies a weaker condition: if  $z=f(x,y)$  either there exist  $g$  such that  $x=g(z,y)$ , or  $h$  such that  $y=h(z,x)$ . In the first case we can restore  $x$  if we know  $z$  and  $y$ , but we

cannot restore  $y$  on the basis of  $z$  and  $x$ . Similarly in the second case only  $y$  may be restored. We will call such operations **semi-reversible**.

### 3 Operations under consideration

In this paper we consider only operations which (like xor or addition) may be easily implemented for any  $m$ . The structure of considered circuit is presented in fig. 2.

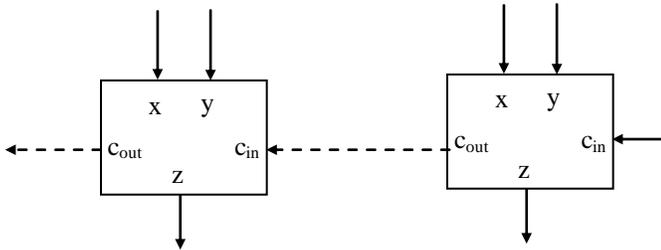


Fig. 2. The structure of considered circuit

It consists of a chain of elements. There are 3 one-bit inputs:  $x, y, c_{in}$  and 2 one-bit outputs:  $z, c_{out}$  in each element. Outputs are computed as follows:

$$z = f_1(x, y, c_{in}) \tag{1}$$

$$c_{out} = f_2(x, y, c_{in}) \tag{2}$$

The circuit is very similar to ordinary adder. The only difference lies in functions  $f_1$  and  $f_2$ .

Function  $f_1$  and  $f_2$  are 3-argument Boolean functions. There exist  $2^8=256$  such functions. It gives  $2^{16}$  combinations of  $f_1, f_2$ . Most of them are out of our interest.

Table 1. Balanced 3-argument functions and their linearity

0F	0	3C	0	69	0	99	0	C5	2
17	2	47	2	6A	2	9A	2	C6	2
1B	2	4B	2	6C	2	9C	2	C9	2
1D	2	4D	2	71	2	A3	2	CA	2
1E	2	4E	2	72	2	A5	0	CC	0
27	2	53	2	74	2	A6	2	D1	2
2B	2	55	0	78	2	A9	2	D2	2
2D	2	56	2	87	2	AA	0	D4	2
2E	2	59	2	8B	2	AC	2	D8	2
33	0	5A	0	8D	2	B1	2	E1	2
35	2	5C	2	8E	2	B2	2	E2	2
36	2	63	2	93	2	B4	2	E4	2
39	2	65	2	95	2	B8	2	E8	2
3A	2	66	0	96	0	C3	0	F0	0

First of all let us notice that, if we want the operation be reversible (or semi-reversible) the  $f_1$  function must be balanced. It means the numbers of zeros and ones in the function truth table must be equal. The  $f_2$  function may be any function. If we want to achieve good propagation properties the  $f_2$  must also be balanced.

In our research we used only balanced functions for  $f_1$  and  $f_2$ , plus zero function (all zeros in truth table) for  $f_2$ . There are only 70 balanced 3-argument Boolean functions.

In cryptography it often matters whether the function is linear or non-linear. Linear and non-linear functions are presented in table 1. We use the truth table in hex form for the function identification. For example 3-argument xor which truth table is {0,1,1,0,1,0,0,1} is described as 69.

### 4 Results

We investigated all possible combinations of functions presented in table 1 plus zero function for  $f_2$ .

Results are presented in table 2.

Table 2. Properties of synthesized operations

Functions	Number of occurrences	%
Reversible	300	6,036
Semi-reversible	1704	34,286
Non-reversible	2966	59,678
Total	4970	100,000

Table 3. Linear reversible operations

$f_1$	$f_2$	$f_1$	$f_2$	$f_1$	$f_2$	$f_1$	$f_2$
66	0F	69	0F	96	0F	99	0F
66	33	69	33	96	33	99	33
66	C3	69	C3	96	C3	99	C3
66	55	69	55	96	55	99	55
66	A5	69	A5	96	A5	99	A5
66	99	69	99	96	99	99	99
<u>66</u>	<u>69</u>	69	69	<u>96</u>	<u>69</u>	99	69
66	96	<u>69</u>	<u>96</u>	96	96	<u>99</u>	<u>96</u>
66	66	69	66	96	66	99	66
66	5A	<u>69</u>	<u>5A</u>	96	5A	<u>99</u>	<u>5A</u>
66	AA	69	AA	96	AA	99	AA
66	3C	69	3C	96	3C	99	3C
66	CC	69	CC	96	CC	99	CC
66	F0	69	F0	96	F0	99	F0

Only about 6% of synthesized operations are reversible. The  $f_1$  and  $f_2$  in reversible operations may be either linear or non-linear, but not both nonlinear.

Operations where both  $f_1$  and  $f_2$  are linear are presented in table 3. There are 56 such operations.

To investigate propagation properties of operations, 8-bit arguments were considered. For each combination of arguments, the result was computed. Then, consecutively, one bit of the argument was changed and result compared to the original one. The final result is the average number of changed bits divided by the bit vector length. The results vary from 0.125 (no propagation) to 0.5625 (best propagation).

Table 4. Propagation properties of selected operations

$f_1$	$f_2$	Changed bits %	Remarks
69	69	56,250	
69	87	34,375	
69	17	21,887	addition
69	2B	21,887	subtraction
69	35	20,337	
69	00	12,500	xor

Propagation properties of selected operations are presented in table 4. Operations with best propagation properties are marked in table 3.

(69,69)				
	0	1	2	3
0	0	3	2	1
1	3	0	1	2
2	2	1	0	3
3	1	2	3	0

(C3,69)				
	0	1	2	3
0	3	0	1	2
1	1	2	3	0
2	3	0	1	2
3	1	2	3	0

(6C,D2)				
	0	1	2	3
0	2	1	0	3
1	3	2	1	0
2	2	3	0	1
3	3	2	1	0

(1D,56)				
	0	1	2	3
0	0	0	0	0
1	2	3	0	1
2	0	0	2	2
3	2	3	2	3

Fig. 3. Examples of reversible (69,69), semi-reversible (C3,69)(6C,D2), and non-reversible (1D,56) operations

## 5 Conclusions

Considered operations may be used for processing of arguments which length exceeds the length of the processor word. In such case we use carry bit – similarly as in ordinary addition.

The best candidate for implementation in hardware is operation (69,69). It is linear, and it possesses the best propagation properties. There are also interesting reversible operations with nonlinear  $f_2$ . In this case further research is needed to investigate their properties (mainly linear and differential approximations).

Semi-reversible operations also deserve further research. They may be used for implementation of encryption and decryption circuits with very regular and simple structure. Another interesting research area is non-reversible functions. If we can establish which functions are most useful for cryptographic purposes we can implement them in hardware.

The main drawback of the circuit in Fig. 2 is long propagation time for carry. For practical implementation we should seek a solution for quick carry generation.

## Acknowledgment

This work was partially supported by the Polish Ministry of Science as a 2010–2013 research project.

## 6 References

- [1] Chmiel K., Grocholewska-Czurylo A., Stoklosa J., *Involutorial Block Cipher for Limited Resources*, Proc. Of the 2008 IEEE Global Telecommunications Conference, GLOBECOM 2008, IEEE eXpress Conference Publishing, 1852-1856, New Orleans 2008.
- [2] Daemen J., Rijmen V., *The Design of Rijndael.AES - The Advanced Encryption Standard*, Springer, 2002.
- [3] *Data Encryption Standard*, Federal Information Processing Standards Publication 46-3, NIST, Springfield, VA, October 1977.
- [4] Massey J. L., *SAFER K-64: One Year Later*, Preneel B. (ed.), *Fast Software Encryption*, LNCS 1008, Springer, New York, 1995, 212–241.

# Proof of Concept Implementation of Trustworthy Mutual Attestation Architecture for True Single Sign-on

Zubair Ahmad Khattak<sup>1,2</sup>, Jamalul-lail Ab Manan<sup>2</sup>, and Suziah Sulaiman<sup>1</sup>

<sup>1</sup>Computer & Information Sciences Department, Universiti Teknologi PETRONAS, Tronoh, Perak, Malaysia

<sup>2</sup>Advanced Information Security Cluster, MIMOS Berhad, Technology Park, Kuala Lumpur, Malaysia

**Abstract** – *To overcome computer network issues, user credentials for security and management have been used for single sign-on solutions and they have apparently helped to boost the security and usability of credentials. For true single sign-on solutions, where trusted entities are assisted by trusted platform module in the client and server platforms, they need a module that plays the role of authentication service provider. In this paper, we discuss the viability of such authentication service module and we also proposed an implementation of a Trustworthy Mutual Attestation Architecture for true single sign-on whereby the client and server first mutually attest themselves to establish a trust interconnection based on integrity verification. Our proposed approach is a trust-aware, flexible, and secure against malicious attacks and based on open standards.*

**Keywords:** remote attestation, true single sign-on, trusted computing, trusted platform module

## 1 Introduction

Internetworking (or Internet) on one hand has numerous advantages such as open access of services, flexibility to use different services, freedom to access these services from anywhere and anytime. However, the problem of managing many credentials (e.g. usernames and passwords) for accessing every resource or service for which they registered raises several security implications [13]. For instance, security and trust is fully entrusted to the system administrators of web service providers, who have access to customer's bank accounts, e-mail, corresponding addresses, credit card numbers etc. In addition to the risks of internal breach of security problem, trust is another important issue in open environment. Online banking services, online resource access scenario increases enterprises security and platform trust related concerns throughout the world. The conventional computer platforms are lacking in any mechanism to establish platform trust between the user and the servers of the service providers. In a typical real case scenario, if a system (might be enterprise or home user) is found running a malicious software or rootkit then it is considered compromised [1] and untrustworthy.

To overcome the aforementioned issues new mechanisms are needed to reduce uncertainties related to the security and trust in open environment. The one approach to diminish security implication in Single Sign-on (SSO) [2]

techniques were introduced, and in-depth analysis of SSO taxonomy is given in [3] and they have identified four main SSO categories (interested readers referred to [3]).

However, the majority of existing open environment schemes for instance Microsoft [4], Liberty Alliance [5] and Shibboleth [6] etc. involve third parties Authentication Service Provider (ASP) [3], [4] or Identity Provider (IDP) [6] which are referred in the later case. In all these schemes, both user and service provider put their trust on the ASP that they would provide correct user authentication assertion. The deficiency of trust relationship among interacting platforms would lead to security and trust hazards, discussed earlier, especially to access sensitive service or resource (medical, defence, bank). However, in this paper we focus our discussion only on true SSO i.e. that falls in local true SSO schemes, the other types of SSO schemes are out of our scope of discussion. In true SSO, user will not trust an entity that is under external control. The trusted component, which in this case, is the TPM within the user system, takes the role of Authentication Service (AS) [13].

In this paper we adopts Trusted Computing Group (TCG) [7] trust definition as, "the device, e.g. a Trusted Platform Module (TPM) [8], behave as expectedly for a specific purpose always in a particular way" [20]. In TCG [7], remote attestation mechanism is designed to measure and report the integrity of computer platforms (client and server). The TPM [8] chip is soldered onto the motherboard and works as a tamper-resistant microprocessor against software based attacks. Each TPM [8] has an Endorsement Key (EK) which is a manufacturer built-in key and uniquely identifies a particular platform. So the owner of platform creates Attestation Identity Key (AIK) which is a pseudonym key and being certified with Privacy-CA. The main purpose of the creation and certification of AIK is to perform a check whether a genuine TPM signs the PCR value or not when used in remote attestation

**Contributions:** Our contributions in this paper are as follows:

- We present a mutual attestation protocol between the user system and relying party, such that both parties mutually establish a platform trust by releasing platform configurations
- We incorporate the notion of integrity measurement in our architecture to verify integrity of both platforms

- We provide an implementation of our notion in the form of a SAML [9] based attestation request and response, which is an XML-based open standard for exchanging authentication and authorization data between security domains (i.e. IdP and SP)

The rest of the paper structured as follows. In Section 2, we provide background information about Trusted Computing, remote attestation and true SSO. In Section 3, we present the proposed architecture, proof of concept and implementation and we conclude the paper in section 4.

## 2 Background

### 2.1 True single sign-on

Pashalidis and Mitchell [13] provide an in-depth discussion how trusted computing technology can be adopted. The authors pointed it would be clearly an advantage to construct a true SSO scheme based on trusted entity, TPM, which removes the need for a third party, or at least limits the amount of trust that the user and SP are required to have in it. Their proposed design scheme is based on two key observations on the work in [14], (a) user authentication can be delegated to Trusted Platform and (b) identity credential, which are actually X.509 public key certificate carrying a unique serial number assigned by Privacy- CA, corresponding to the TPM identity. However, it is important to note that Pashalidis and Mitchell [13] did not provide any implementation of such system. In addition, in their proposed scheme, the client cannot assess SP platform integrity, whether it is really the one whom user trusts. Further, they introduce only the design scheme without its practical implementation, and their proposed scheme is quite complex.

### 2.2 Trusted computing

In early 2000s, Trusted Computing Platform Alliance (TCPA) [10] now known as Trusted Computing Group (TCG) [7] launched the notion of trusted platform. The basic idea of TCG to bring trust into traditional computing platforms through TPM chip and also it is a response to the security breaches [7], [11]. By definition TPM is a small co-processor chip that provides various security functionalities such as Random Number Generator (RNG), asymmetric key generation and shielded memory locations called Platform Configuration Registers (PCRs). These individual PCRs can store platform configurations in the form of cryptographic hashes of varied entities. The currently only supported mechanism for obtaining these cryptographic hashes is SHA-1 [12]. The entity may be varied such as BIOS, boot loader, kernel and application.

The PCR can only be manipulated via a mechanism known as *PCR extend*, for instance whenever a value has to be stocked in PCR, firstly, its hash is appended with the existing value of the PCR and secondly, SHA-1 of the resulting structure is stored back in the same PCR. The coupling of this technique with the SHA-1 irreversibility ensures that infinite number of configuration can be stocked in a single PCR and protects it through a tamper-resistance

facility. Later, the accumulated measurements submitted to a challenging party (client/server) to confirm a platform trustworthy configuration proof to which the TPM belongs. The AIK is a pseudonym key that is only accessible to the TPM and using to sign the accumulated values for vouching trust in it. The signing with AIK actually provides assurance that the accumulated guarantees it is really signed by a genuine TPM. However, the AIK private part never released outside of the TPM because of platform privacy concerns.

### 2.3 Remote attestation

Numerous remote attestation techniques were developed on the notion of load-time measurement, where a system (desktop, notebook, server, smart phone, iPod, iPhone, iPad etc.) measures the loaded components such as OS, applications, and hardware's. The TCG's traditional approach is limited to measuring the software system loaded before the OS such as BIOS, boot loader etc. Therefore, how to bring trust up to OS and application layers, Sailer et al. [1] presented Integrity Measurement Architecture (IMA) which utilizes the loaded-time measurement method to confirm the integrity of a remote system. The IMA was a first scheme to extend the TCG mechanism to within the operating system by measuring all the libraries and executables loaded during and after the Linux operating systems boot process. The hashes of applications it records at load-time and upholds a Stored Measurement Log (SML) in the Linux *securityfs*. In attestation process the SML and PCR values are reported to the challenger. The challenger can then analyze the load application at host and make a decision on it whether the interacting platform is trustworthy or untrustworthy.

## 3 Proposed Attestation Architecture

In open environment SSO models the ASP and SP also deals with the question, whether a user is authentic and the requested resource will be granted on basis of its correct authenticity proof. The work presented in this paper is a new, more trusted practical implementation, in a sense that both parties mutually attest, confirmation of platform authenticity, each other before invoking any further transaction.

In proposed architecture we assume a scenario whereby SP has some protected sensitive information, client has protected AuthN credential and Identification Service Provider (ISP) whose integrity must be verified before sending authentication assertion. As a proof of concept, we would like to show how interacting platform assures they are in trustworthy state and that no harmful activity running on each system. By this concept, we want to guarantee the ISP, which runs on user system will be in a state trusted by SP. Based on mutual trust the protected service will or will not be accessible to the user, and the protected user credentials will or will not be released to the service provider.

### 3.1 Proof of concept

In our proof of concept, for simplicity in rest of this paper we will use client and server instead of a user system

and service provider respectively. We construct an integrity service provider on both client and server. This service issues integrity request to the Corroboration Service (CS) that generate and send a nonce to the target platform, and validation daemons *VD-ClientA* & *VD-ServerA* located each on client and server responsible for the validation of received SML, PCR and Nonce. In addition, on client and server, integrity reporting demo module was also developed—which consists of attestation presenter that convey integrity information either to server or client as a response to the request they receive from integrity provider. The *PD-ClientA* & *PD-ServerA* is an entity that listens to the attestation requests to extract PCR from TPM and SML from *securityfs*. The attestation request and response between module and the daemon is achieved via Security Assertion Markup Language (SAML) [9] that further ensures scalability of our architecture into web services environment. The SML, PCR and Nonce verification process is initiated in XML [15] format that can benefit not only to communicate between internal systems but also external systems such as vendors, customers and partners.

As shown in Fig 1, in Step 1, client requests for a protected resource (e.g. a document (medical report)) on the server. After receiving the request and recognizing the target as being restricted the request handler at server side passes control to the integrity service module. In Step 2a and 2b, integrity service via CS makes an attestation request to the client which consists of a sign-challenge (e.g. a nonce). The, *PD-ClientA* that continuously listens to attestation request received this request, daemon-attestation presenter on client carries out Steps 3a, and 3b to assemble client platform integrity information. In Step 3a, the collection of Stored Measurement Log (SML) from */sys/kernel/security* is performed. In Step 3b it passes the received nonce to the Trusted Software Stack (TSS) (jTSS in our case) [16] along with a request for a *TPM\_Quote* over the value of 10<sup>th</sup> PCR. The SML and quote are accumulated in an attestation response and is returned to the module-corroboration service (Step-4) located at server.

The validation daemon located each on both server and client performs basic functionalities such as (a) SML validation: to make ensure the hashes in the SML received are validated against known good hashes in the validation database. This check confirms that no malicious software or rootkits running on client platform (Step-5), (b) PCR validation: it check the PCRComposite structure signed by valid TPM AIK. This ensures that the client TPM actually signed this PCR quote and the quote vouches for the authenticity of SML (Step-6), and (c) Nonce validation: The nonce signed by client that assures nonce freshness. However, due to flexibility of proposed approach we can include in future the Certificate validation module: it checks the signature of TPM and ensure a genuine TPM has already signed the quote (Step-7). In proposed architecture (see Fig. 1) for SML verification, a validation databases (both on client and server) must exists to verify the hashes returned by the client or server with corresponding executables (Step-8). The validation databases are constructed to collect individual known good hashes of all executables on target and challenger platforms.

Further, we are assuming that server request ISP integrity, software runs on client, for user credential. The ISP in our scenario also assumes purely a daemon/ service (with only username and password). The integrity of ISP has been assessed during the integrity challenge/ response Therefore, the client should, before delivering the user credential to the server side, first remotely attest server platform trustworthiness (Step-9). The attestation process will be performed in a similar fashion as per client platform. The steps 10a, 10b, 11a, 11b, 12, 13, 14, 15, 16 are similar to the steps 2, 3a, 3b, 4, 5, 6, 7, 8 given in Fig. 1 (see below).

### 3.2 Proof of concept implementation

We have implemented a prototype of a trustworthy mutual attestation protocol. We perform an experiment where each client and server (setup with java, Linux Ubuntu, Apache web server, *MySQL* database, Trusted Java, and machine embedded with the TPM). The *PD-ClientA*

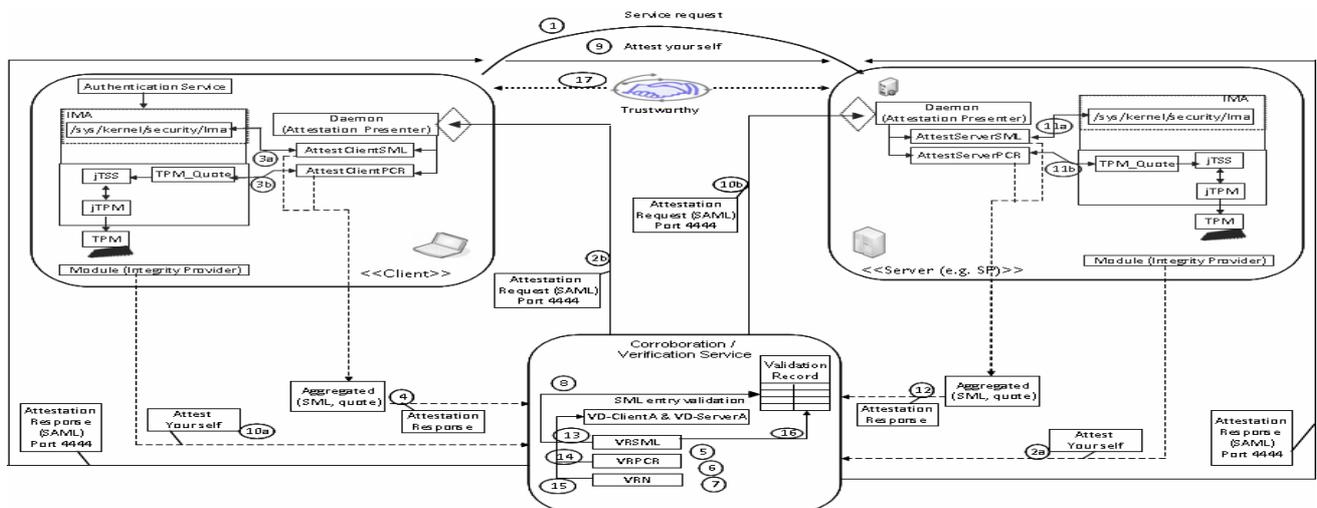


Fig1. Trustworthy mutual attestation architecture for true SSO

(Demeanour attestation validation daemon at client) and *VD-ServerA* (Demeanour attestation validation daemon at challenger) presents at client and server. The *run\_challenger* will execute a Java program on challenger (server) to generate and send a nonce to the target platform (client). The daemon (*PD-ClientA*) on target platform listening to this request will perform the PCR quote (over PCR-10), extract SML, PCR and send back the tokens to the challenger platform. The basic requirement for platforms, taking part in interaction, states that the sorted TPM on motherboard must be enabled, activated, owned and configured with IMA. Beside this the AIK must be generated to perform the quote operation. For the purpose of proof of concept implementation we generated AIK using the trusted Java (*jTSS* libraries, *jTPM* tools) [17] and then passes it to the *baaikExporter* class to generate an *AIK\_Pub* key. The EK will attest AIK when we execute the above process. However, the P-CA provided by IAIK [18] can also use for signing the AIK or design own P-CA that perform this process.

The CS on server and client sends a nonce encapsulated in attestation request to the target platform that communicate with TPM through *jTSS*. The *jTSS* is used to perform quote over 10th PCR and nonce value which guarantees nonce freshness and provides protection against replay attacks. The attestation presenter also forwards the SML to the server in an attestation response. At client and server the attestation response is handled by a class name - Client and Server class respectively. Each class implements internally two private functions which generates and adds-on a unique nonce in the attestation request. For this we create an interface which exposes public function for validation. Currently in our implementation we adopt three comprehension of this interface such as *ValidationofReceievedPCR (VRPCR)*, *ValidationofReceievedSML (VRSML)*, and *ValidationofReceievedNonce (VRN)*. For detail internal representation of *PcrCompositestructure* we refer the interested readers to [19]. The *PD-ClientA* & *PD-ServerA* acts as a service that located on both client and server and it provides seamless handling of attestation requests either broadcast from client or server and provides a single public function to perform the Attestation (*PA*) of the client or server when it receives the attestation request. Currently in our scenario we implement this class as an abstract interface used to provide attestation architecture presently composed of two attestors such as *AttestClientPCR*, *AttestClientSML* (at a client) and *AttestServerSML*, *AttestServerPCR* (at a server). The *AttestClientPCR* & *AttestServerPCR* provides a *tpm-signed* quote over the current PCR values and the nonce sent by corroboration service. Internally we are using AIK for signing the PCR values. Also the *AttestClientSML* and *AttestServerSML* the implementation of the attestation architecture which returns SML retrieved from *securityfs* of client or server. The collected trust token returned via the integrity provider to the CS and *VD-ClientA* & *VD-ServerA* (located each on client and server) validates it respectively.

To verify a system security and loaded executables trust chain, we place a rootkit on both client and server. On first

run both platforms successfully verified each other's platforms authenticity and trust chain of executables are successfully verified. After rootkit execution the system fails to establish the mutual trust because system is already compromised and trust chain collapse.

## 4 Conclusion

In this paper we have discussed issues related to how to synergise the strength of trusted computing, namely platform mutual attestation and integrity verification based on TPM, with authentication service provider module within a SSO collaborative open environment such as the Internet. As a proof of concept implementation we demonstrate a trustworthy mutual attestation architecture using remote attestation and trusted platform module. However, due to content constraint from organizers we excluded related work, experiment setup, results and discussion details related to the performance of the proposed approach.

## 5 References

- [1] R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn, "Design & implementation of a tcg-based integrity measurement architecture," in *Proc. 13th Usenix Conf. security symposium*, San Diego, 2004, pp. 223-238.
- [2] J. Hursti, Single Sign-on, Department of Computer Science, Helsinki University of Technology, 1997.
- [3] A. Pashalidis, and C. Mitchell, "A taxonomy of single sign-on systems," in *Proc. of 8th Aust. Conf. Information Security and Privacy*, 2003, pp. 249-264.
- [4] Microsoft, mMicrosoft.Net Passport Review Guide, November 2002.
- [5] Liberty Alliance, Liberty Alliance Project, <http://www.projectliberty.org>
- [6] M. Bruhn, M. Gettes, and A. West. It is 9:30 a.m. Do you know who your users are? Educause Quaterly, 2003.
- [7] Trusted Computing Group (TCG). <http://www.trustedcomputinggroup.org/>.
- [8] Trusted Platform Module (TPM) Specifications. <https://www.trustedcomputinggroup.org/specs/TPM/>.
- [9] Security Assertion Markup Language (SAML) v2.0. <http://www.oasis-open.org/committees/download.php/20645/sstc-saml-tech-overview-2%2000-draft-10.pdf>
- [10] Trusted Computing Platform Alliance (TCPA): Trusted Platform module Protection Profile. [http://www.commoncriteriaportal.org/files/ppfiles/pp\\_tcpatpmp v1.9.7.pdf](http://www.commoncriteriaportal.org/files/ppfiles/pp_tcpatpmp v1.9.7.pdf)
- [11] D. Challener, K. Yoder, R. Catherman, D. Safford, and L. Van Doorn. A Practical Guide to Trusted Computing. 2008.
- [12] D. Eastlake and P. Jones. US secure hash algorithm 1 (SHA1), 2001.
- [13] A. Pashalidis, C.J.Mitchell, "Single sign on using trusted platform," in *Proc. of 6th Conf. Information Security*, Bristol, 2003.
- [14] S. Pearson. 2003. *Trusted Computing Platforms: TCPA Technology in Context*, Prentice-Hall.
- [15] Extensible Markup Language (XML). <http://www.w3.org/XML/>
- [16] *Trusted Software Stack Specification*, Technical report, Trusted Computing, 2004.
- [17] Trusted computing for java (tm) platform. <http://trustedjava.sourceforge.org>.
- [18] Institute for Applied Information Processing and Communication (IAIK), Graz University of Technology. <http://www.iaik.tugraz.at/>
- [19] A. Lee-Thorp, "Attestation in Trusted Computing: Challenges and Potential Solutions," 2010.
- [20] TCG Specification Architecture Overview v 1.2, page 11-12. Technical report, Trusted Computing Group, April 2004.

# Presenting a New Approach for Predicting and Preventing Active/Deliberate Customer Churn in Telecommunication Industry

Majid Joudaki  
Islamic Azad University,  
Doroud Branch  
Doroud, Iran  
m.joudaki@gmail.com

Mehdi Imani  
Islamic Azad University,  
Science and Research,  
Qazvin Branch  
Qazvin, Iran  
m.imani@gmail.com

Maryam Esmaeili  
17shahrivar Higher  
Education Center  
Karaj- Iran  
laleh\_kimiya@yahoo.com

Mahtab Mahmoodi  
17shahrivar Higher  
Education Center  
Karaj- Iran  
mahtab\_empire2@yahoo.com

Niloofar Mazhari  
Allameh Dehkoda Higher  
Education Institute  
Qazvin, Iran  
nl.mazhari@gmail.com

**Abstract** - Like other storing and retrieval systems, the systems of telecommunication networks contain large databases which should store loads of data temporarily/permanently per moment. This data set includes data concerning customers and calls placed on the telecommunication networks. Extracting useful and relevant information buried under these vast telecommunication data sets is time-consuming and tiresome in emergency cases for considerations or finding occurred troubles a solution.

In this paper, we firstly study the different techniques of data mining in some cases such as fraud detection, customer profiling and marketing. Then we consider the specified algorithms of fraud detection systems, creating a customer profile for distinguishing or classifying business and residential customers and how to increase the number of members of these networks in marketing. In the end, a new method of estimating time of customers' giving-up for any reason and consequently membership contract cancellation called "customer churn" is presented to avoid customer churn in telecommunication companies.

**Keywords** - Fraud detection, Marketing, Customer profile, Customer Churn

## I. INTRODUCTION

The telecommunication networks generate and store a large amount of data such as Call Detail Data, information on each call placed and Customer Data, specifications of each customer. Manual analysis of this great amount of data in the urgent circumstances is not impossible but too difficult to handle and for this reason we need a kind of systems to identify unusual manners or illegal process for instance the fraudulent phone calls on time. But unfortunately the required information of these systems must be obtained from human resources that it was time-consuming in many cases. It is expected that data mining techniques are able to remove all the existing problems related to the above-mentioned matter in the telecommunication industry.

The below notes are considered in all the data mining applications studied in this paper:

- Scale of data inserted in the records of telecommunications databases.
- The raw data such as call detail data should be summarized by the useful summary features before effectively mined.

- Real-time performance: any model / rule must be applied in real-time for instance a fraud detection.

## II. TYPE OF TELECOMMUNICATION DATA

Before using data mining technology in any field, we should first understand which types of data will be evaluated. Two types of telecommunication data are necessary for the following applications and described completely in this section.

### A. Call Detail Data

Information on the call placed on a telecommunication will be saved as call detail record. Telecommunication databases contain a lot of call detail records generated in real time and typically should be kept online. They have sufficient information like the originating and terminating phone number, date, time and duration of each call and etc. For describing customer's behavior we need to combine these records with customer data and summarized in a single record. The features inserted in this summary will provide us with enough and short information on each customer immediately in case of need. Below is a sample of a customer's profile based on the received/dialed calls in a month.



Figure 1- A sample of customer's profile

We can profile residential and business customers easily according to item 5, 6, 9. According to item 9, telemarketing customers obviously call many different

area codes in comparison with residential customers and time-period for item 6 (8 am to 5 pm) used in distinguishing between residential and business calling patterns as working hour is typically 8 am to 5 pm.

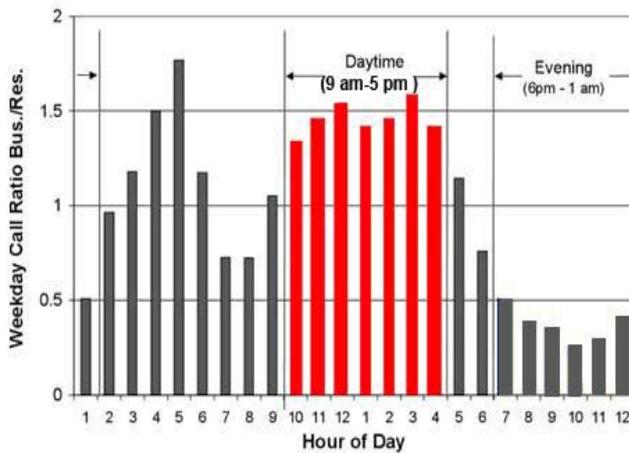


Figure 2- Comparison of Business and Residential Hourly Calling Patterns

The plots in Figure 2 for each weekday hour, *h*, compute the business to residential ratio as below:

$$h = \frac{\%Weekday\ Business\ Calls}{\%Weekday\ Residential\ Calls}$$

As already informed this figure also indicates that business customers place most of their calls during the period of 9 am to 4 pm in comparison with a residence. Sometimes a customer’s profile called signature [1] and should be updated in real-time hence they are necessarily simple and short to expedite the update process.

**B. Customer Data**

Like other businesses, telecommunication companies may have millions of customers which should have a database to maintain information such as name, address, service plan, contract details, credit score, family income and payment history. However the customer data is often combined with other data to improve results for instance this data as already stated used to supplement call detail data to identify phone fraud.

**III. DATA MINING APPLICATION**

Fraud is a serious problem for telecommunication companies because it causes billions of dollars to be lost each year. In March 2006 the Communications Fraud Control Association (CFCA) recently estimated that the global annual losses of fraud in the telecom sector are \$54 to \$60 billion which increased 52% from 2003 [8]. Fraud types are divided in two categories: Subscription fraud and Superimposition fraud.

**A. Subscription Fraud**

Fifty different categories has been by identified by The GSM Association but a TUFF recent survey in the UK found subscription fraud more current in the world.

This fraud happens when someone using a false identity opens an account in order to purchase services from operators but they have no intention to pay. One of the major issues in detecting subscription fraud is that they are unable to distinguish between it and simple bad debts made when genuine customers cannot afford to pay but as proved nearly % 30- 35 of all the bad debts are actually regards as subscription frauds. Like other kinds of crime, Criminal either use the identities of real people (135000 people were affected by identity theft in the UK in 2005 [8]) or alternatively create a new identity in order to reach their goals. Fraudsters usually prefer the second way to avoid being identified, make a fraudulent financial gain and avoid incurring financial liability.

**B. Subscription Fraud Detection System**

Current subscription fraud detection systems works based on detecting unusual behaviors such as not conforming to pre-set profiles, breaching pre-determined rules or exceeding call thresholds. When a customer’s account is flagged as suspicious by these systems investigators review data and search for the linked customers and accounts. This process is often manual and time-consuming and for this reason not economical for small scale fraud. Subscription fraudsters using multiple identities can successfully evade detection process for prolonged periods and their activities appeared as unrelated thus may not be detected easily.

With respect to the above, Telecommunication industry requires new techniques to identify networks of the fraudsters operating below the threshold of current systems and eventually flag potential new fraudsters much earlier to reduce loss if not prevent. Social network analysis is a new approach mapping even the invisible relationships of people, groups, computers, organizations and documents and analyzing networks of collaborating and apparently un-linked individuals and organizations. By combining multiple sources of data about people, their transaction with different organizations and their lives in general, it is possible to link groups of people into fraud networks. Since fraudsters in subscription frauds use multiple false identities often leave a footprint such as name, delivery address or bank account that can be identified and analyzed by social networks analysis approach. For instance signing up for a new phone contract makes a fraudster fill in the fields of name, home address and bank account details to pass the standard credit checks. Due to a valid bank account is required, the fraudsters sets up an apparently legitimate bank account to support his activities accordingly. Fraudster gangs usually share dozens of such bank accounts between their false identities as managing an individual account for each false identity is uneconomical for them. Thus all the customers using the same bank accounts or multiple similar names having the same delivery address will be identified by social network analysis and flagged as suspicious consequently. (See Figure 3, 4)

In the meantime if a customer is already associated with bad debts and according to the existing clues and

documents known as a fraudster, from now on investigators watch any new account intends to join this network carefully. Addresses contain multiple numeric/alpha tags such as Flat 23P, Flat 51G are useful in avoiding detection stage. Considering the above cases will terminate to a pattern represents that separate customers apparently sharing identity information. Although it is not conclusive evidence of fraud it just makes such cases suspicious and worthy of further investigation. As proved social network analysis- based solutions in comparison with present methods looking for large organized frauds are more economical.

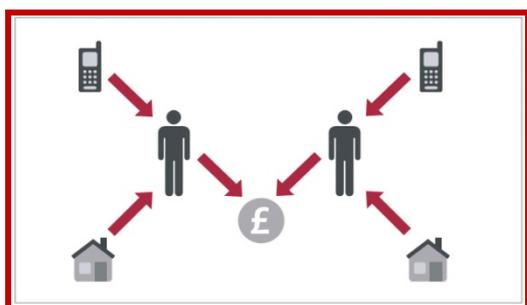


Figure 3- Multiple Customers with the same bank account

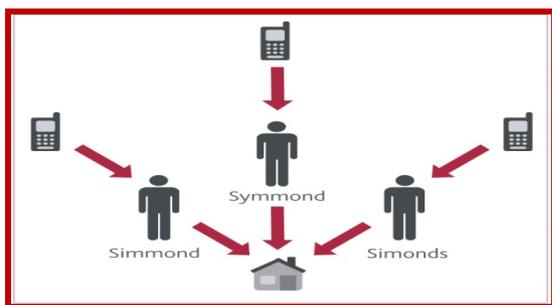


Figure 4- Multiple Similar Customers with the same delivery address

**C. Superimposition Fraud**

This fraud involves a legitimate account with legitimate activity but also includes some superimposed illegitimate activity by a strange person other than the account holder. Superimposition fraud is more critical for telecommunication industry and data mining applications usually identify this type of fraud by call detail data and once a fraud is detected or suspected it will ideally take a real-time such as immediately blocking the call/deactivate the account or start an investigation which ultimately results in a notification to customer for verifying the legitimacy of his/her account activity.

This type of fraud is commonly identified by comparing customer’s activities with what is explained in customer’s calling behavior (As described in Section 1-1). On the other words any deviation or unusual manner in a customer’s behavior is important for this data mining technique. In case of updating call details summaries in real-time, fraud cases will be detected soon after they occur. Since we cannot consider any change in a customer’s calling behavior as a fraud case it is sensible to compare the new calling behavior with a

set of fraud profiles and if it matches one of them, that account will be suspicious [1].

**D. Cellular Cloning Fraud**

Cellular cloning fraud means modification and illegitimate use of a service or cell phone and occurs when identity information associated with a cell phone is monitored for a while and then programmed into a second phone. Fraudster opens an account by illegally using the identity information of a legitimate customer and joins a telecommunication network to use services but the legitimate customer is not aware of this action and unfortunately it takes long to prove that she/he is not responsible for the debts.

Each cell phone manufactured in a factory has an exclusive Electronic Serial Number (ESN) and a phone number (MIN). Fraudsters monitor radio waves illegally for a time and can receive a valid ESN and MIN conveniently by a stimulator called cloning cell phone programmed to transfer a legitimate ESN and MIN. Since false ESN and MIN are entirely similar to the valid ones hence telecommunication systems are unable to distinguish which one is legitimate. For this reason in 1998 Wireless Phones Protection Law prohibited using/manufacturing/selling cloning software and hardware equipments because as per the offered statistics this type of fraud results in \$150 billion for each carrier annually. Finally authentication systems were developed in 90s and approximately remove this fraud from all the telecommunication companies. This data mining application analyzed large amounts of cellular call detail to identify patterns of fraud [2] used to generate the monitors and either of them watches a customer’s calling behavior with respect to a pattern of fraud. Soon after these monitors were fed into neural networks that raise an alert if there is sufficient evidence of fraud [1].

**IV. MARKETING**

One of the most well-known and successful promotions in the marketing section of telecommunication industry is MCI’s family and friends which initially launched in the United States in 1991 after starting to use long-distance services. It suggested to reduce calling fees when calls are placed to others in one’s calling circle. For example call detail records show that Steve Calls Jack, Jack Calls Joe, Juliet Calls Kate (No matter who makes a phone call).

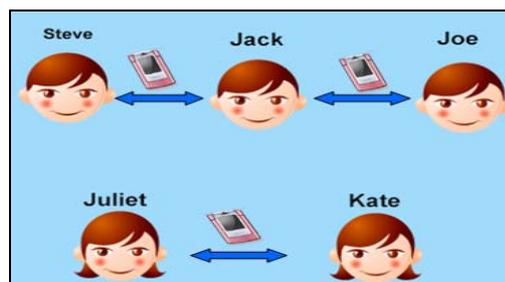


Figure 5- Samples of Calling Circles

Either of these samples is a separate calling circle. In a calling circle a couple may not have a direct telephone connection and they are connected to each other through telephone conversations indirectly (Joe have no telephone connection with Steve and vice versa). Telecommunication companies always ask customers to determine their calling circles in order to be included in particular discounts in their calling fees because when customer calls someone in his/her calling circle the telecommunication company considers the conversation lists monthly and make a particular discount in his/her invoice. Note the below algorithm in this regard:

```

Static int findNumCallingCircles (int [ ][ ]
conversations) {
Conversations [i] [j] == 1 if i have had a conversation
with j.
Conversations [i] [j] = 0 otherwise. } [6]
public class CallingCircle {
// Data: conversations[i][j] == 1 if i has a
conversation with j.
// It's zero otherwise.
static int findNumCallingCircles (int[][]
conversations) {
// INSERT YOUR CODE HERE.
return -1; }
// Test code -----
public static void main (String[] argv) {
int [][] A1 = {
{0, 0, 1, 0, 1},
{0, 0, 0, 1, 0},
{1, 0, 0, 0, 1},
{0, 1, 0, 0, 0},
{1, 0, 1, 0, 0} };
int n = findNumCallingCircles (A1);
System.out.println ("Test 1: 2 calling circles, your
answer: " + n);
int [][] A2 = {
{0, 0, 1, 1, 1},
{0, 0, 0, 1, 0},
{1, 0, 0, 0, 1},
{1, 1, 0, 0, 1},
{1, 0, 1, 1, 0} };
n = findNumCallingCircles (A2);
System.out.println ("Test 2: 1 calling circle, your
answer: " + n); } } [5]

```

This promotion initialized when market researchers noticed small sub-graphs in the graph of network activity and concluded that converting these graphs to an entire calling circle is more economical than adding new individual customers [3].

Despite MCI could generate a list of the people in each calling circle via call detail data but preferred its customers add the people of their calling circles themselves as it cared about privacy concerns. Another point is that MCI teaches loyalty and reminds its customers of faithfulness because if one of the members in a calling circle decides to leave this company due to competitors' considerable offers, the others persuade

him/her not to do so as it will endanger their benefits and cause their calling fees to be increased.

## V. CUSTOMER PROFILING

As already stated, Telecommunication industry like other kinds of business stores a large amount of data on customers and calls in order to describe the calling behavior of each customer. This information will be used in profiling process as their consumption patterns can be extracted from call detail data and as per these patterns we are able to profile the customers of a telecommunication company. Telecommunication companies use these profiles for knowing their customers and promotion of their marketing purposes. We have to combine and summarize these 2 types of data by the previous methods to expedite the mining process and make a classifier to distinguish between business and residential customers. SAS Enterprise Miner, a sophisticated data mining package which supports multiple data mining techniques, generated 2 below rules for classifying residential and business customers [4]. Neural networks are also used to predict the probability of a customer as residential/business based on division of the number of calls by time of a day (i.e 24 inputs one call per hour of the day). Evaluations show the accuracy of Rule 1, 88% and Rule 2, 70%.

Rule 1: *if* < 43% of calls last 0-10 seconds *and* < 13.5% of calls occur during the weekend *and* neural network says that  $P(\text{business}) > 0.58$  based on time of day call distribution *then* Business Customer

Rule 2: *if* calls received over two-month period from at most 3 unique area codes *and* < 56.6% of calls last 0-10 seconds *then* Residential Customer [7]

Given that telecommunication companies are able to consider a pattern even for non-customers observed in their customers' calling circles to extend this mining process in the marketing section overall and at high level as they have access to call detail data. However this action obliged to be accompanied by special legal restrictions regarding accurate use of the relevant data.

## VI. Proposed method

Nowadays Customer Churn Phenomenon considered as a critical matter and a concern for all the telecommunication companies in the marketing section and happens when a customer intends to cancel the membership contract with one's operator and switch to competitors due to unsatisfaction with the quality of service, high costs, bad support, no reward for customer loyalty, privacy concerns, competitors' remarkable offers and etc. Rewarding \$50 or \$100 by the Long-distance telecommunication companies in return for signing up is known as the worst type of Customer Churn ever occurred in the recent years as it caused customers to switch their service providers regularly to gain the rewards. Since Customer Churn Prediction can keep existing customers and prevent sales losses and terrible financial consequences, this paper is presenting a new

method for the prediction of Active/Deliberate Customer Churn using data mining techniques along with a special data set extracted from each customer's profile which is already illustrated separately on Section 1.1. If telecommunication companies have the ability to estimate the time of customers' giving-up, they can prevent Customer Churn on time by considerable offers to them. Our suggestion is drawing a special diagram which shows increase or decrease amount of a customer's present activity in comparison with one last month.

TYPE OF CUSTOMER	WHCS	DCS	COSD	RCS	ACS	WECS
Business	✓	✓	✗	✓	✓	✗
Residence	✗	✓	✓	✓	✗	✓

It is worthy to note that the required data set of this diagram for a residential customer somehow differs from a business one.

**ABBREVIATIONS:**

RCS: Total grand of Received Calls. DCS: Total grand of Dialed Calls. WHCS: Total grand of Working Hour Calls. ACS: Total grand of Exclusive Area Codes. WECS: Total grand of Weekends Calls. COSD: Total grand of Calls on Special Dates (Conditional). INC/DEC: Increase / Decrease

**NOTE 1:** These parameters must be computed and updated per month.

**NOTE 2:** WHCS, ACS and WECS as explained above are the most significant features used in profiling customers and distinguishing between Business and residence.

**NOTE 3:** COSD refers to calls (received/dialed) placed on special dates such as New Year, Birthday, Religious Festivals, Wedding Anniversary or whatever is asked from a customer while signing up. It is used for an exact consideration on a customer's activity but obviously conditioned that the customers have membership contract with a telecommunication company for more than one year in order to evaluate customer's activity on these dates in the present year in comparison with last years. Otherwise it will be blank and not included in the diagram.

A sample of this monthly diagram is drawn for a business customer as per the amounts of the below columns (See Table 1 and Figure 6).

Month	RCS	DCS	WHCS	ACS	SUM	INC/DEC
Jan	5	55	60	30	150	
Feb	36	70	80	22	208	+%38
Mar	70	85	90	25	270	+%29
Apr	96	105	150	10	316	+%17
May	50	135	100	10	295	-%6
Jun	63	144	207	17	224	-%24
Jul	100	56	110	32	298	+%33
Aug	136	80	73	50	339	+%13
Sep	140	100	200	16	456	+%34
Oct	110	44	105	5	264	-%42
Nov	50	62	67	11	190	-%28
Dec	19	13	15	14	61	-%67

Table 1- A sample of Monthly Activity Table

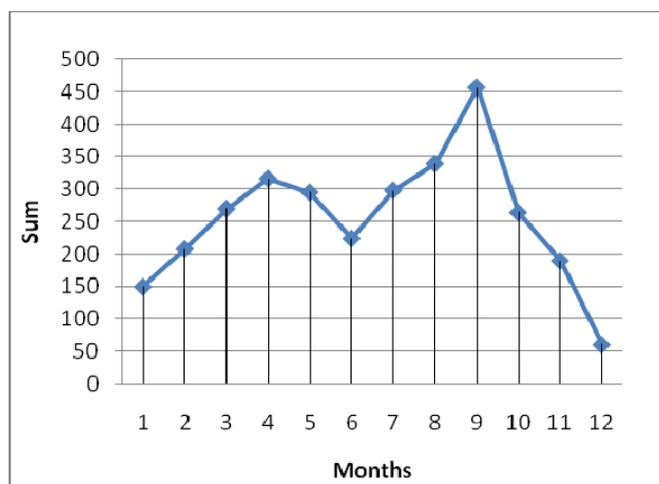


Figure 6- A sample of Monthly Activity Chart

**A. HOW TO CALCULATE THE MONTHLY INCREASE/ DECREASE**

The amounts of the above- mentioned parameters generated for per month are added together and the result inserted in SUM column. Then SUM of the present month should be subtracted from SUM of the previous one. The subtraction answer will be divided by SUM of the previous one as comparison base. The final outcome will be called the increase/decrease percentage of customer's activity in comparison with the last month. If the present SUM is more than what is calculated in the previous row, the relevant percentage will show an increase. Otherwise it will be considered as a decrease and it will be % 0 in case of equality. Meanwhile decimal answers have been rounded as usual for more convenience. For instance it is calculated for February in Table 1 as below.

$$SUM = RCS + DCS + WHCS + ACS$$

$$SUM = 36 + 70 + 88 + 22 = 208$$

$$208 (SUM \text{ of Feb.}) - 150 = 58$$

$$58/150 = \%38 \text{ (Inc)}$$

**NOTE 4:** This diagram will be started drawing from the second month of signing up as there is no base comparison for the first month (As you see nothing mentioned for Jan. in Table 1). But for the customers who have membership contract with a telecommunication company for more than one year, the first month of New Year will be compared with the month last of last year. However the diagram will be drawn separately and they have a link just on this matter.

**NOTE 5:** This approach is invented for one-year contract as it can consider a customer's activity during a year more exactly to understand whether she/he is probably a churner or not.

After this activity diagram becomes complete it will be sent to a neural network to determine whether that customer is a churner or not or even estimate the time of his/her giving-up if so. As you know Neural Networks always make decision according to past experience and for this reason we are going to use their feature to predict and prevent Customer Churn. Notice that all the diagrams of the above kind should be stored in telecommunication companies as Churners' Profiles durably and regularly because they can determine next churners based on these profiles. These networks are composed of 3 layers as below: Input Layer, Hidden Layer containing calculating nodes and Output Layer (See Figure 7).

The diagram initially enters to the input layer as an input and this layer will submit it to the hidden layer. Then the calculating nodes will compare it with a set of Churners' Diagrams. This comparison can be based on *Number of hotspots (Decreasing points), Increase/decrease amount, Sequence of increase/decrease points* and etc. Each calculating node will indicate the similarity percentage of the under-consideration diagram with the relevant Churners' diagram which is called weight. The highest weight refers to that Churner's diagram which is the most similar with the under-consideration diagram of all and will be considered by Neural Network to determine the probability of being a Churner and estimate the time of contract cancellation. However the weight goes higher, it will increase the value of investigation on that customer.

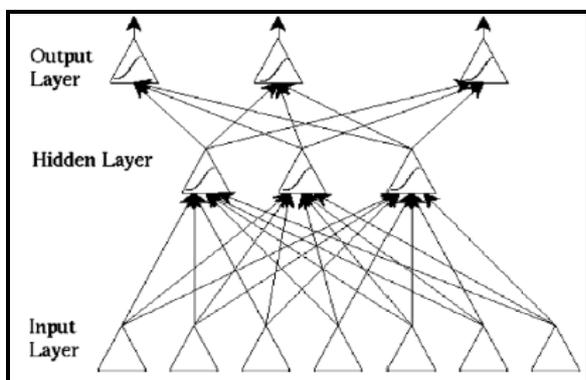


Figure 7- A Three-Layer Neural Network

**B. HOW TO PREVENT TO CUSTOMER CHURN**

If a customer is known as a churner, a telecommunication company should do its best to keep him/her by considerable offers and persuade not to leave. These offers must suit the customer otherwise it's no use. For this reason we recommend operators to study and use customer demographics first which includes age, gender, level of education, social status, geographical data and etc. For instance if a probable churner is a young person, we have better offer him/ her Free High-Speed Internet Services for a specified period and something like this while this offer may be unsuitable for an elder. Another important point is that sometimes customers' giving-up reasons are not just because of competitors' remarkable offers and possibly a telecommunication company provide its customers with high-technology services and facilities but pay no attention to security matter or privacy concerns although customers usually care about these matters. If a customer leave one's telecommunication service provider due to lack of sufficient security, that company should compensate the occurred losses via beneficial offers. This action is mutually useful as it will cause a telecommunication make its security aspects stronger for keeping its customers and customers are not forced to pay imposed debts resulting from frauds as well as operating good advantages. Also there is no need to change their operators regularly.

**C. ADVANTAGES**

- *Possibility of exactly considering sudden changes in a customer's activity.*  
If this diagram drawn annually is stored in telecommunication databases for each year separately, we are able to make definite decision on sudden/ unpredictable changes happened in a period of time and estimate the probability of being churner for a customer more exactly with reference to the diagrams to understand whether they are natural and usual in comparison with previous years because any single change is not a single of Customer Churn as already proved. (In case a customer is a multiple-year member.)
- *Showing increasing/decreasing amount of customers' activity diagrammatically by simple calculations without involving in a tremendous amount of digital data.*
- *Decreasing the required data set for analysis to 4 or 5 parameters as already mentioned.* Specially for Neural Networks which need a large volume of data set and a lot of time in order to calculate a reasonable weight age for the predictor attributes.[Vladislav Lazarov & Marius Capota]
- *This approach will be done for Business and residential customers separately through the most significant features of their profiles.*

[8] A white paper of Detica Information Intelligence; *Detecting telecoms subscription fraud*, 2006

## VII. CONCLUSION

In this paper we stated how to use data mining techniques in telecommunication industry and introduced 2 kinds of telecommunication data (Call Detail Data & Customer Data) along with the most current applications of data mining for Fraud Detection, Marketing and Customer Profiling. Then we studied the algorithms of Fraud Detection Systems, building customer's profile for distinguishing between a business and residence and how to increase number of a telecommunication companies. In the end Customer Churn is considered and a new approach also offered to predict and prevent it. Needless to say that this approach is an induced model, it means that probably it doesn't predict this phenomenon or estimate the giving-up time definitely but we can specify and flag the discovered cases in a telecommunication network as they are valuable for investigation. With respect to the daily-increasing growth of various telecommunication services by different operators, existing challenges for these companies and competition for attracting more customers, investigators and researchers are always keen on this scope for more study.

## VIII. REFERENCES

- [1] Cortes, Corinna; Pregibon, Daryl; *Signature-based methods for data streams*, Data Mining and Knowledge Discovery, ISSN: 1384-5810, Kluwer Academic Publishers Hingham, MA, USA, p.p. 167-182, 2001.
- [2] Fawcett, Tom; Provost, Foster; *Activity monitoring: Noticing interesting changes in behavior*. Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999
- [3] Han, Jiawei; Altman, Russ B.; Kumar, Vipin; Mannila, Heikki; Pregibon, Daryl; *Emerging scientific applications in data mining*. Communications of the ACM, ISSN: 0001-0782, ACM, New York, USA, p.p. 54 – 58, Aug. 2002
- [4] Kaplan, Haim; Strauss, Martin; Szegedy, Mario; *Just the fax — differentiating voice and fax phone lines using call billing data*. Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms, ISBN: 0-89871-434-6, Society for Industrial and Applied Mathematics Philadelphia, USA, p.p. 935 – 936, 1999
- [5]<http://www.seas.gwu.edu/~simhaweb/cs153/lectures/module7/examples/CallingCircle.java>
- [6]<http://www.seas.gwu.edu/~simhaweb/cs153/lectures/module7/examples/CallingCircle.html>
- [7] Gary M. Weiss; *DATA MINING IN TELECOMMUNICATION*, Department of Computer and Information Science, Fordham University

# An Automated Signature Generation Approach for Polymorphic Worms Using Factor Analysis

Mohssen M. Z. E. Mohammed<sup>1</sup>, H. Anthony Chan<sup>2</sup>, Neco Ventura<sup>2</sup>, Mohsin Hashim<sup>3</sup>, and Izzeldin Amin<sup>3</sup>

<sup>1,2</sup> Department of Electrical Engineering, University of Cape Town, Cape Town, Western Cape, South Africa

<sup>3</sup> Faculty of Mathematical Sciences, University of Khartoum, Khartoum, Khartoum, Sudan

Emails: m\_zin44@hotmail.com; h.a.chan@ieee.org; neco@crg.ee.uct.ac.za ; mohsinhashim@yahoo.com; izzeldinamin@yahoo.com

**Abstract** - Internet worms pose a major threat to Internet infrastructure security, and their destruction will be truly costly. Therefore, the networks must be protected as much as possible against such attacks. In this paper we propose automatic and accurate system for signature generation for unknown polymorphic worms. We have designed a novel double-honeynet system, which is able to detect new worms that have not been seen before. We apply Factor Analysis to determine the most significant substrings that are shared among polymorphic worm instances and use them as signatures. The system is able to generate accurate signatures for polymorphic worms.

**Keywords:** Polymorphic Worms, Honeynet, IDSs.

## 1 Introduction

The yearly growth of internet worms increasingly threatens the availability and integrity of Internet-based services. Internet worm is a malicious program that spreads automatically among hosts on a network by exploiting various vulnerabilities present on those hosts. A computer worm differs from a computer virus in that a computer worm can run itself. A virus needs a host program to run, and the virus code runs as part of the host program. A polymorphic worm is a worm that changes its appearance with every instance [1]. It has been shown that multiple invariant substrings must often be present in all variants of worm payload. These substrings typically correspond to protocol framing, return addresses, and in some cases, poorly obfuscated code [8].

Intrusion detection is the process of monitoring computers or networks for unauthorized entrance, activity, or file modification. IDS can also be used to monitor network traffic, thereby detecting if a system is being targeted by a network attack such as a denial of service attack. There are two basic types of intrusion detection: host-based and network-based. Each has a distinct approach to monitoring and securing data. Host-based IDSs examine data held on individual computers that serve as hosts, while network-based IDSs examine data

exchanged between computers. There are two basic techniques used to detect intruders: Anomaly Detection and Misuse Detection (Signature Detection). Anomaly Detection is designed to uncover abnormal patterns of behavior, the IDS establishes a baseline of normal usage patterns, and anything that widely deviates from it gets flagged as a possible intrusion. Misuse detection (signature detection) commonly called Signature Detection, this method uses specifically known patterns of unauthorized behavior to predict and detect subsequent similar attempts. These specific patterns are called signatures [16, 17].

Our research is based on Honeypot technique. Developed in recent years, honeypot is a monitored system on the Internet serving the purpose of attracting and trapping attackers who attempt to penetrate the protected servers on a network. Honeypots fall into two categories. A high-interaction honeypot such as (Honeynet) operates a real operating system and one or multiple applications. A low-interaction honeypot such as (Honeyd) simulates one or multiple real systems. In general, any network activities observed at honeypots are considered suspicious [1, 10].

Security experts need a great deal of information to perform signature generation. Such information can be captured by tools such as honeynet. Honeynet is a network of standard production systems that are built together and are put behind some type of access control device (such as a firewall) to watch what happens to the traffic [1]. We assume the traffic captured by honeynet is suspicious. Our system reduces the rate of false alarms by using honeynet to capture traffic destined to a certain network.

This paper is organized as follows: Section 2 discusses the related work regarding automated signature generation systems. Section 3 reviews anatomy of polymorphic worms. Section 4 introduces the proposed system architecture to address the problems faced by current automated signature systems. Signature generation for Polymorphic Worm using Factor Analysis will be discussed in section 5. Section 6 concludes the paper.

## 2 Related work

Honeypots are an excellent source of data for intrusion and attack analysis. Levin et al. described how honeypot extracts details of worm exploits that can be analyzed to generate detection signatures [3]. The signatures are generated manually.

One of the first systems proposed was Honeycomb developed by Kreibich and Crowcroft. Honeycomb generates signatures from traffic observed at a honeypot via its implementation as a Honeyd [5] plugin. The longest common substring (LCS) algorithm, which looks for the longest shared byte sequences across pairs of connections, is at the heart of Honeycomb. Honeycomb generates signatures consisting of a single, contiguous substring of a worm's payload to match all worm instances. These signatures, however, fail to match all polymorphic worm instances with low false positives and low false negatives.

Kim and Karp [6] described the Autograph system for automated generation of signatures to detect worms. Unlike Honeycomb, Autograph's inputs are packet traces from a DMZ that includes benign traffic. Content blocks that match "enough" suspicious flows are used as input to COPP, an algorithm based on Rabin fingerprints that searches for repeated byte sequences by partitioning the payload into content blocks. Similar to Honeycomb, Autograph generates signatures consisting of a single, contiguous substring of a worm's payload to match all worm instances. These signatures, unfortunately, fail to match all polymorphic worm instances with low false positives and low false negatives.

S. Singh, C. Estan, G. Varghese, and S. Savage [7] described the Earlybird system for generating signatures to detect worms. This system measures packet-content prevalence at a single monitoring point such as a network DMZ. By counting the number of distinct sources and destinations associated with strings that repeat often in the payload, Earlybird distinguishes benign repetitions from epidemic content. Earlybird, also like Honeycomb and Autograph, generates signatures consisting of a single, contiguous substring of a worm's payload to match all worm instances. These signatures, however, fail to match all polymorphic worm instances with low false positives and low false negatives.

New content-based systems like Polygraph, Hamsa and LISABETH [8, 11 and 12] have been deployed. All these systems, similar to our system, generate automated signatures for polymorphic worms based on the following fact: there are multiple invariant substrings that must often be present in all variants of polymorphic worm payloads even if the payload changes in every infection. All these systems capture the packet payloads from a router, so in the worst case, these systems may find multiple polymorphic worms but each of them exploits a different vulnerability from each other. So, in this case, it may be difficult for the above systems to find

invariant contents shared between these polymorphic worms because they exploit different vulnerabilities. The attacker sends one instance of a polymorphic worm to a network, and this worm in every infection automatically attempts to change its payload to generate other instances. So, if we need to capture all polymorphic worm instances, we need to give a polymorphic worm chance to interact with hosts without affecting their performance. So, we propose new detection method "Double-honeynet" to interact with polymorphic worms and collect all their instances. The proposed method makes it possible to capture all polymorphic worm instances and then forward these instances to the Signature Generator which generates signatures, using a particular algorithm.

An Architecture for Generating Semantics-Aware Signatures by Yegneswaran, J. Giffin, P. Barford, and S. Jha [9] described Nemean, Nemean's incorporates protocol semantics into the signature generation algorithm. By doing so, it is able to handle a broader class of attacks. The coverage of Nemean is wide which makes us believe that our system is better in dealing with polymorphic worms specially.

An Automated Signature-Based Approach against Polymorphic Internet Worms by Yong Tang and Shigang Chen[10] described a system to detect new worms and generate signatures automatically. This system implemented a double-honeypots (inbound honeypot and outbound honeypot) to capture worms payloads. The inbound honeypot is implemented as a high-interaction honeypot, whereas the outbound honeypot is implemented as a low-interaction honeypot. This system has limitation. The outbound honeypot is not able to make outbound connections because it is implemented as low-interaction honeypot which is not able to capture all polymorphic worm instances. Our system overcomes this disadvantage by using double-honeynet (high-interaction honeypot), which enables us to make unlimited outbound connections between them, so we can capture all polymorphic worm instances.

## 3 Polymorphic worms

Every worm has a unique bit string which can be used to identify the worm (i.e. all instances of the worm in the network have the same bit string representation). Hence worms can be detected easily using simple signature based techniques (i.e. by comparing the network packets against a database of known signatures). Polymorphic worms, on the other hand, change their representation before spreading i.e. each instance of a polymorphic worm will have a different bit stream representation [4].

### 3.1 Polymorphic worm techniques

- Encryption

Here, the worm encrypts its body with a random key each time before spreading. A small executable code is then attached to the body of the worm. This executable code is responsible for encrypting the encrypted body of the worm on the victim's machine and then gives control to the worm.

- Code substitution

Here, the instructions in the worm body are substituted with semantically equivalent instructions. Some examples are mentioned below:

1. Multiplication can be achieved by successive addition.
2. Addition can be achieved using xor operator.
3. Register renaming: if you want to transfer a value from register B to A, first move the value to any unused register and then move it to A [4].

### 3.2 Parts of a polymorphic worm

- Body/Code of the worm

This is the part of the worm which is malicious and does the actual damage.

- Polymorphic Engine (PE)

The polymorphic engine is responsible for changing the representation of the worm either by encryption, code substitution or both.

- Polymorphic Decryptor (PD)

The polymorphic decryptor is responsible for decrypting the worm (if encryption technique is used for polymorphism) on the victim's machine and then give control to the worm [4].

## 4 Double-honeynet system

### 4.1 System architecture

We propose a double-honeynet system to detect new worms automatically. A key contribution of this system is the ability to distinguish worm activities from normal activities without the involvement of experts.

Figure 1 shows the main components of the double-honeybet system. firstly, the incoming traffic goes through the Gate Translator which samples the unwanted inbound connections and redirects the samples connections to Honeynet 1.

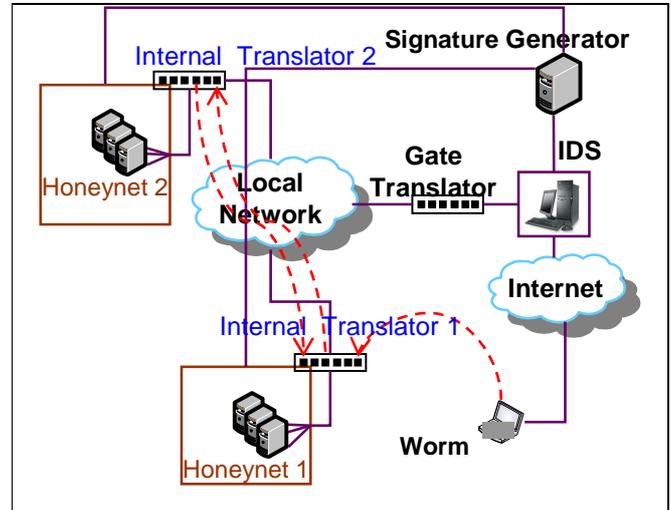


Figure 1. System architecture.

The gate translator is configured with publicly-accessible addresses, which represent wanted services. Connections made to other addresses are considered unwanted and redirected to Honeynet 1 by the Gate Translator.

Secondly, Once Honeynet 1 is compromised, the worm will attempt to make outbound connections. Each honeynet is associated with an Internal Translator implemented in router that separates the honeynet from the rest of the network. The Internal Translator 1 intercepts all outbound connections from honeynet 1 and redirects them to honeynet 2 which does the same forming a loop.

Only packets that make outbound connections are considered malicious, and hence the Double-honeynet forwards only packets that make outbound connections. This policy is due to the fact that benign users do not try to make outbound connections if they are faced with non-existing addresses.

Lastly, When enough instances of worm payloads are collected by Honeynet 1 and Honeynet 2, they are forwarded to the Signature Generator component which generates signatures automatically using specific algorithms that will be discussed in the next section. Afterwards, the Signature Generator component updates the IDS database automatically by using a module that converts the signatures into Bro or pseudo-Snort format.

For further details on the double-honeynet architecture the reader is advised to refer to our published works [13,14].

## 5 Siganture generation algorithms

In this section, we describe signature generation process steps (Substring Exaction Algorithm and Factor Analysis method). The Substrings Extraction algorithm aims to extract substrings from polymorphic worm whereas the Factor Analysis method aims to get the most significant substrings that shared among polymorphic worm instances and to use them as signatures.

### 5.1 Substrings Extraction

Let's assume we have a polymorphic worm A that has n instances ( $A_1, \dots, A_n$ ). Assume further that  $A_i$  has length  $M_i$  for  $i=1, \dots, n$ . Assume that we select  $A_1$  to be the instance from which we extract substrings. Now consider the instance  $A_1$  to be the string ( $a_1 a_2 a_3 \dots a_{m_1}$ ). Let X to be the minimum length of the substrings that we are going to extract from  $A_1$ . The first substring from  $A_1$  with length X is ( $a_1 a_2 \dots a_x$ ). Then shift one position to the right to extract a new substring, which will be ( $a_2 a_3 \dots a_{x+1}$ ). Continuing this way the last substring from  $A_1$  will be ( $a_{m_1-x+1} \dots a_{m_1}$ ). In general if instance  $A_i$  has length equal to M and minimum length equal to X, then the Total Substrings Extraction of  $A_i$  TSE ( $A_i$ ) will be obtained by this equation:

$$TSE(A_i) = M - X + 1$$

The next step is to increase X by one and start new substrings extraction from the beginning of  $A_1$ . The first substring will be ( $a_1 a_2 \dots a_{x+1}$ ). The substrings extraction will continue satisfy this condition  $X < M$ .

Figures 2 and Table 1 show all substrings extraction possibilities from the string ZYXCBA assuming the minimum length of X is equal to three (3).

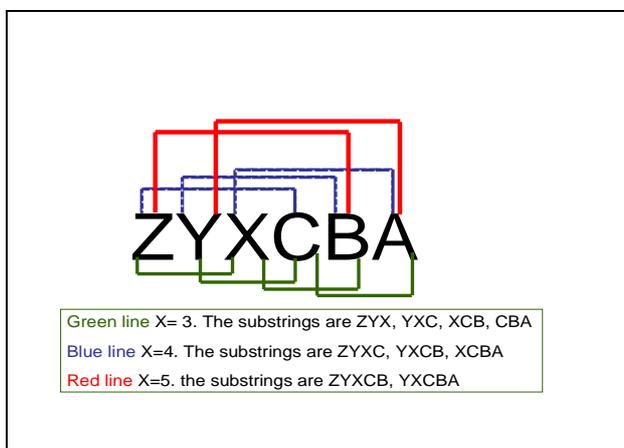


Figure 2. Substrings extraction.

TABLE 1: SUBSTRINGS EXTRACTION.

No. of Substractions	Length of X	Substrings
S1,1	3	ZYX
S1,2	3	YXC
S1,3	3	XCB
S1,4	3	CBA
S1,5	4	ZYXC
S1,6	4	YXCB
S1,7	4	XCBA
S1,8	5	ZYXCB
S1,9	5	YXCBA

### 5.2 Factor Analysis

In this subsection we give a brief introduction to factor analysis and how it is done [15]. Factor Analysis is a multivariate method used for data reduction purposes. The basic idea is to represent a set of variables by a smaller number of variables. In this case they are called factors. These factors can be thought of as underlying constructs that cannot be measured by a single variable. The objective of the use of Factor Analysis in this paper is to reduce the Polymorphic worm payloads dimensions, so the most important factors will appear and use them as signatures.

- **Assumptions**

Factor analysis is designed for interval data, although it can also be used for ordinal data. The variables used in factor analysis should be linearly related to each other. This can be checked by looking at scatterplots of pairs of variables. Obviously the variables must also be at least moderately correlated to each other, otherwise the number of factors will be almost the same as the number of original variables, which means that carrying out a factor analysis would be pointless.

- **The steps in factor analysis**

The factor analysis model can be written algebraically as follows. If you have p variables  $X_1, X_2, \dots, X_p$  measured on a sample of n subjects, then variable i can be written as a linear combination of m factors  $F_1, F_2, \dots, F_m$  where, as explained above  $m < p$ . Thus,

$$X_i = a_{i1} F_1 + a_{i2} F_2 + \dots + a_{im} F_m + e_i$$

Where the  $a_i$ s are the factor loadings (or scores) for variable i and  $e_i$  is the part of variable  $X_i$  that cannot be 'explained' by the factors.

There are three main steps in a factor analysis:

### 1. Calculate initial factor loadings

This can be done in a number of different ways; the two most common methods are described very briefly below:

- *Principal component method*

As the name suggests, this method uses the method used to carry out a principal components analysis. However, the factors obtained will not actually be the principal components (although the loadings for the  $k^{\text{th}}$  factor will be proportional to the coefficients of the  $k^{\text{th}}$  principal component).

- *Principal axis factoring*

This is a method which tries to find the lowest number of factors which can account for the variability in the original variables that is associated with these factors (this is in contrast to the principal components method which looks for a set of factors which can account for the total variability in the original variables).

### 2. Factor rotation

Once the initial factor loadings have been calculated, the factors are rotated. This is done to find factors that are easier to interpret. If there are 'clusters' (groups) of variables — i.e. subgroups of variables that are strongly inter-related — then the rotation is done to try to make variables within a subgroup score as highly (positively or negatively) as possible on one particular factor while, at the same time, ensuring that the loadings for these variables on the remaining factors are as low as possible. In other words, the object of the rotation is to try to ensure that all variables have high loadings only on one factor.

### 3. Calculation of factor scores

When calculating the final factor scores (the values of the  $m$  factors,  $F_1, F_2, \dots, F_m$ , for each observation), a decision needs to be made as to how many factors to include. This is usually done using one of the following methods:

Choose  $m$  such that the factors account for a particular percentage (e.g. 75%) of the total variability in the original variables.

Choose  $m$  to be equal to the number of eigenvalues over 1 (if using the correlation matrix). [A different criteria must be used if using the covariance matrix.].

Use the scree plot of the eigenvalues. This will indicate whether there is an obvious cut-off between large and small eigenvalues.

The final factor scores are usually calculated using a regression-based approach.

## 6 Conclusion

We have proposed automated signature generation for Zero-day polymorphic worms using double-honeynet. We have proposed new detection method "Dou-ble-honeynet" to detect new worms that have not been seen before. The system is based on Factor Analysis that determines the most significant data that are shared among all polymorphic worms instances and use them as signatures. The main objectives of this research are to reduce false alarm rates and generate high quality signatures for polymorphic worms.

## 7 References

- [1] L. Spitzner. "Honeypots: Tracking Hackers". Addison Wesley Pearson Education: Boston, 2002.
- [2] D. Gusfield. "Algorithms on Strings, Trees and Sequences". Cambridge University Press: Cambridge, 1997.
- [3] J. Levine, R. La Bella, H. Owen, D. Contis, and B. Culver. "The use of honeynets to detect exploited systems across large enterprise networks"; Proc. of 2003 IEEE Workshops on Information Assurance, New York, Jun. 2003, pp. 92- 99.
- [4] P. Fogla M. Sharif R. Perdisci O. Kolesnikov W. Lee. "Polymorphic Blending Attacks"; Proc. of the 15th conference on USENIX Security Symposium, Vancouver, B.C., Canada, 2006.
- [5] C. Kreibich and J. Crowcroft. "Honeycomb—creating intrusion detection signatures using honeypots"; Workshop on Hot Topics in Networks (Hotnets-II), Cambridge, Massachusetts, Nov. 2003.
- [6] H.-A. Kim and B. Karp. "Autograph: Toward automated, distributed worm signature detection"; Proc. of 13 USENIX Security Symposium, San Diego, CA, Aug., 2004.
- [7] S. Singh, C. Estan, G. Varghese, and S. Savage. "Automated worm fingerprinting"; Proc. Of the 6th conference on Symposium on Operating Systems Design and Implementation (OSDI), Dec. 2004.
- [8] James Newsome, Brad Karp, and Dawn Song. "Polygraph: Automatically generating signatures for polymorphic worms"; Proc. of the 2005 IEEE Symposium on Security and Privacy, pp. 226 – 241, May 2005.
- [9] V. Yegneswaran, J. Giffin, P. Barford, and S. Jha. "An architecture for generating semantics-aware signatures"; Proc. of the 14th conference on USENIX Security Symposium, 2005.

[10] Yong Tang, Shigang Chen. "An Automated Signature-Based Approach against Polymorphic Internet Worms"; IEEE Transaction on Parallel and Distributed Systems, pp. 879-892 July 2007.

[11] Zhichun Li, Manan Sanghi, Yan Chen, Ming-Yang Kao and Brian Chavez. Hamsa. "Fast Signature Generation for Zero-day Polymorphic Worms with Provable Attack Resilience"; Proc. of the IEEE Symposium on Security and Privacy, Oakland, CA, May 2006.

[12] Lorenzo Cavallaro, Andrea Lanzi, Luca Mayer, and Mattia Monga. "LISABETH: Automated Content-Based Signature Generator for Zero-day Polymorphic Worms"; Proc. of the fourth international workshop on Software engineering for secure systems, Leipzig, Germany, May 2008.

[13] Mohssen M. Z. E. Mohammed, H. Anthony Chan, Neco Ventura. "Honeycyber: Automated signature generation for zero-day polymorphic worms"; Proc. of the IEEE Military Communications Conference, MILCOM, 2008.

[14] Mohssen M. Z. E. Mohammed and H. Anthony Chan. "Fast Automated Signature Generation for Polymorphic Worms Using Double-Honeynet"; Proc. of the Third International Conference on Broadband Communications, Information Technology & Biomedical Applications, 2008 .

[15] Manly, B.F.J., 'Multivariate Statistical Methods', Third edition 2005, Chapman and Hall.

[16] Snort – The de facto Standard for Intrusion Detection/Prevention. Available: <http://www.snort.org>, 23 May 2011.

[17] Bro Intrusion Detection System. Available: <http://www.bro-ids.org/>, 23 May 2011.

# A key agreement protocol based on Identity-Based Proxy Re-encryption

Adrian Atanasiu<sup>1</sup> and Adela Mihaita<sup>1</sup>

<sup>1</sup>Department of Computer Science, Faculty of Computer Science, University of Bucharest, Bucharest, Romania

**Abstract**—*In this paper, we present a problem and propose an elegant solution for it: a protocol that allows a manager to choose his team from a database of experts and then establish with them a common secret shared key. There are some restrictions on the protocol which led us to the use of proxy re-encryption: the list of experts chosen is not known outside the team and the secret key agreed on is known only by the team. The construction and security of the protocol is based on the concept of Identity-Based Proxy Re-encryption (IB-PRE). In the second part of this paper, we extend the IB-PRE scheme by combining it with an Identity-Based Time Specific Encryption (IB-TSE) scheme obtaining a time specific encryption scheme that allows not only encryption, but also re-encryption.*

**Keywords:** key agreement, proxy re-encryption, identity-based setting, time specific encryption, knapsack problem

## 1. Introduction

Let be a manager who wants to build a team of experts from a large database and establish together a shared communication secret key. We present here a possible solution which is secure, and which has as underlying concepts the knapsack vector problem and identity-based proxy re-encryption scheme.

In the last part of the paper, in order to limit the waiting times of acceptance, we combine IB-PRE and ID-TSE in an Identity-Based Time Specific Re-encryption Scheme.

The content of the paper is the following: section 2 introduces the problem which will be solved in this paper together with a provisory protocol; in section 3 we discuss Identity-Based Proxy Re-encryption which provides the encryption part of our protocol. Section 4 presents the construction of our protocol, while section 5 and 6 discuss some general considerations and security issues. Section 7 introduces the new scheme, IB-TSRE. The paper ends with conclusions.

## 2. Formatting Instructions

## 3. A simple key agreement protocol

We consider the following problem:

There is a database of experts. In order to evaluate a project, a manager (from the database) is established. He chooses his team from the members of the database. There are some initial conditions:

- No one outside knows exactly the list of experts chosen by the manager.
- The team members must agree on a communication key which only they know.

We first propose the following simple manner of addressing the problem:

### Initial data

- the database  $\mathcal{B} = \{P_1, \dots, P_n\}$  is associated with a knapsack vector  $A = (a_1, \dots, a_n)$  and a large prime number  $p > \max_{1 \leq i \leq n} \{a_i\}$ , both of them public.
- the knapsack problem is NP-complete for anyone outside the database. The members of the database can solve it in linear time.
- Each  $P_i$  has a public key  $e_i$  for encryption, a secret key for decryption  $d_i$  and a signing algorithm  $(sig_i, ver_i)$ .

### A simple key agreement protocol

Let's suppose that the manager  $M \in \mathcal{B}$  wants to select his team  $T_M = \{P_{i_1}, \dots, P_{i_k}\}$  which will share the same secret key. We denote by  $I = \{i_1, \dots, i_k\} \subseteq [1, n]$ . The procedure is:

#### Algorithm A

- 1) M makes public

$$S = \sum_{i \in I} a_i \text{ mod } p. \quad (1)$$

- 2) Each  $P_i \in \mathcal{B}$  solves the knapsack problem  $(A, S)$  and checks if  $i \in I$ ; if so, then he generates a random number  $\alpha_i$ .
- 3) Each  $P_i$  sends to each  $P_j$  ( $j \in I, j \neq i$ ) the message

$$\{a_i, \alpha_i, sig_i(\alpha_i)\}_{e_j} \quad (2)$$

(we denoted by  $\{w\}_e$  the encryption of  $w$  under key  $e$ )

- 4) Each  $P_i \in T_M$ :
  - a) Decrypts the  $k - 1$  received messages
  - b) Checks if

$$\sum_{j \in I} a_j = S \text{ mod } p \quad (3)$$

- c) Checks if

$$ver_j(\alpha_j, sig_j(\alpha_j)) = True, \forall j \in I - \{i\} \quad (4)$$

- d) If both conditions are satisfied, then he computes the secret shared key

$$K = \sum_{j \in I} \alpha_j \pmod{p} \quad (5)$$

This protocol looks simple, but it has certain disadvantages. Step 3 is followed by all members of  $\mathcal{B}$  and it requires too many message exchanges between members of the team:  $k(k-1)$ . In order to reduce the number of sent messages, one can use a central authority, but here arises another problem since the CA doesn't have to know the team  $E_M$  nor the common key. Moreover, we would like to add some further conditions which are not met by the previous proposal:

- Only the members of the team should know that they were chosen; all the other experts should ignore this.
- One expert should have the possibility to reject the proposal of being part of an evaluation team; in this case, the manager must remove him from  $T_M$  and, eventually, replace him with another expert.

## 4. Preliminaries

For the construction of our protocol, we will make use of the concept of identity-based proxy re-encryption. Next we recall this notion.

Proxy re-encryption(PRE) allows a semi-trusted proxy to convert a ciphertext originally intended for Alice into one encrypting the same plaintext for Bob. The proxy needs for the conversion a re-encryption key issued by Alice and can not learn anything about the plaintext. An identity-based proxy re-encryption(IB-PRE) scheme [2] allows a proxy to translate an encryption under Alice's identity into one computed under Bob's identity. We will focus our attention on IB-PRE since we work in the identity-based setting.

An Identity-Based Proxy Re-Encryption scheme is an extension of Identity-Based Encryption scheme. Let's see a formal definition of IB-PRE scheme. An identity-based proxy re-encryption scheme [2] consists of the following algorithms:

- **Setup** takes as input the security parameter  $k$  and a value indicating the maximum number of consecutive re-encryptions permitted by the scheme and outputs the master public parameters which are distributed to the users and the master secret key ( $msk$ ) which is kept private.
- **KeyGen** takes as input an identity  $id$  and the master secret key and outputs a private decryption key  $sk_{id}$  corresponding to that identity.
- **Enc** on input a set of public parameters, an identity and a plaintext, outputs the encryption of  $m$  under that identity.
- **RKGen** on input a secret key  $sk_{id_1}$  and identities  $id_1, id_2$ , outputs a re-encryption key  $rk_{id_1 \leftarrow id_2}$ .

- **Reenc** on input a ciphertext  $c_{id_1}$  under identity  $id_1$ , and a re-encryption key  $rk_{id_1 \rightarrow id_2}$ , outputs a re-encrypted ciphertext  $c_{id_2}$ .
- **Dec** decrypts the ciphertext  $c_{id}$  using the secret key  $sk_{id}$  and outputs message  $m$  or failure symbol  $\perp$ .

## 5. Our construction of the key agreement protocol

The protocol suggested in the first section has many vulnerabilities, but one of them is that the manager sends to every expert a solvable instance of the knapsack problem and therefore, any expert from the database knows the team composition. But this should be revealed only to the selected members of the team. So it is important that, in the first phase, the knapsack problem remains NP-complete for all the experts. Those who were selected to be part of the team will be sent an encrypted trapdoor for solving the knapsack problem. The protocol that we propose allows an expert to refuse the request of joining the team and so he can be replaced by another expert who accepts the request, if  $k$ , the number of members of the team is fixed.

For encryption of the trapdoor, we use the encryption algorithm from the identity-based proxy re-encryption scheme. The idea of this scheme allows re-encryption of a trapdoor under a key available to the recipients.

Now that we have seen the scheme we want to use for encryption, we can go on with the main issue of this paper, the key agreement protocol.

### Algorithm B

- 1) M chooses a team  $T_m = \{P_{i_1}, \dots, P_{i_k}\}$  where  $I = \{i_1, \dots, i_k\}$  and a knapsack vector  $A = (a_1, \dots, a_n)$  which he makes public.
- 2) M computes and makes public the sum

$$S = \sum_{i \in I} a_i \pmod{p}. \quad (6)$$

This instance of the knapsack problem is NP-complete for all the experts.

- 3) M sends to each  $P_j (j \in I)$  a nonce encrypted under his identity together with the trapdoor encrypted under M's identity (of course, in this case, none of the  $P_j$  is able to decrypt, since this is what we want for the moment).
- 4) Each  $P_j$  is able to decrypt the nonce and sends it back to M only if he accepts the proposal to be part of the team; note that  $P_j$  is not able yet to decrypt the trapdoor.
  - a) If each  $P_j$  gives a positive answer, then M computes (using Reenc algorithm from the IB-PRE scheme) and makes public a vector of re-encryption keys

$$Rk = (rk_{id_{i_1}}, \dots, rk_{id_{i_k}}), \quad (7)$$

where each  $rk_{id_j}$  corresponds to  $P_j$ ,  $\forall j \in I$ . Then each  $P_j$  uses the key published for himself which allows him to re-encrypt the trapdoor received at step 3 under his own identity; once he obtains the trapdoor encrypted under his identity, he will use the private key associated to his identity and decrypt.

- b) There might be experts who don't accept to join the team (they don't send back the answer); then M chooses other experts instead, and will repeat steps 3 and 4. If the new chosen experts accept, then M publishes re-encryption keys for each member, as in the step above, which allow them to obtain the encryption of the trapdoor under their own identity. After this, simply using their private keys, they will be able to decrypt the trapdoors. We must emphasize that before publishing the re-encryption keys, M will recompute

$$S = \sum_{i \in I} a_i \text{ mod } p \quad (8)$$

as the sum of elements of the knapsack vector according to the new created team.

- 5) Each  $P_j$  is now in possession of the trapdoor, so he is able to solve the knapsack vector problem and find out who are the other members of the team; then he generates a random number  $\alpha_j$  and sends the message

$$\{a_j, \alpha_j, sig_j(\alpha_j)\}_{e_i} \quad (9)$$

to every  $P_i$ , where  $i \in I, i \neq j$ .

- 6) Every member  $P_j$  of the team follows the steps:
- Decrypts the  $k - 1$  received messages;
  - Checks if

$$\sum_{j \in I} a_j = S \text{ mod } p; \quad (10)$$

- c) Checks if

$$ver_i(\alpha_i, sig_i(\alpha_i)) = True, \forall i \in I - \{j\}; \quad (11)$$

- d) If both conditions are satisfied, then he computes the secret shared key

$$K = \sum_{j \in I} \alpha_j \text{ (mod } p) \quad (12)$$

- 7) The last step verifies if all the members of the team share the same key:

- a) Each  $P_j, j \in I$ , generates a random  $\beta_j$  and sends to M the message

$$\{\{\beta_j\}_{e_j}, a_j, sig_j(a_j)\}_K; \quad (13)$$

- b) M sends back to  $P_i$  the message

$$\{\beta_j, a_j - 1, sig_M(a_j - 1)\}_K. \quad (14)$$

## 6. Considerations about the protocol

We notice that even if we use the scheme of proxy re-encryption, there is no proxy in our protocol; the experts from the team play the role of the proxy and apply re-encryption algorithm. On the other hand, in order to reduce the number of messages sent during the protocol, M sends the nonce together with trapdoor, both encrypted, in a single step 3.

We mention that step 4.a) might be repeated more than once since the new chosen experts might as well refuse joining the team. In order to limit the waiting times of acceptance from step 4.b), we introduce in section 7 an Identity-Based Time Specific Re-Encryption scheme where precisely time is essential. Anyway, refusing experts are not a problem for the security of the protocol.

We remark that we assume from the beginning that the members of the team are honest. It is very unlikely that an expert will want to fail the protocol, but still he can easily do this, for example, by sending different random numbers  $\alpha_j$  to different members or by signing a different  $\alpha_j$  in step 5. We have also omitted the situation when two or several managers want to create their own team from the same database of experts, in the same time. We leave this for future work together with the situation where each of the two managers  $M_1$  and  $M_2$  of two simultaneous teams, from the same database, is a member in the team of the other one.

We also recall that the knapsack problem used in our protocol is NP-complete. Many of the knapsack cryptosystems have been proven to be weak against low-density attacks. In this paper, we propose for the construction of our protocol a knapsack cryptosystem based on elliptic curve discrete logarithm [3]. We note that the cryptosystem from [3] has been broken by [1] who also proposes a simple solution in order to avoid their attack: in [3] instead of defining

$$C_{m_i} = \{k\alpha, P_{m_j} + ks_i\}, \quad (15)$$

one should define

$$C_{m_i} = \{ka_{\pi(i)}\alpha, P_{m_j} + ks_i\}. \quad (16)$$

This cryptosystem enjoys high-density and, therefore, avoids low-density attacks. The trapdoor for the knapsack vector is, as in the case of Merkle-Hellman cryptosystem, a super-increasing vector (which represents the private key) which allows linear solving of the problem. We refer the reader to [3] for more details on the construction of the knapsack cryptosystem suggested.

## 7. Security of the protocol

First of all, we notice that every expert chosen by M will receive the trapdoor encrypted, before he gives an answer. But this is not a problem, even if an expert doesn't accept participation, since the trapdoor is encrypted under M's identity, so nobody else, except him, is able to decrypt.

So, if an expert refuses joining the team, he can keep the trapdoor encrypted, but he won't be able to decrypt it, even if later M publishes re-encryption keys for the members of the team. Working in the identity-based setting enables decryption only for the intended recipients.

As we indicated at the beginning, we use an identity-based proxy re-encryption scheme from [2], section 4, where Green and Ateniese present two non-interactive identity-based proxy re-encryption schemes which are secure under the Decisional Bilinear Diffie-Hellman Assumption (DBDH) in the random oracle model. The first, IBP1, is secure under chosen plaintext attack (in fact, it is IND-Pr-ID-CPA secure), while the second one, IBP2, presents stronger security under adaptive chosen ciphertext attack (IND-Pr-ID-CCA secure). Any of the two constructions presented in [2] based on bilinear pairings might be used for our protocol.

## 8. A Time-Specific Encryption scheme

In this section, we introduce an identity-based time-specific re-encryption scheme, starting from an IB-PRE scheme combined with Time Specific Encryption, a concept that we detail in the next subsection. This scheme can be used in order to limit waiting time at step 4, but we believe it might be useful also in some other applications where encryption and decryption are done in a timely manner. Briefly, the scheme allows re-encryption.

### 8.1 Identity-Based Time Specific Encryption

The cryptographic primitive Time Specific Encryption (TSE) was introduced in 2010 [4] and it's closely related to the concepts of Timed-Release Encryption (TRE) and Broadcast Encryption.

The idea behind TSE is allowing a user to specify during what time interval a ciphertext can be decrypted by the receiver. This is done in the following manner in TSE: a Time Server broadcasts a key, a Time Instant Key (TIK)  $k_t$  at the beginning of each time unit,  $t$ . The TIK is available to all users. A sender, who wants to encrypt a message  $m$  to form a ciphertext  $c$ , can specify any interval  $[t_0, t_1]$ , with  $t_0 \leq t_1$ . In *Plain* TSE, a receiver can recover the message  $m$  only if he holds a TIK  $k_t$  for some  $t \in [t_0, t_1]$ .

Plain TSE was extended to public-key and identity-based settings. We remain in the identity-based setting (ID-TSE), where decryption requires also a private key corresponding to the identity of the receiver besides the appropriate TIK.

Formally, an ID-TSE scheme consists of the following algorithms[4]:

- **TS-Setup.** This algorithm is run by the Time Server, takes as input the security parameter  $k$ ,  $T$ , the number of allowed time units and outputs the master public key TS-MPK and the master secret key TS-MSK.
- **ID-Setup.** This algorithm is run by the Trusted Authority (TA), takes as input the security parameter  $k$  and

outputs the master public key ID-MPK and the master secret key ID-MSK.

- **TIK-Ext** This algorithm is run by the TS, takes as input TS-MPK, TS-MSK,  $t$  and outputs  $k_t$  which is broadcast by TS at time  $t$ .
- **ID.Key-Ext** This algorithm is run by the TA, takes as input ID-MPK, ID-MSK, an  $id$  and outputs the private key  $sk_{id}$  corresponding to  $id$ .
- **ID.Enc** This algorithm is run by the sender, takes as input TS-MPK, ID-MPK, a message  $m$ , a time interval  $[t_0, t_1]$  and an identity  $id$  and outputs a ciphertext  $c$ .
- **ID.Dec** This algorithm is run by the receiver, takes as input TS-MPK, ID-MPK, a ciphertext  $c$ , a key  $k_t$  and a private key  $sk_{id}$  and outputs either a message  $m$  or a failure symbol  $\perp$ .

Paterson and Quaglia [4] use a binary tree for the construction of the TSE schemes. The leaves of the binary tree represent time instants. They also define two particular set of nodes. The idea is to view the nodes of the tree as identities and make use of identity-based encryption techniques to instantiate plain TSE. The number  $T$  of allowed time units will be of the form  $T = 2^d$ . The tree associated in [4] to the scheme has some properties:

- 1) The root of the tree has label  $\emptyset$ ; the other nodes are labelled with binary strings of lengths between 1 and  $d$ . Therefore, each node has associated a binary string  $t_0t_1\dots t_{l-1}$ , of length  $l \leq d$ . The leaves are labelled from  $0\dots 0$  to  $1\dots 1$  and each leaf will represent a time instant.

$$t = \sum_{i=0}^{d-1} t_i 2^{d-1-i}. \tag{17}$$

- 2) There are two particular set of nodes defined with respect to the tree:
  - $\mathcal{P}_t$  - the path to  $t$ . For a time instant

$$t = \sum_{i=0}^{d-1} t_i 2^{d-1-i}, \tag{18}$$

the following path  $\mathcal{P}_t$  corresponding to  $t$  can be constructed in the tree:

$$\emptyset, t_0, t_0t_1, \dots, t_0\dots t_{d-1} \tag{19}$$

- the set  $\mathcal{S}_{[t_0, t_1]}$  which covers the interval  $[t_0, t_1]$  - the minimal set of roots of subtrees that cover leaves representing time instants in  $[t_0, t_1]$ . The labels of the nodes in this set are computed in a particular order by running Algorithm 1 from [4].

The two sets  $\mathcal{P}_t$  and  $\mathcal{S}_{[t_0, t_1]}$  intersect in a unique node only if  $t \in [t_0, t_1]$ .

## 8.2 Identity-based Time Specific Re-encryption Scheme

We present here an identity-based time specific encryption scheme combined with identity based proxy re-encryption scheme; in fact, the aim of this time specific encryption scheme is to allow not only encryption, but also re-encryption. We start from an IB-PRE scheme  $I = (Setup, KeyExt, Enc, RKeyExt, ReEnc, Dec)$  with message space  $\{0,1\}^l$  in order to derive an ID-TSE  $X = (Plain.Setup, Plain.TIK-Ext, Plain.Enc, Plain.Dec)$  with the same message space. We call our scheme an Identity-Based Time Specific Re-Encryption scheme:

- **Setup(k,T)**. Run Setup on input  $k$  to obtain a master public key TS-MPK and the secret key TS-MSK. We define  $T = 2^d$  where  $d$  is the depth of the binary tree used in TSE, and  $T$  is the number of allowed time units.
- **ID-Setup(k,T)**. Run by the TA (trusted authority), this algorithm generates the public key ID-MPK and the secret key ID-MSK.
- **TIK-Ext(TS-MPK,TS-MSK,t)**. Construct the path  $\mathcal{P}_t$  to obtain the list of nodes  $\{0, p_1, \dots, p_d\}$  on the path to  $t$ . Run Key-Ext algorithm for all nodes  $p$  in  $\mathcal{P}_t$  to obtain a set of private keys

$$\mathcal{D}_t = \{d_p : p \in \mathcal{P}_t\}. \quad (20)$$

Return  $\mathcal{D}_t$  which represents the key  $k_t$  broadcasted at moment  $t$ .

- **RKGen** ( $\mathcal{D}_t, [t'_0, t'_1]$ ). This algorithm returns a set of re-encryption keys for messages that were initially encrypted under interval  $[t_0, t_1]$  to encrypt them under another interval  $[t'_0, t'_1]$ .  $\mathcal{D}_t$  represents the set of private keys associated to the identities from the set  $\mathcal{P}_t$ , where  $t \in [t_0, t_1]$ . For every  $d_p \in \mathcal{D}_t$ , run  $RKeyExt(TS - MPK, d_p, [t'_0, t'_1])$ , and obtain the set

$$Rk_{[t_0, t_1] \rightarrow [t'_0, t'_1]} = \{rk_{d_p} : d_p \in \mathcal{D}_t\}. \quad (21)$$

- **Encryption** ( $TS - MPK, m, [t_0, t_1]$ ). Run Algorithm 1 from [4] on input  $[t_0, t_1]$  to compute a list of nodes  $\mathcal{S}_{[t_0, t_1]}$ . For each  $s \in \mathcal{S}_{[t_0, t_1]}$  run  $Enc(TS - MPK, m, s)$  to obtain a list of ciphertexts

$$\mathcal{CT}_{[t_0, t_1]} = \{c_p : p \in \mathcal{S}_{[t_0, t_1]}\}. \quad (22)$$

Then, each ciphertext obtained is encrypted under the identity of the recipient.

- **Re-Enc** ( $TS - MPK, \mathcal{CT}_{[t_0, t_1]}, Rk_{[t_0, t_1] \rightarrow [t'_0, t'_1]}$ ). For each  $c_p \in \mathcal{CT}_{[t_0, t_1]}$  and each corresponding  $rk_{d_p} \in Rk_{[t_0, t_1] \rightarrow [t'_0, t'_1]}$ , run  $ReEnc(params, rk_{d_p}, c_p)$ , and obtain a set of ciphertexts encrypted under interval  $[t'_0, t'_1]$ .
- **Decryption** ( $TS - MPK, C, \mathcal{D}_t$ ). Here  $C = (CT, [t_0, t_1])$  represents a list of ciphertexts together with a time interval. If  $t \notin [t_0, t_1]$ , then decryption can

not be applied. Otherwise run Algorithm 1 from [4] to generate an ordered list of nodes  $\mathcal{S}_{[t_0, t_1]}$  and generate the set  $\mathcal{P}_t$ . The intersection of these sets is the unique node  $p$ . Obtain the key  $d_p$  corresponding to  $p$  from  $\mathcal{D}_t$ . Run  $Dec(TS - MPK, c_p, d_p)$ , where  $c_p \in \mathcal{CT}$  is in the same position in the list  $\mathcal{CT}$  as  $p$  is in  $\mathcal{S}_{[t_0, t_1]}$  and obtain either the message  $m$  or a failure symbol.

## 8.3 Security of the scheme

Paterson and Quaglia [4] concentrate on achieving IND-CPA security for ID-TSE and even IND-CCA security for the *Plain TSE*, but they didn't manage to achieve IND-CCA security for ID-TSE. We already discussed in section 6 the security of the IB-PRE scheme.

## 9. Conclusions

We suggested in this paper a problem for which we built a protocol based on the idea of proxy re-encryption. We first proposed a simple solution improved later by using re-encryption. The protocol's security relies on the identity-based setting in which we work and on the security of the IB-PRE scheme used.

We also built an IB-TSRE scheme in order to limit waiting time at step 4 in algorithm B, which allows time specific encryption and proxy re-encryption in the same time. We think that this scheme is valuable in certain situations since it extends the notion of time specific encryption.

## References

- [1] Jingguo Bi, Xianmeng Meng, and Lidong Han. Cryptanalysis of two knapsack public-key cryptosystems. [online]. Available: <http://eprint.iacr.org/2009/537.pdf>
- [2] M. Green, G. Ateniese. "Identity-Based Proxy Re-encryption" in *Applied Cryptography and Network Security (ACNS '07)*, 2007. [online]. Available: <http://eprint.iacr.org/2006/473.pdf>
- [3] Min-Shiang Hwang, Cheng-Chi Lee, and Shiang-Feng Tzeng. "A knapsack public-key cryptosystem based on elliptic curve discrete logarithm", *Applied Mathematics and Computation*, vol. 168, Issue 1, September 2005, pp 40-46
- [4] K.G. Paterson and E.A. Quaglia. *Time Specific Encryption*, In J. Garay and R. De Prisco (eds.), SCN 2010, Lecture Notes in Computer Science Vol. 6280, pp. 1-16, Springer, 2010.

# Double Bit Sterilization of Stego Images

Imon Mukherjee<sup>1</sup> and Goutam Paul<sup>2</sup>

<sup>1</sup> Department of Computer Science & Engineering,  
Institute of Technology & Marine Engineering,  
South 24 Parganas 743 368, India,  
Email: mukherjee.imon@gmail.com.

<sup>2</sup>Department of Computer Science & Engineering,  
Jadavpur University, Kolkata 700 032, India,  
Email: goutam.paul@ieee.org.

**Abstract**—Image sterilization is the process of removing steganographic information embedded in digital images. The only work known in this area is targeted to LSB-based steganography algorithm. In this paper, we extend the idea to sterilize two least significant bits of pixel intensities. The technique does not need to know how the information has been embedded inside the image. We performed extensive experiments over stego images created by multibit steganography algorithm and our technique succeeded in sterilizing around 77% of stego pixels on average (with a maximum of 99%).

**Keywords:** Image Sterilization, Multibit Steganography, Security, Steganalysis.

## 1. Introduction

Steganography [4] is the technique of hiding information inside a media, called *cover*, so that the information is undetected by any person other than the intended recipient. The media after embedding information is called *stego*. Steganalysis [1], on the other hand, is aimed at detecting steganography and (possibly) recovering the hidden message. A rich literature exists on various steganography and steganalysis schemes.

A recent work [6] developed an algorithm to intelligently destroy stego information inside an image without affecting the image quality by reverting as many stego pixels of an image as possible to their original cover form. This method is referred as *image sterilization* that may find important utility in defense applications. In this work, we extend the above idea to multibit sterilization. In particular, we attempt to sterilize the two least significant bits of pixel intensities, where a multibit steganography algorithm is used. We also demonstrate the effectiveness of our method by extensive experimental results.

## 2. Proposed Method

We devise a neighbourhood-based multi-bit sterilization technique in spatial domain by altering the pixel values as

minimum as possible. The method consists of two rules, namely, the Selection Rule and the Substitution Rule. We describe both of these in this section.

### 2.1 Selection Rule

This rule specifies how to select the pixels whose values would be modified by the sterilization algorithm.

#### 2.1.1 Rule:1

Selection Rule 1 is valid for 1<sup>st</sup> row of pixel values except the top-left and the top-right corner position.

	$P_{i,j}$	$P_{i,j+1}$
$P_{i+1,j-1}$	$P_{i+1,j}$	$P_{i+1,j+1}$

Select the LSBs of two adjacent pixels  $P_{i+1,j-1}$  and  $P_{i+1,j}$  and make a bit sequence  $b_x$  of length two. Similarly, select the LSBs of the other two adjacent pixels  $P_{i+1,j+1}$  and  $P_{i,j+1}$  to make another bitsequence  $b_y$  of the same length.

#### 2.1.2 Rule:2

Selection Rule 2 is applied when the target pixel is present at the last row of the stego image except the bottom-left and the bottom-right positions.

$P_{i-1,j-1}$	$P_{i-1,j}$	$P_{i-1,j+1}$
$P_{i,j-1}$	$P_{i,j}$	

Select the LSBs of two adjacent pixels  $P_{i-1,j+1}$  and  $P_{i-1,j}$  and make a bit sequence  $b_x$  of length two. Similarly, select the LSBs of the other two adjacent pixels at  $P_{i-1,j-1}$  and  $P_{i,j-1}$  to make another bit sequence  $b_y$  of the same length.

#### 2.1.3 Rule:3

Selection Rule 3 is applicable for the target pixel at the top-left corner.

$P_{i,j}$	$P_{i,j+1}$
$P_{i+1,j}$	$P_{i+1,j+1}$

Select the LSBs of two adjacent pixels  $P_{i+1,j}$  and  $P_{i+1,j+1}$  and make a bit sequence  $b_x$  of length two. Similarly

select the LSB of the right pixel  $P_{i,j+1}$  and consider 0 (since no other adjacent pixel is available) to make another bit sequence  $b_y$  of the same length.

#### 2.1.4 Rule:4

Selection Rule 4 is applied when the target pixel is at the bottom-left corner of the image.

$P_{i-1,j}$	$P_{i-1,j+1}$
$P_{i,j}$	$P_{i,j+1}$

Select LSBs of two adjacent pixels  $P_{i-1,j}$  and  $P_{i-1,j+1}$  and make a bit sequence  $b_x$  of length two. Similarly, select LSB of the right pixel  $P_{i,j+1}$  of target pixel and consider 0 (since no other adjacent pixel is available) to make another bit sequence  $b_y$  of the same length.

#### 2.1.5 Rule:5

Selection Rule 5 is used when the target pixel is at top-right corner of the image.

$P_{i,j-1}$	$P_{i,j}$
$P_{i+1,j-1}$	$P_{i+1,j}$

Select the LSBs of the two adjacent pixels  $P_{i+1,j-1}$  and  $P_{i+1,j}$  and make a bit sequence  $b_x$  of length two. Similarly, select the LSB of the left pixel  $P_{i,j-1}$  and consider 0 (since no other adjacent pixel is available) to make another bit sequence  $b_y$  of the same length.

#### 2.1.6 Rule:6

Selection Rule 6 can be used when position of target pixel is at the bottom right corner of the image.

$P_{i-1,j-1}$	$P_{i-1,j}$
$P_{i,j-1}$	$P_{i,j}$

Select the LSBs of two adjacent pixels  $P_{i-1,j-1}$  and  $P_{i-1,j}$  and make a bit sequence  $b_x$  of length two. Similarly, select the LSB of the left pixel  $P_{i,j-1}$  and consider 0 (since no other adjacent pixel is available) to make another bit sequence  $b_y$  of the same length.

## 2.2 Substitution Rule

**Step-1:** Perform bit-wise XOR operation between the two bit sequences of length two as follows.

$$r_z \leftarrow b_x \oplus b_y.$$

**Step-2:** Get the last two bits of the target pixel intensity and refer it as  $b_z$ . Perform bit-wise AND operation of  $b_z$  with the complement  $r'_z$  of  $r_z$  to get  $r''_z$ .

$$r'_z \leftarrow \bar{r}_z, \quad r''_z \leftarrow r'_z \& b_z.$$

**Step-3:** Substitute the two LSBs of the target pixel with  $r''_z$ .

**Input:** A stego image.

**Output:** Sterilized version of the input stego image.

Read the intensity values from the stego image;

**for each pixel  $p$  do**

    Choose the appropriate rule to make the bit sequences  $b_x, b_y$ ;

    Find the last two bits of  $p$ , and denote this bit sequence by  $b_z$  ;

$b_z \leftarrow b_z \& \text{complement of } (b_x \oplus b_y)$ ;

**end**

Output the transformed image;

**Algorithm 1:** DoubleSterilize

## 3. Experimental Results and Accuracy Measurement

To estimate the accuracy of our technique, we need to take as inputs some sample stego images for which we know which pixel values are actually changed due to the double bit embedding. Let  $S$  be the number of stego pixels and  $S'$  out of those  $S$  pixels actually differ in intensity values when compared with the corresponding cover image. Now, suppose  $S''$  out of those  $S'$  pixels are recovered due to the sterilization process. By recovery, we mean at least one of the two least significant bits are recovered. We calculate the accuracy of double bit sterilization for this image as  $S''/S'$ .

We have used a database of 150 24-bit color images in BMP format and 100 gray-scale images (downloaded from internet). We have also prepared different text files containing the story of Evidence (downloaded from [2]). We have used MATLAB 7.7.0 as a software tool for implementation. We have applied the above algorithm on stego images to obtain their sterilized versions. Histogram and Pixel Value Difference (PVD) analysis are shown in tabular format in the following subsections, where it is clearly obtained that maximum PVD value change lies in between -3 to +3. We find that almost 77% to 99% of stego information has been destroyed without distorting the quality of image.

In Table 2, we show the performance of our sterilization algorithm on two multibit steganography algorithms. Algorithm A refers to ordinary sequential embedding, whereas algorithm B refers to the technique of [5].

### 3.1 Histogram Analysis:

The main purpose to analyze the histogram [3] is to detect significant changes in the frequency of occurrence of each color component in an image by comparing the sterilized image with the stego image. Fig. 4 and Fig. 5 are showing the histograms of one bmp image and its sterilized version.

Table 1: Sample text embedded in an image before and after sterilization

There was the usual morning rush, people hurrying for the office, traffic on the road. Therefore it was quite crowded and compact, excluding a police vehicle which rushed through the crowd. The atmosphere around became tense, as the vehicle was rushing with its ear splitting siren sound. The crowd on the road split immediately to provide a way through. The ones who were walking on the footpath were looking back at the departing vehicle with curious, fearful expressions on their faces. As the vehicle departed it left a notion of tension on the road for a while yet things gradually returned to normality.

Embedded Message [2] using [5] before Sterilization



Vjfrd!wcp"tjg# tptao# omrmjmd  
 # stqk!/rfosmd!hvpp{jlg# gls tjg"  
 neekbe."usbeeia"mm thf pnbg/ Vhpgg  
 emse# hv!ucs# pwiuf!brnvdfg#  
 bnd# 'oor"t,# dybotghnd# ' pmmicd  
 ugkj"od"uhkaj qypiee vjrnuek tie# crnue/#  
 Vjg 'umnssjerg# bsowmg#  
 cfb"le# wdlrf-!ap"whd"tfhjamg"v's  
 pwpiknd wjuk# hwr!gbr# rsljvkvod  
 skqen!qlumf-# Vkg cplvf#  
 ml"wje# smaf"rqniw"inedha  
 ueoz ul# rpouhdd ""ubx#  
 vipovdh. Wkd# onfs# wil!tfqd  
 # waokhlf!mm# tkg# elm  
 us'vk"udpf"lnnhkle cb'i cu thd  
 egsarvkod!udjhamg!vhvi"cwrluq  
 !gecqw d{qsgsqklnr# oo!tjghq dabfs.!  
 Ap!whd# tgkjb!g"fdqcpufe#  
 jt!ldeu!c mmtknn!mf!udmqjml"lo wke!so  
 'f gos!c vjkog!xfw# uhjnr#  
 escev'no{# qdvtvsodg"tm!omplcoit/{-

Changed Message using our proposed algorithm after Sterilization

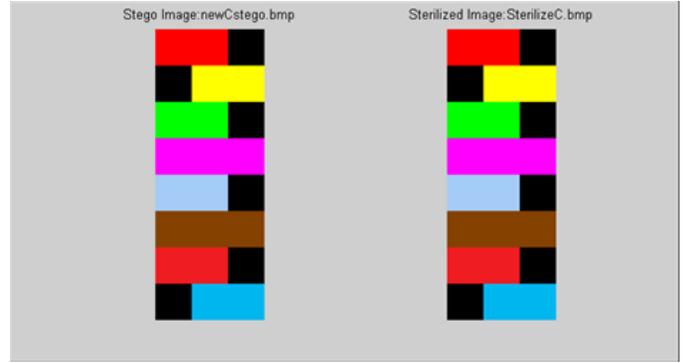


Fig. 1: Sample 24-bit color stego and sterilized images



Fig. 2: A grey scale stego image (stegoCameraman.bmp)

Visibly, the histograms of the stego image and the sterilized image are almost similar.

### 3.2 PVD Analysis

The Pixel Value Difference (PVD) is determined by :

$$PVD_{i,j} = C_{i,j} - S_{i,j},$$

where  $C_{i,j}$  is the pixel value of the red component of the cover image at the  $(i, j)$ -th position and  $S_{i,j}$  is the pixel value of the red component of the stego image at the  $(i, j)$ -th position. Table 3 and Table 4 show that the  $PVD_{i,j} \in \{-3, -2, -1, 0, 1, 2, 3\}$ , thus changes in pixel intensity values between both images cannot be visualized by human eye.

Table 2: Accuracy (minimum, maximum, average) of sterilization over 100 gray-scale and 150 color images for two different algorithms A and B.

		Grey scale	24 bit color image		
			R	G	B
Minimum %	A	65.87	66.58	68.69	67.89
	B	69.22	69.25	69.50	70.44
Average%	A	<b>77.71</b>	<b>76.48</b>	<b>78.35</b>	<b>77.89</b>
	B	<b>77.49</b>	<b>78.60</b>	<b>78.43</b>	<b>77.11</b>
Maximum %	A	86.79	85.77	85.29	87.50
	B	98.77	97.05	98.12	99.92



Fig. 3: A grey scale sterilized image (steriCameraman.bmp)

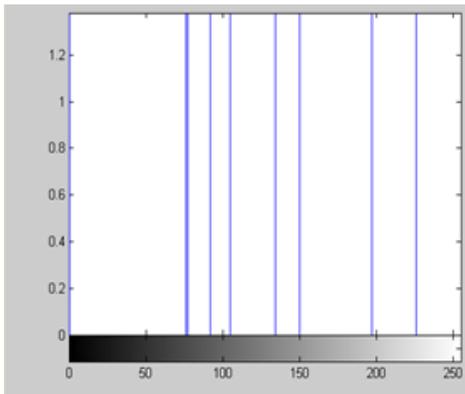


Fig. 4: Histogram of stego image (stegoC.bmp)

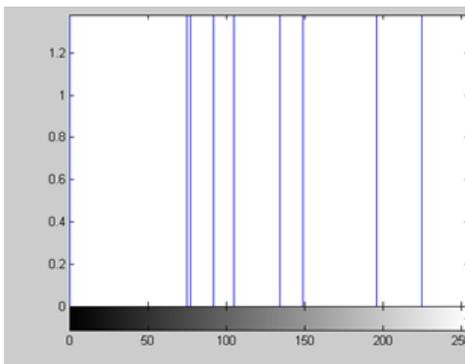


Fig. 5: Histogram of sterilized image (sterilizeC.bmp)

Table 3: Pixel value Analysis of RED intensity of newC-stego.bmp

255	255	0
0	255	255
1	1	0
254	254	254
167	167	0
129	129	129
236	236	0
0	1	1

Table 4: Pixel value Analysis of RED intensity of sterilizeC.bmp

252	252	0
0	252	252
0	0	0
252	252	252
164	164	0
128	128	128
236	236	0
0	0	0

### Acknowledgement

The authors would like to thank Mr. Anjan Payra of Department of Computer Science and Engineering, Kolyani Govt. Engg College, Nadia, India, for his help in implementation issues of this project.

### References

- [1] R. Chandramouli and K.P. Subbalakshmi. Current Trends in Steganalysis: A Critical Survey. In Control, Automation, Robotics and Vision Conference, 2004, pages 964-967.
- [2] www.englishnovels.net/2008/04/ch-2-evidence-free-novels-aghast.html
- [3] R. C. Gonzalez and R. E. Woods. Digital Image Processing. Pearson Education, 2008.
- [4] N. F. Johnson. Steganography. Technical Report, November 1995. Available at <http://www.jjtc.com/stegdoc>
- [5] Y. Park, H. Kang, S. Shin and K. Kwon. An Image Steganography Using Pixel Characteristics. In International Conference on Computational Intelligence and Security (CIS 2005), pages 581-588, vol. 3802, Lecture Notes in Computer Science, Springer.
- [6] G. Paul and I. Mukherjee. Image Sterilization to Prevent LSB-based Steganographic Transmission. In arXiv.org e-Print Archive, arXiv:1012.5573v1 [cs.MM], Dec 27, 2010.

# Smart Phones Security - Touch Screen Smudge Attack

Khalid AlRowaily, Majed AlRubaian, and Dr. Abdulrahman Mirza  
Information Systems Department, King Saud University, Riyadh, Saudi Arabia

**Abstract** - One of the famous and attractive features of a mobile device is the touch screen, especially in smart phones, where the same physical space could be used for various functions in different modes. Smudge or the remaining mark of the figure after touching the screen is very dangerous and could be used by hackers and malware applications to gain sensitive information.

In this paper we will study the problem of smudge attacks on smart-phone touch screens and their impacts on e-commerce applications. The main focus will be on how such smudge attacks could be used for fraud and identity spoofing. Finally, we will try to propose some solutions for this attack and discuss the feasibility of their implementations.

**Keywords:** Smart phone, Smudge, Authentication pattern, Oily residuals.

## 1. Problem Statement

Some models of smart phones like Android based phones provide pattern recognition as a screen unlock technique. This technique can be misused and hacked through smudge attacks as per [1].

To unlock such phone, the user needs to connect specific dots in a specific pattern defined early by him. The problem starts after unlocking the phone since the screen keeps some oily residuals that are used by the attacker to predict the authenticated pattern as shown in Figure1.

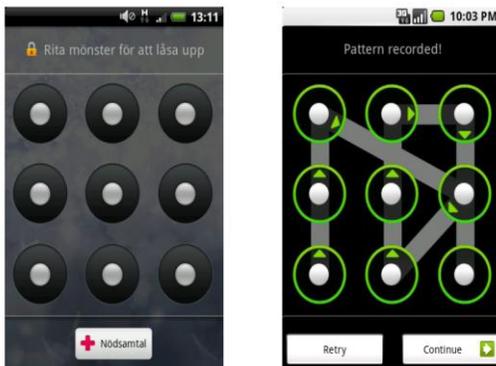


Figure 1: Unlocking Smart phone by Pattern [2]

Moreover, number of possible patterns is comparatively limited (389,112 possible patterns) since there are some constraints like each dot cannot be used more than once. From a statistical point of view, there is 92% success rate for Android smudge attacks [3].

## 2. The impacts on E-Commerce

Reading the screen smudges or oily residuals will lead to ID theft which ultimately gives the thief the ability to use the smart phone and get all the stored information. In this section, we will discuss the ID theft issue and its impact on E-Commerce.

ID theft is very dangerous since the thief can do or commit any crime on your behalf because he knows that responsibility will be totally on you not on him. This dilemma became more dangerous now after the dramatic growth in E-Commerce, Internet, and communication systems. The growth of E-Commerce and its applications increases the potential impact more and more. Most of the E-Commerce applications have been developed to work on smart phones, which makes it very easy for the customer to shop any time, and, any place just by using his phone. This is good from a service point of view but risky from security point of view. In the next paragraph, we will give a scenario explaining why it is risky to have E-Commerce applications on your smart phone.

As known, some E-Commerce applications keep sensitive information on the customer smart phone like credit card information, last bought orders, and so on. It can happen to you that your identity is stolen. For example, ID Theft can occur by a person simply stealing and unlocking the smart phone by reading the smudges or oily residuals after you had used it, and run the E-Commerce application installed on your smart phone to steal the sensitive information stored on that smart phone. Imagine that your phone is stolen or missing, the first thing for you to think about is how to return it back (the handset itself) and forget about the installed applications since your wallet, credit cards, debit cards, etc, are still with you. You will be in one of the following positions:

- 1) Lucky and wise: You find the smart phone and stop all credit cards.
- 2) Lucky but unwise: You find the smart phone and do nothing about your credit cards.

- 3) Unlucky but wise: You do not find the smart phone but you stop all credit cards.
- 4) Unlucky and unwise: You do not find the smart phone and do nothing about your credit cards.

This misuse may affect and limit the E-Commerce growth on smart phones. The users should know how to use their smart phones wisely to mitigate the risk of such occurrences.

### 3. Conclusion

In this paper, we put the spot on one of the latest attacks that is targeted, but not limited to Android based smart phones through exploiting the vulnerability of using pattern based unlocking mechanism. We have proposed the problem statement and the impact of this threat on E-Commerce growth on smart phone environments. In the full paper, we will propose some solutions to this issue along with a guidance to the smart use of the smart phones.

### 1 References

- [1] Amit Basu, Steve Myulle, Authentication in e-commerce. Commun. ACM, 2003
- [2] David Barrera, Android Security, <http://ssrg.site.uottawa.ca/slides/androidsec.pdf> accessed on May 28, 2011
- [3] <http://www.which.co.uk/news/2010/08/android-phones-vulnerable-to-oil-smudge-attacks-223309>, Accessed on May 22, 2011.

# A Novel Approach for Light Weight User Authentication Scheme in Wireless Sensor Network

Vivek Patel, Sankita Patel, Devesh Jinwala

Dept. of Computer Engineering, S. V. National Institute of Technology, Surat, Gujarat, INDIA-395007

**Abstract** - *Wireless Sensor Networks (WSNs) are operated in hostile unattended environment so authentication is one of the important security requirements. Because of the resource constrained characteristics of WSN, the authentication scheme should sustain a lesser amount of computational as well as communication overhead. Some schemes proposed in literature are vulnerable to node compromised attack. Some schemes do not provide session-key agreement. In this paper, we concentrate on improvement of authentication schemes to withstand against the node compromise attack. Additionally, our scheme also provides mutual authentication, session key agreement and protection against replay attack.*

**Keywords:** Authentication, Light weight, Wireless Sensor Network

## 1 Introduction

The Wireless Sensor Networks are composed of thousands of tiny sensor nodes and mainly used for the data-centric information-gathering applications.

WSNs are deployed in hostile unattended environment, so security is the main concern for wireless sensor networks. Designing security mechanism for WSNs is challenging because of the resource constrained nature of tiny sensor nodes [1]. Sensor nodes have limited energy and computational capabilities so security mechanisms designed for WSN must be lightweight and efficient [2].

Among the various security mechanisms, authentication is important for the wireless sensor networks. A robust authentication scheme must be designed to prevent unauthorized use of data by attacker. Researchers have proposed various authentication schemes for wireless sensor networks [3][4][5][6][7][8]. Some of the schemes are vulnerable to node compromise attack [3][5], some do not provide session key agreement[4][5][6][7][8]. Protection against the node compromise attack is strongly required because if the attacker succeeds to compromise any single coordinator node, then the entire network is compromised.

In this paper, we aim to propose authentication scheme that is resistant to node compromise attack by applying

asymmetric key encryption [9]. Additionally, our scheme also provides session key agreement, mutual authentication and protection against replay attack.

The remainder of paper is organized as follow. In Section 2 we describe related work on user authentication scheme. In section 3 we present the network and intruder model. In section 4 we describe our proposed scheme. In section 5 we discuss security analysis. In section 6 we show comparison of our scheme with existing user authentication schemes and finally we conclude and provide future work.

## 2 Related Work

In E-commerce and M-commerce many authentication schemes have been designed. WSNs have different properties and newer constraints than classical insecure network therefore several user authentication schemes which had been designed for classical insecure network could not be applicable to WSNs. The limited power energy and computation capability render classical user authentication schemes unusable for WSNs. In 2004, Benenson et. al.[10] had designed user authentication scheme for WSNs. In most of the cases, wireless sensor nodes are deployed in hostile environments, therefore WSNs are vulnerable to node compromise attack (attacker will physically attack the sensor node and retrieve all the information from the sensor node required for authentication). In classical user authentication schemes, mostly the verifier's role (the entity that verifies whether the user is valid user or not) is restricted to a single node. As the WSNs are vulnerable to node compromise attack, role of the verifier must not be restricted to a single node in a WSN. In most of the authentication schemes, the WSN divides the role of the verifier to  $t$ (threshold) sensor nodes instead of a single node. In 2005, Benenson et. al.[4] had designed a user authentication scheme for WSNs using public key cryptography. This scheme also concentrated on the node compromise attack, which means this scheme is secure from unauthorized use of the data even if any node is compromised. This scheme is secure until at least  $t$  sensor nodes are compromised (where  $t$  is threshold and  $t < n$  where  $n$  is number of nodes in user's communication range). Thus this scheme is a  $t$ -out- $n$  scheme. In this scheme process of authentication is done as shown in Figure 1.

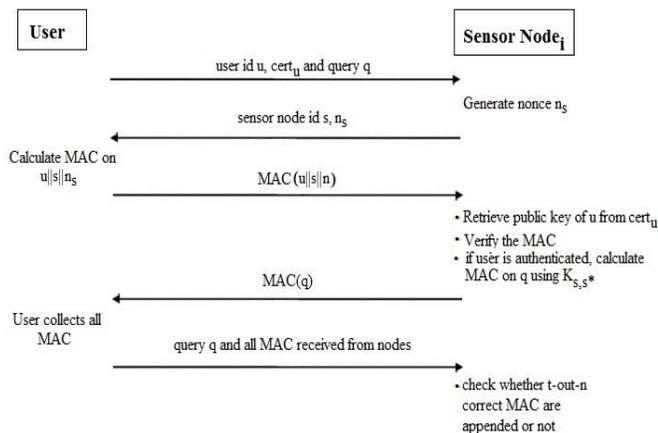


Figure 1 Robust user authentication scheme

In this scheme, it is required for the user to be verified by at least  $m$  nodes out of  $n$  nodes in its communication range, to put a query in the network. This scheme has a few drawbacks as mentioned below [11][3]:

1. In the network, only one node has the ability of querying. So the nodes in user communication range must identify this node, but this scheme does not provide any solution to identify this node. Hence, each node must have the knowledge of the entire network.
2. Each pair of nodes shares a secret key which requires large storage space but WSNs have limited storage capacity. So this scheme does not scale well.
3. This scheme does not concentrate upon the case where the node responsible for querying is compromised and provide false data.
4. In this scheme, the verification of user takes long time.

In 2006, Banerjee et. al.[5] proposed a user authentication scheme based on symmetric key cryptography. In this scheme they had used Blundo et. al[12]'s scheme for pair wise key sharing. The scheme in [5] provides that a set of nodes have the capability of querying. This scheme works as shown in Figure 2.

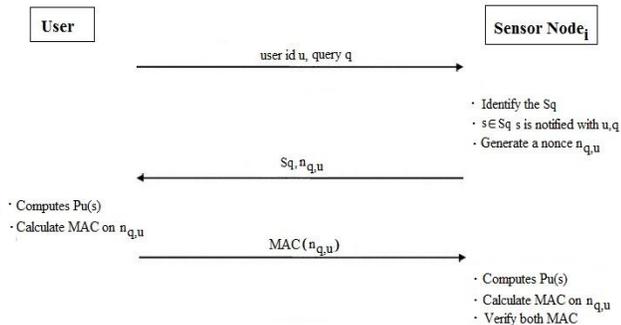


Figure 2 Symmetric key based authentication in WSN

This scheme has some drawbacks [13][14], such as it is vulnerable to node compromise attack and it does not provide any method to decide which nodes will participate in the user's query. It does not provide mutual authentication as well.

In 2007, Jiang et. al.[6] proposed a Self-certified Key Cryptography(SKC) based on the distributed user authentication scheme. They used a Key Distribution Center (KDC) for the generation of private/public keys for every sensor node and user. In this scheme, the user broadcasts a request which contains user's identity and a parameter  $R$  which is required for calculating the user's public key. After receiving a request, each node computes the session key and shares it with the user using ECC. Each node generates a nonce, encrypts it using the session key and sends it to the user. The user must be able to decrypt at least  $k$  nonce (where  $k$  is the threshold value) to gain the access of the network.

Wong et. al.[15] proposed a dynamic user authentication scheme which was vulnerable to the replay attack and the forgery attack[15][9]. In this scheme, the user cannot change his password[15]. It is also vulnerable to the threat of multiple users with a single login id[16]. Tseng et. al.[7] enhanced the dynamic user authentication scheme to withstand against replay attack and forgery attack. This scheme is shown in Figure 3.

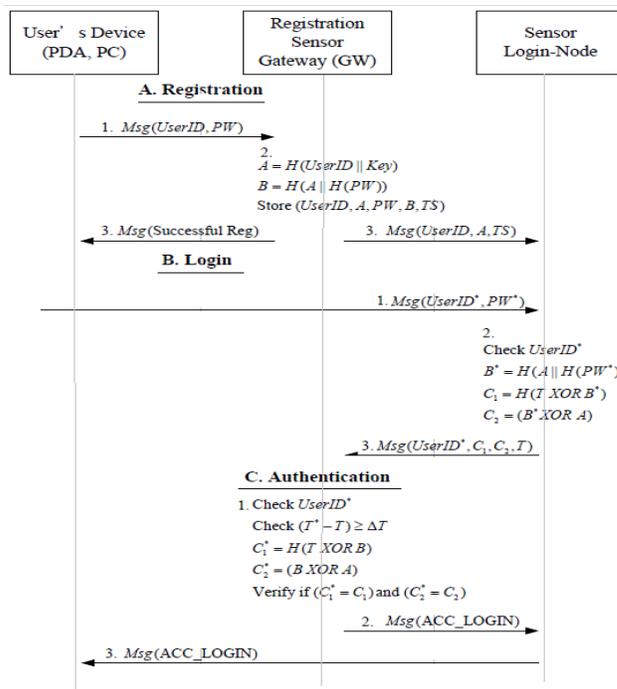


Figure 3 Improved Dynamic User Authentication schemes[7]

In this scheme, a user can login from any node and it allows the user to change his password. Here sensor nodes forward user's message to gateway to verify whether the user

is valid user or not. The gateway is also responsible for a user's registration. This scheme is vulnerable to the node compromise attack and time synchronization is required, which is a very difficult task in WSNs.

In Chai et al [8] had proposed the use of  $(t,n)$  secret sharing scheme[19] in which a secret is shared among  $n$  servers in such a way that at least  $t$  servers can combine to get the secret. So a user is required to authenticate with at least  $t$  servers. Thus, this scheme provides security against the node compromise attack.

In 2010, Omar et. al[3] proposed a light-weight user authentication scheme which provides mutual authentication and session key agreement. It provides confidentiality and data integrity, but it is vulnerable to the node compromise attack. In our work, we propose to improve this light-weight user authentication scheme in order to withstand the node compromise attack, using the asymmetric key cryptography.

### 3 Network and Intruder Model

In the given section we explained the network and intruder model for wireless sensor network.

#### 3.1 Network model

We consider WSN as star topology and entire network is managed by special node called coordinator node. Coordinator node is a communication link between the user and remaining WSN means user can communicate to WSN through coordinator node. User sends command to coordinator node, coordinator node passes command to sensor node. According to command sensor nodes perform the operation and sends data to coordinator node, coordinator node passes data to the user. Therefore coordinator node is also responsible for user authentication process.

Example of this kind of WSN is health-care system in which sensor nodes are deployed on patient's body and they send physiological parameters like ECG, heart rate etc to monitor which is a coordinator node of health-care system.

In short our system has two types of node: coordinator node and end devices. End devices do not having the ability of querying as well as they communicate with user. End devices collect the data and pass it to the coordinator node as well as end devices run the commands sent by the coordinator node. While coordinator node is relay between user and end devices, Coordinator node has ability to send commands to end devices to respond user's query. Coordinator node is also responsible for user's authentication. In our scheme we assume that user has a mobile device(such as PDA, mobile phone etc.) to wirelessly communicate with coordinator node.

In some cases user can be present near the WSN (i.e. health-care system). In this case user can directly communicate with coordinator node (figure 4).



Figure 4 Direct access to WSN [3]

While in some cases user can not be present near the WSN (i.e. nuclear power plant system). In this case user can remotely access the WSN (figure 5). In this case user will communicate through an infrastructure network.

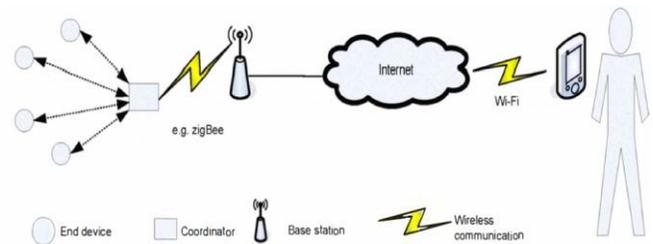


Figure 5 Remote access to WSN [3]

In general we can say that our network is modeled as follow (figure 6):

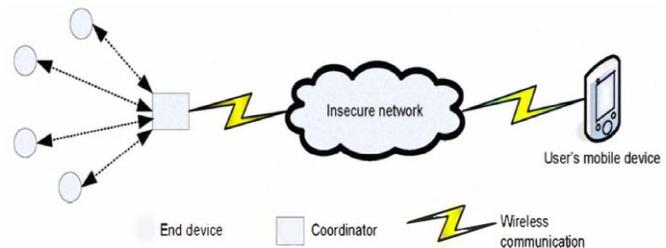


Figure 6 Network Model [3]

In both the case communication is made as follow: user's mobile device sends request to establish connection. After receiving request, coordinator node starts authentication process. Mutual authentication is done by exchanging set of messages between user's mobile device and coordinator node. If user is authenticated successfully then session-key is established and communication starts securely.

#### 3.2 Intruder model

Here user's mobile devices and coordinator node communicate over insecure network. In this scenario attacker can do different types of attacks like reply attack, eavesdrop message, sends false data to user, send false command to coordinator node etc. It is also possible that user steal user's

mobile device and retrieve the information required for authentication. Here we assume that only user knows his password. Any third party doesn't know user's password. In this scheme we wish to provide authentication, message integrity, and confidentiality. Authentication is the process to authenticate users so each party trusts each other. Confidentiality guarantees that unauthorized user can not expose the data. Due to message integrity both parties ensure that data does not modified in between. In this paper we try to improve an authentication scheme which provides mutual authentication and session key agreement, from node compromise attack. Session key can be useful to encrypt data and for message integrity (by calculating MAC using keyed hash function).

### 4 The Proposed Authentication Scheme

In this section we describe the proposed user authentication scheme which contains two phase: 1. Registration phase and 2. Login and Authentication phase. Registration phase is same as the scheme describe in [3]. In registration phase, we register user in the system. In login and authentication phase, user sends a request for authentication, process of mutual authentication is done. For user's connivance, we list the notation in table 1 which will be used throughout in our scheme.

Table 1 Notation

Symbol	Meaning
$x$	The secret key of the system
$\parallel$	Concatenation
$\oplus$	Exclusive-OR(XOR) operation
$N_u$	Nonce value of the user
$N_s$	Nonce value of the coordinator
$H()$	A one way hash function
$Enc(N,s)$	Encryption of the value N using the secret key s
$Dec(M,s)$	Decryption of the Message using the secret key s
$Enc(ID,PU_{ci})$	Encryption of the ID using the public key of Coordinator node i
$Dec(ID,PU_{ci})$	Decryption of the ID using the private key of Coordinator node i

#### 4.1 Security initialization

In our scheme there is an administrator who is responsible for loading necessary secrete keys in the WSN. He is also responsible for user's registration.

Administrator chooses a secret key  $x$ , loads secret key  $x$  into system server and coordinator nodes. Secret key  $x$  is use by System server for user's registration and by coordinator node for user's authentication.

#### 4.2 Registration phase

When a new user wishes to register he interacts with the system server. The roll of administrator is to allow legal user to register. Therefore user's password does not expose to administrator. User registration phase requires the following steps as shown figure 7.

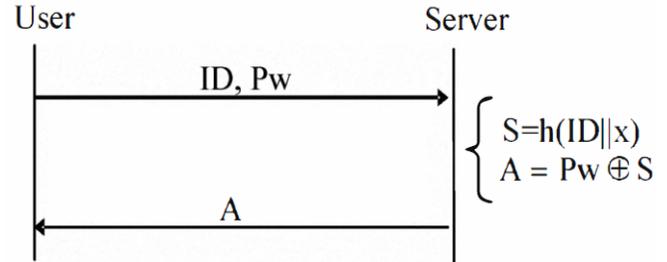


Figure 7 Registration Phase[3]

1. The user selects a password and input his identity (ID) and password(pw) to server.
2. At servers side concatenation of ID and secret key  $x$  is done to compute  $S$  by applying one way hash function[18][19] on concatenation of ID and  $x$ . After calculating  $S$ , ex-ORing this  $S$  with user's pwd to calculate  $A$ . server gives this  $A$  to user. User registers  $A$  in his mobile device.

#### 4.3 Login phase

When user wants to communicate with WSN, He must initialize authentication process as shown in figure 8.

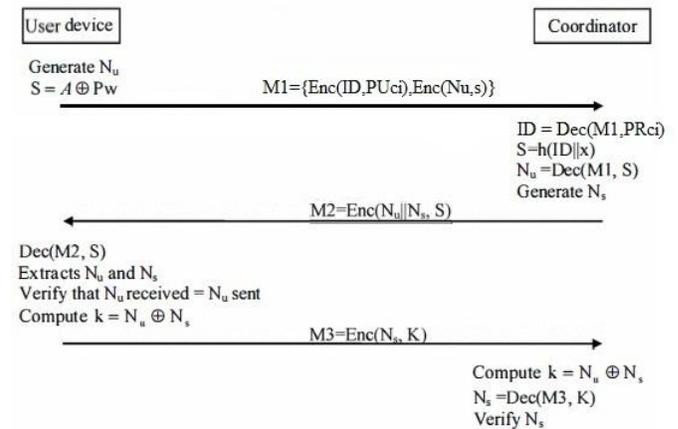


Figure 8 Authentication Process

1. User device derives  $S$  by performing ex-or of  $A$  and  $Pw$ . And it generates a nonce  $N_u$  than it encrypts the  $N_u$  with Encryption algorithm like AES and uses  $S$  as key. User will transmit Encrypted nonce and its ID to coordinator.
2. To retrieve ID coordinator will decrypt  $M1$  using its private key  $PR_{ci}$ . After retrieving ID of user coordinator will calculate  $S$  by applying hash function on concatenation of ID and secret key  $x$ . using this  $S$  it will decrypt the nonce  $N_u$ . After retrieving nonce  $N_u$  coordinator will generate a new nonce  $N_s$ . And then

send  $M2 = \text{Enc}(N_u || N_s, s)$ . Where  $\text{Enc}$  is a symmetric encryption function, such as AES (Advanced Encryption Standard), and  $\text{Dec}$  is its associated decryption function. Note that the AES algorithm [20] is used in low-rate and low-power networks [21], [22], [23].

3. User will decrypt  $M2$  to extract  $N_u$  and  $N_s$ . Then it will compare whether  $N_u$  received is equal to  $N_u$  sent or not. If both are same then user will calculate  $k$  as ex-or of  $N_u$  and  $N_s$  and send  $M3 = \text{Enc}(N_s, k)$  to coordinator.
4. Coordinator will compute  $k$  as ex-or of  $N_u$  and  $N_s$  and decrypt the  $M3$  to extract  $N_s$ , then verify that whether  $N_s$ , sent is equal to  $N_s$  receive or not. If both  $N_s$  are same then user is trusted and allowed to communicate with the network. For further communication  $k$  is used as master key.

## 5 Theoretical Analysis

The security of the proposed scheme relies on the security of the secret key  $x$  and private key of coordinators. That is, the secret key  $x$  and private key of coordinators must be kept secret and not revealed to a third party, even legal users. In addition, the secret key  $x$  must be chosen appropriately to avoid guessing attack and any coordinator should not have secret key of other coordinator.

our proposed user authentication scheme is robust against the following attacks:

- *Impersonation attack*: an attacker cannot impersonate a legal user and he will be blocked in message  $M3$  because attacker cannot decrypt the message  $M2$  as attacker does not know value of  $S$ , therefore attacker cannot retrieve nonce  $N_s$ .
- *Replay attack*: Replay attack is not possible in this scheme because if an adversary will intercept any message from legal user and replay it to user or coordinator then also attacker cannot succeed in impersonating user or coordinator as he is not able to extract the new value of  $N_u$  and  $N_s$ .
- *User's mobile device stolen attack*: when attacker compromise user's mobile device user will extract the value of  $A$  from it but attacker will not have the password so he can't calculate  $S$  so he is not able to compute a correct message.
- *Guessing attack*: Guessing attack is not possible in our scheme because plaintext value of nonce  $N_u$  and  $N_s$  does not transmit. Only encrypted value of  $N_u$  and  $N_s$  transmits so guessing attack is not possible in our scheme.
- *Coordinator compromise attack*: if any one coordinator node will compromise then also entire network will not be hacked. Because if attacker compromise a coordinator node then he will have secret key  $x$  and the private key of that coordinator node, but he will not have a private key of another coordinator to extract the ID from  $M1$  so

attacker is not able to compute  $S$  at coordinator side. Hence attacker cannot gain the access of the entire network.

In addition, to the robustness against the above attacks, the proposed scheme presents the following advantages:

- *Mutual authentication*: In this scheme both user and coordinator are authenticated, therefore attacker cannot impersonate as coordinator and provide false data to user. Therefore, users are sure about the authenticity of the received data.
- *Session key agreement*: in our scheme, at the end of a successful authentication the user and the coordinator establish a secret key  $k$ . This key can be used as a session key in order to secure the communication between the two entities (the user and the coordinator).
- *Synchronization independence*: To prevent replay attack some scheme adds timestamp to every sent message and based on this timestamp freshness of message is checked and replay attack can be prevented using this timestamp. However, one of the disadvantages of using a time-stamp is that it requires synchronization between entities. In this scheme we are using concept of nonce therefore synchronization between two entities is not required.

## 6 Comparison to other schemes

We have made comparison of over improved scheme with existing user authentication schemes described in literature survey. Comparison is shown in table 2

## 7 Conclusion

In this paper, we improve the light weight user authentication scheme, which provides mutual authentication and session key agreement. The security of proposed scheme is based on password memorized by user and secret key  $x$  as its predecessor. Thus this scheme does not require any infrastructure and it also light weight consumes less computation and communication overhead. This scheme also establishes a session key. Light weight user authentication scheme is vulnerable to node compromise attack, thus in our scheme we have improve the light weight user authentication scheme to withstand against node compromise attack. In our scheme we have applied asymmetric key cryptography. To reduce the problem of key management one can use ID based encryption[24]. In registration phase we have assumed that communication is done over secure channel. One can modify the registration phase such that registration phase is secure even if communication channel is insecure. All this work will be considered as future for improvement of light weight user authentication scheme.

Table 2 comparison

	<i>Benenson et. al[4]</i>	<i>Banerjee et. al[5]</i>	<i>Jiang et. al[6]</i>	<i>Tseng et. al[7]</i>	<i>Chai et. al[8]</i>	<i>Omar et. al[3]</i>	<i>Our Scheme</i>
Authentication	Unilateral	Unilateral	Unilateral	Unilateral	Mutual	Mutual	Mutual
Session-key agreement	No	No	No	No	No	Yes	Yes
Data integrity	Not Maintained	Not Maintained	Not Maintained	Not Maintained	Not Maintained	Maintained	Maintained
Confidentiality	Not Maintained	Not Maintained	Not Maintained	Not Maintained	Not Maintained	Maintained	Maintained
Cryptographic Techniques	PKI with ECC	Symmetric Cryptography based on Blundo et al's technique	Based on the Self-Certified Keys Cryptosystem	Hash and Xor	Threshold Cryptography (Shamir)	Symmetric key Encryption and XOR	Symmetric as well as Asymmetric Key Encryption and XOR
Infrastructure	PKI The CA could be the BS	No	KDC for providing private/public key	No	No	No	PKI The CA could be the administrator
Scalability	Yes	No (Due to Blundo)	Yes	Yes	Yes	Yes	Yes
Target of the query	Single Sensor	Set of Sensors	Set of Sensors	Set of Sensors	Set of Sensors	Coordinator	Coordinator
Vulnerability	Possibility of DOS attack by broadcasting several bogus certificate	Computation and communication overhead	Computation and communication overhead	Require Synchronization between nodes	Require Synchronization between nodes	Node Compromise attack	-
Robustness	Avoids node capture attack	Avoids node capture attack	Avoids node capture attack	Efficiency	Avoids node capture attack	Efficiency	Avoids node capture attack

## References

- [1]. N. Li, N. Zhang, S. K. Das, and B. Thuraisingham, "Privacy preservation in wireless sensor networks: A state-of-the-art survey," *Ad Hoc Networks*, vol. 7, no. 8, pp. 1501 - 1514, 2009.
- [2]. M. Sharifi, S. S. Kashi, and S. P. Ardakani, "Lap: A lightweight authentication protocol for smart dust wireless sensor networks," *International Symposium on Collaborative Technologies and Systems*, pp. 258-265, 2009.
- [3]. Omar Cheikhrouhou, Anis Koubaa, Manel Boujelben, Mohamed Abid "A Lightweight User Authentication Scheme for Wireless Sensor Networks" in *Computer Systems and Applications (AICCSA)*, pp 1-7,2010.
- [4]. Z. Benenson, N. Gedicke, and O. Raivio, "Realizing robust user authentication in sensor networks," *Workshop on Real- World Wireless Sensor Networks, REALWSN 2005*, 2005.
- [5]. S. Banerjee and D. Mukhopadhyay, "Symmetric key based authentication querying in wireless sensor networks," in *Proceedings of the First International Conference on Integrated Internet Ad Hoc and Sensor Networks*, 2006.

- [6]. C. Jiang, B. Li, and H. Xu, "An efficient scheme for user authentication in wireless sensor networks," 21st International Conference on Advanced Information Networking and Applications Workshops, AINAW-07, 2007.
- [7]. H. R. Tseng, R. H. Jan, and W. Yang, "An improved dynamic user authentication scheme for wireless sensor networks", in proceedings of IEEE globe comp, pp. 986-990, 2007.
- [8]. Z. Chai and A. R. L. Zhenfu Cao, "Threshold password authentication against guessing attacks in ad hoc networks," *Ad Hoc Networks*, vol. 5, 2007.
- [9]. R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120-126, 1978.
- [10]. Z. Benenson, F. Gartner, and D. Kesdogan, "User authentication in sensor networks (extended abstract)," *Workshop on Sensor Networks, Informatik 2004*, 2004.
- [11]. H. Wang, B. Sheng, and Q. Li, "Elliptic curve cryptography based access control in sensor networks," *Int. Journal of Security and Networks*, 1(2), 2006.
- [12]. C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Advances in Cryptology CRYPTO 92*, pp. 471-486, 1993.
- [13]. S. J. Yoon, H. Lee, S. B. Ji, K. Kim, "A user authentication scheme with privacy protection for wireless sensor networks," In *Proceedings of the 2nd Joint Workshop on Information Security*, pp. 233-244, 2007.
- [14]. H. R. Tseng, R. H. Jan, and W. Yang, "A robust user authentication scheme with self-certificates for wireless sensor networks," *Security and Communication Networks*, 2011.
- [15]. K. H. M. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, SUTC '06*, 2006.
- [16]. T. H. Chen, W. K. Shih, "A Robust Mutual Authentication Protocol for Wireless Sensor Networks," *ETRI J.* 2010, pp. 704-712, 2010.
- [17]. A. Shamir, "How to share a secret," *Communications of the ACM* 22 (11) pp. 612-613, 1979.
- [18]. C. Cid, "Recent developments in cryptographic hash functions: Security implications and future directions," *Information Security Technical Report*, vol. 11, no. 2, pp. 100 - 107, 2006.
- [19]. Y. Zheng, T. Matsumoto, and H. Imai, "Structural properties of one-way hash functions," *CRYPTO '90: Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, Springer-Verlag, pp. 285-302, 1991.
- [20]. "National institute of standards and technology (nist), advanced encryption standard (aes)." *Federal Information Processing Standards Publications (FIPS PUBS) 197*, 2001.
- [21]. S. Didla, A. Ault, and S. Bagchi, "Optimizing AES for embedded devices and wireless sensor networks," *Trident Com '08: Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities*, pp. 1-10, 2010.
- [22]. A. J. Elbirt, "Accelerated aes implementations via generalized instruction set extensions," *J. Com put. Secure*, vol. 16, no. 3, pp. 265-288, 2008.
- [23]. L. Huai, X. Zou, Z. Liu, and Y. Han, "An energy-efficient AES-CCM implementation for IEEE 802.15.4 wireless sensor networks," *NSWCTC '09: Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, IEEE Computer Society, pp. 394-397, 2009.
- [24]. A. Shamir, "Identity-based cryptosystems and signature schemes, in *Advances in Cryptology*" *Crypto '84*, *Lecture Notes in Computer Science*, Vol. 196, Springer-Verlag, pp. 47-53, 1984.

# Security of Handheld devices- Short Overview

Suhair H. Amer, Ph.D.

Department of Computer Science, Southeast Missouri State University, Cape Girardeau, MO,  
USA

**Abstract**—*The use of handheld devices is increasing steadily; however they are imposing great security risks. This paper briefly explores some of the handheld devices' vulnerabilities and some current handheld security solutions.*

**Keywords:** handheld devices, security, mobile security, PDA vulnerabilities.

## 1 Introduction

The use of handheld devices has been increasing steadily in recent years and their use is becoming more useful with free software, shareware and commercial software. At the same time they are imposing great security risks with their use. Their security can be controlled with enforcing security policies, authentication, or image based authentication, anti-virus software, and using data encryption, passwords and appropriate device configuration. Due to the increase in demand, securing handheld devices should continue to consider better performance while adhering to the strict power consumption limitations.

Free software, shareware and commercial software for handheld devices are making the use of such devices easier, more convenient and useful. They have megabytes of memory, processor speeds, wireless networking and Bluetooth technology and generous amounts of removable storage. Unfortunately many organizations have not yet established policies on exactly how to appropriately support and secure these small but powerful devices [1].

## 2 Handheld Devices Vulnerabilities

With the trend toward a highly mobile workforce, the use of handheld devices is increasing rapidly. Handheld devices are manufactured using a broad range of hardware and software. They have small physical size, limited storage and processing power, restricted user interface and strict power consumption limitations. They can, also, communicate wirelessly to other devices using infrared or radio signals. They are capable of sending and receiving electronic mail and accessing the Internet. In addition, they are extremely useful in managing appointments and contact information, reviewing documents, corresponding via electronic mail, delivering presentations, and accessing

corporate data. Moreover, because of their relatively low cost, they are often purchased by the employees themselves as an efficiency aid [2].

Handheld devices are used in a variety of environments as a lightweight, portable, multi-purpose technology tool and they are posing greater threat because they can be easily plugged-in to computers and laptops behind perimeter security defenses and without the knowledge of information technology department [1].

The most common and number one security risk associated with handheld devices is the physical loss of the device itself. This means, if device is not, at least, password enabled, losing business names, addresses and email addresses [1].

The second biggest security threat is virus infection and transmission. This is because some handheld devices are running Microsoft's handheld operating system and use applications that are vulnerable to viruses. In addition, they can be subject o Denial of Service attacks[1].

Another common use of handheld devices is downloading and reading email. Many users, unknowingly, carry unencrypted copies of their entire email inbox with them at all times. This information, if stolen, can be used to launch social engineering attacks [1].

Modern handheld computers include several wireless technologies that allow communication with other computing devices. For example, infrared communications allows handheld devices' users to send programs and data to other devices, laptop and computers. In some cases, this facilitates transferring viruses, Trojan horse software and computer worms to other devices. Many anti-virus programs do not monitor infrared communications between systems [1]. For example, a Trojan horse application can be beamed from a handheld device to another without being analyzed by the corporate firewall. Similarly, a user can browse the Internet using his/her handheld device and download an application that violates the corporate security policy [2].

In addition, Bluetooth is a very powerful wireless communication technology that allows communications

over short distances. If it is not configured correctly, it can allow any device to initiate communications with it [1].

### 3 Examples of Current-Handheld-Devices' Security Solutions

The security on modern handheld devices is currently controlled with enforcing security policies [3][4], authentication [5][6], image based authentication [7], anti-virus software [8][9][10], and using data encryption, passwords and appropriate device configuration [11]. Securing handheld devices should be balanced with the increase demand for better performance while adhering to the strict power consumption limitations.

Several issues should be considered when securing and developing an intrusion detection system for handheld devices [12] such as:

- Speed: as it should run in real time without causing performance degradation.
- Small profile size: as the security system should if and when using profiles of normal behavior capture such information and present while adhering to the storage limitations.
- Generalization and Convergence: as the method should converge quickly, requiring a minimal amount of data and resources to capture an approximation of normal program behavior and use it to detect intrusion.
- Anomaly sensitivity: as the method must be able to detect security related anomalies especially when normal behavior of device changes.

Rankin [1] suggests that in order to create a more secure handheld device environment the following should, also, be considered:

- Establish policies that address the appropriate use, support, management and security of these devices.
- Use passwords and if possible to incorporate biometrics into the password.
- Encrypt, if possible, sensitive and specific information or the entire contents of the handheld device.
- Install anti-virus software that will monitor the most common transmission technologies especially email attachments.

- Use secure transmission between the handheld device and the office or home network.

Olzak [13] recommends a layered security model to manage the risks caused by wireless handheld devices. In the model, the information moving to and from a wireless handheld device must pass through several actual and virtual tests before reaching its target. These layers comprise administrative, physical, and technical safeguards and must include all devices whether located on the company network, at home, or at a customer site. In general the layers consist of:

- Carrier Security which is concerned with the wireless carrier used for communication and whether users are keeping the handheld operating system up to date.
- Management Support is the foundation of the security program where policies are enforced.
- Security Program of an organization ensures the policies and procedures of management are carried out.
- User Awareness and training to end-users is essential to make sure that the users (employees) are aware of the risks.
- Physical Access Controls has great impact on the effectiveness of the security program.
- Logical Access Controls prevent either unauthorized users from gaining access to any information resources or authorized users from gaining access to information for which they have no permissions.
- Personal Firewall which is a set of related programs and acts. They are considered the first logical line of defense against penetration attacks.
- Antivirus Software is important to detect malicious code attacks, including spyware.
- Host-based Intrusion Protection Systems protect individual systems. Network-based Intrusion Protection Systems protect the entire network or a network segment.
- Version Management is a set of policies, processes, and tools employed to ensure that all handheld devices are at the proper operating system level.

- Device Configuration allows companies to install centrally managed device policies.

Each of the layers in Olzak model supports the layer below it and the implementation of different safeguards at each layer provides effective protection.

## 4 Conclusion and Future Work

This paper briefly discussed the security vulnerabilities of handheld devices. It also discussed examples of security techniques used to secure modern handheld devices such as enforcing security policies, authentication, image based authentication, anti-virus software, using data encryption, and appropriate device configuration. Such techniques has minimal effect on defending host-based attacks such as malicious code execution and network based attacks such as denial of service attacks. Similar to computer systems, intrusion detection systems can be developed and tested for handheld devices. Some intrusion detection systems, built for computer systems, are showing promising results with minimal overhead. In general, securing handheld devices should be balanced with the increase demand for better performance while adhering to the strict power consumption limitations.

## 5 References

- [1] David B. Rankin. "Handheld Computer Security". Information Security Management. East Carolina University. July 22, 2004. <http://www.unc.edu/~dbr/pda/pdasecurity.html>
- [2] Gavrilă, S. I., Korolev, V., and Jansen, W. A. Karygiannis, T. Assigning and Enforcing Security Policies on Handheld Devices. Canadian Information Technology Security Symposium , May 13-17, 2002 , Ottawa, Canada - May 01, 2002
- [3] Wayne Jansen, Tom Karygiannis, Serban Gravila, and Vlad Korolev. Assigning and Enforcing Security Policies on Handheld Devices. Proceedings of the Canadian Information Technology Security Symposium. May 2002.
- [4] Wayne Jansen, Tom Karygiannis, Michaela Lorga, Servan Gravila, and Vlad Korolev. Security Policy Management for Handheld Devices. The 2003 International Conference on Security and Management (SAM'03) June 2003
- [5] Wayne Jansen, Serban Gavrilă, Clement Seveillac. Smart Card Authentication for Mobile Devices. Australian Information Security Management Conference. September 2005.
- [6] Wayne Jansen, Serban Gavrilă, Vlad Korolev. Proximity-Based Authenticaion for Mobile Devices. The 2005 International Conference on Security and Management (SAM'05) June 2005.
- [7] Wayne Jansen, Serban Gavrilă, Vlad Korolev, Rick Ayers, Ryan Swanstrom. Picture Password: A Visual Login Technique for Mobile Devies, INSTIR 7030. July 2003.
- [8] <http://www.utimaco.us/products/pda/>
- [9] <http://www.bluefiresecurity.com/products.html>
- [10] <http://www.trustmarquesolutions.com/security/default.asp?content=handheld&bhcp=1>
- [11] T. Karygiannis, L. Owens. Wireless Network Security 802.11, Bluetooth and Handheld Devices. NIST Special Publication 800-48. NIST. November 2002.
- [12] A. B. Somayaji. Operating System Stability and Security Through Process Homeostasis. PhD thesis, University Of New Mexico, 2002.
- [13] Tom Olzak. "Wireless Handheld Device Security". March 2005