eJADE-S: Encrypted JADE-S for Securing Multi-Agent Applications

Basit Ali[1], Umar Manzoor[2], and Bassam Zafar[2]
[1]Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan
[2]Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
umarmanzoor@gmail.com, bzafar@kau.edu.sa

Abstract – *In the last two decades, Multi Agent System (MAS) has been used in diverse areas because of its tremendous capabilities; however, security and reliability issues are two major hurdles in adaptation of this paradigm in real distributed applications. Java Agent Development Framework (JADE) is one of the most popular frameworks used for the implementation of MAS, however, JADE has many security issues. To overcome these security issues Jade-S is proposed, however, it still has many security weaknesses. In this paper, we have proposed Encrypted Jade-S (eJade-S) for securing Multi-agent based applications. Cryptography new algorithm using two s-box for encryption / decryption is integrated in JADE-S to overcome the security issues. We have tested the proposed solution on large number of multi-agent based applications; experimental results show the effectiveness of the proposed solution in securing multi-agent based applications.*

Keywords: Securing MAS, JADE, JADE-S, Multi-Agent Systems, Encrypted JADE-S

1 Introduction

Agents are software programs that capable of performing autonomous action on the behalf of the user [6, 7]. The central concept of autonomy in agent means it does not have to receive instruction from the user and can decide the action itself according to the situation. When these agents work together to perform some common task(s) such a system is known as Multi Agents System. Agents within Multi-agent based application make can make virtual society for negotiation / obtaining their common goals [13].

In the last two decades, Multi Agent System (MAS) has been used in diverse areas such as (health sciences, bioinformatics, software modeling, distributed systems, biomedical engineering, parallel computing, autonomic computing [10, 11, 14, 15] etc) because of its tremendous capabilities; however, security and reliability issues are two major hurdles in adaptation of this paradigm in real distributed applications. In the last two decades, many frameworks for development / implementation of multi-agent system have been proposed; Java Agent Development Framework (JADE) [8] is one of the better and most often used frameworks for the implementation of MAS. Is JADE a secure framework for developing multi-agent based application? The answer is not yet however several steps have been taken for making JADE secure. The architecture layer of JADE is unsecure which implies that any application developed with JADE is also unsecure and can easily get infected by different attacks. JADE-S [1] has been design to overcome the security weakness of JADE; JADE-S is the security add-on incorporated in the JADE version 3.2 and provides authorization, authentication, message encryption, message signature and security policy features for the agents.

Although JADE-S aims to provide the security goal, however, it still has many security weaknesses [2, 3]. JADE-S saves the information related to security policy / authentication in plain text files, which can easily be modified / updated. In this paper, we have proposed Encrypted JADE-S (eJADE-S) for securing Multi-agent based applications. Cryptography new algorithm using two s-box for encryption / decryption is integrated in JADE-S to overcome the security issues.

The paper is organized as follows. The first section discusses the background related to multi-agent system, possible attack on agent based applications and possible solutions. This section is followed by the discussion of the Encrypted JADE-S (eJADE-S) which provides mechanism for securing multi-agent based applications. In Section 4 performance analyses of proposed solution on different test cases is presented. At the end conclusion is drawn and we outline some questions for future research.

2 Background

Because of tremendous capabilities of Agent technology, it has been used in diverse areas especially distributed systems. Many efficient solutions have been proposed using agent paradigm, as a result, agent development has moved from researcher communal to the practical fields. Java Agent Development Framework (JADE) is one of the most often used framework for the development / implementation of multi-agent based applications. Foundation for Intelligent

Physical Agents (FIPA) [9] is responsible for standard stipulations for multi-agent expertise and JADE supports (FIPA) stipulations.

In 2005, JADE proposed a plug-in named as JADE-S which provides the security features for the agent development framework (i.e. JADE). By extending the security model of Java, JADE-S used customizable sandbox and provided security modules which are essential for the development of secure multi-agent system.

Mobile Agents [4, 5] are usually distributed over different hosts, to provide security in such a distributed and open environment JADE-S presented the idea of multi-user system where all the activities are performed by the authenticated agents. The administrator of the system assign privilege to these agents. Moreover, each agent owned a public and private key through which it can encrypt and sign messages.

2.1 Authentication – JADE-S

The authentication process is the base of whole system and all other services that are used by this JADE-S. The authentication of the JADE-S module is supported by the security service which is composed of two elements of JAAS API (Java Authentication and Authorization Service) namely [6]:

- A Callback Handler that force the user to provide username and password.
- A Callback Handler that authorize the user.

Every user that start / create the container or agent must authenticate itself by providing username / password to the system, if the authentication is successful, the user becomes the owner of the container. This implicates that the user who starts the platform, by executing the main container also own the AMS and Directory Facilitator (DF). With the help of Callback Handler the user send their username and password to the local containers which passes the same to the main container. After receiving the information, Main container verifies the same with the login module which matches the user name / password to the corresponding user name / password stored in the password text file as shown in figure 1. If the login is successful, the authenticated agent and its container may start execution in the form of platform otherwise the system will generate error message and exits.

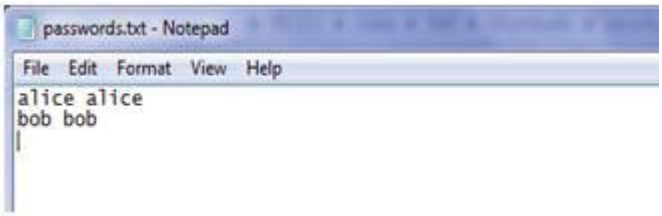

Figure 1. Sample Password.txt file

2.2 Authorization – JADE-S

The Authentication module of JADE-S assigns permissions to the agents (i.e. all possible actions and operations which the agent can perform are listed). With the help of these rules the action are permitted and denied by JADE-S. These set of rules or access control list is saved in the policy file usually named as policy.txt. This file follows the default JAAS syntax, any action that is not defined in the policy is denied forcefully by the system. The following rights can be issued to the agents:

Platform Permission: Right to create / kill Main Container of the system, Create container / agent, Kill agent / container etc. These permissions are normally assigned to the administrator of the system.

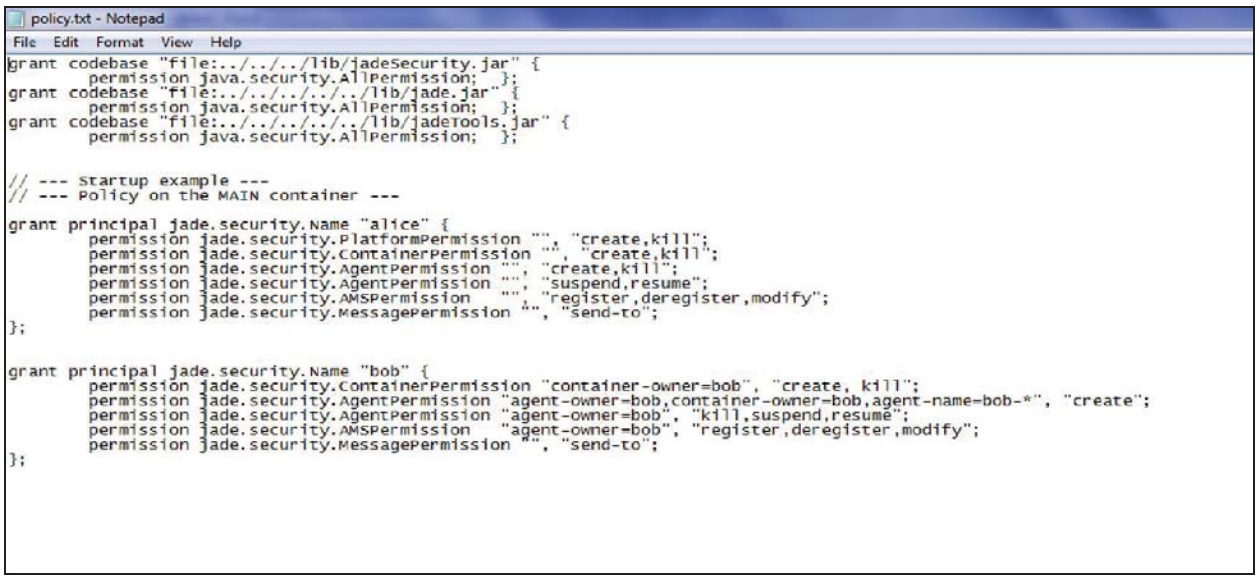

Figure 2. Sample Policy.txt file

- Container Permission: Right to create / kill a container.
- Agent Permission: Right to create / kill an agent in the local container.
- AMS Permission: Right to register and de-register agent in the AMS.
- Message Permission: Allows agent to send messages to other agents.

2.3 Encryption and Signature – JADE-S

Agents communicate with each other through messages on same or different platform, signature mechanism ensure the data reliability and non-manipulation whereas encryption assurance confidentiality. The signature and encryption mechanism of JADE-S is assured by assigning asymmetric public and private key to each agent. The signing and verifying operations are assured by the signature whereas encrypt and decrypt operation are handled by Encryption. These operations always operates on the platform in order to protect the sensitive data / information.

Agents call the appropriate security function in order to achieve the security and confidentiality of messages. Before sending message the Security Helper class function is applied on ACL Message. The sending agent has no concern with the decryption and verification of signature. These services are performed on the receiving end. In case of any failure during decryption or signature verification process, ACL Failure Messages is send to the sender.

2.4 IMTP over SSL – JADE-S

This is last and most important security feature provided by the JADE-S which includes privacy, data reliability and authenticated connection between the agents' Container Transport Protocol (IMTP) over SSL (i.e. RMI over SSL). The container-to-container mechanism of JADE-S is slightly different from the agent-to-agent where each container is awarded with a certificate. The certificate of all containers in the platform is saved in the trusted block of container. The security algorithms like mutual authentications, encryption and signature are performed by the TSL/SSL Protocol. Every container must share his owns certificate to the other communicating container and verifies the one presented to it using the trusted store. After Successfully verification, the TLS/SSL protocol continues with encryption and signature of all information exchanged between the containers.

3 Encrypted JADE-S

Jade-S creates some configuration files during installation in the root directory of the system. These files contain necessary credentials of user i.e. login username and password, privacy and policies of the system, agent roles etc are defined in the same configuration files.

Figure 3. Encrypted JADE-S Password File

These files are simple text files as shown in figure 1 and 2 respectively. These configuration files are usually placed at the project root, and any mobile agent running on the host (where these files are stored) can access / change these configurations. For example, adding more privileges to itself by making changes to the configuration files.

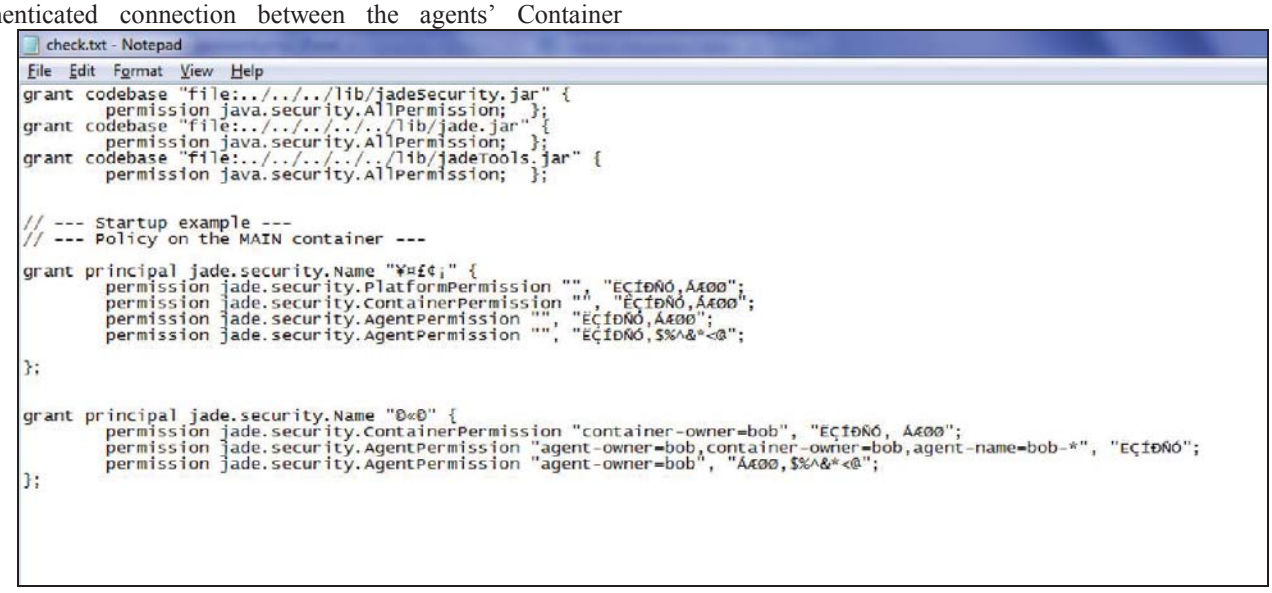

Figure 4. Encrypted JADE-S Policy File

In this paper, we have proposed Encrypted JADE-S (eJADE-S) by incorporating a new cryptographic algorithm in JADE-S to secure the configuration files (i.e. secure the login credentials, agent roles, and policy etc) generated by JADE-S. Our proposed techniques uses multi-substitution box in data encryption which makes it nearly impossible to change / modify the configuration settings. In eJade-S the password file is renamed to e-jade.txt, if any agent locates / open this file, it can't understand the contents because they are encrypted using our proposed algorithm discussed later in the section.

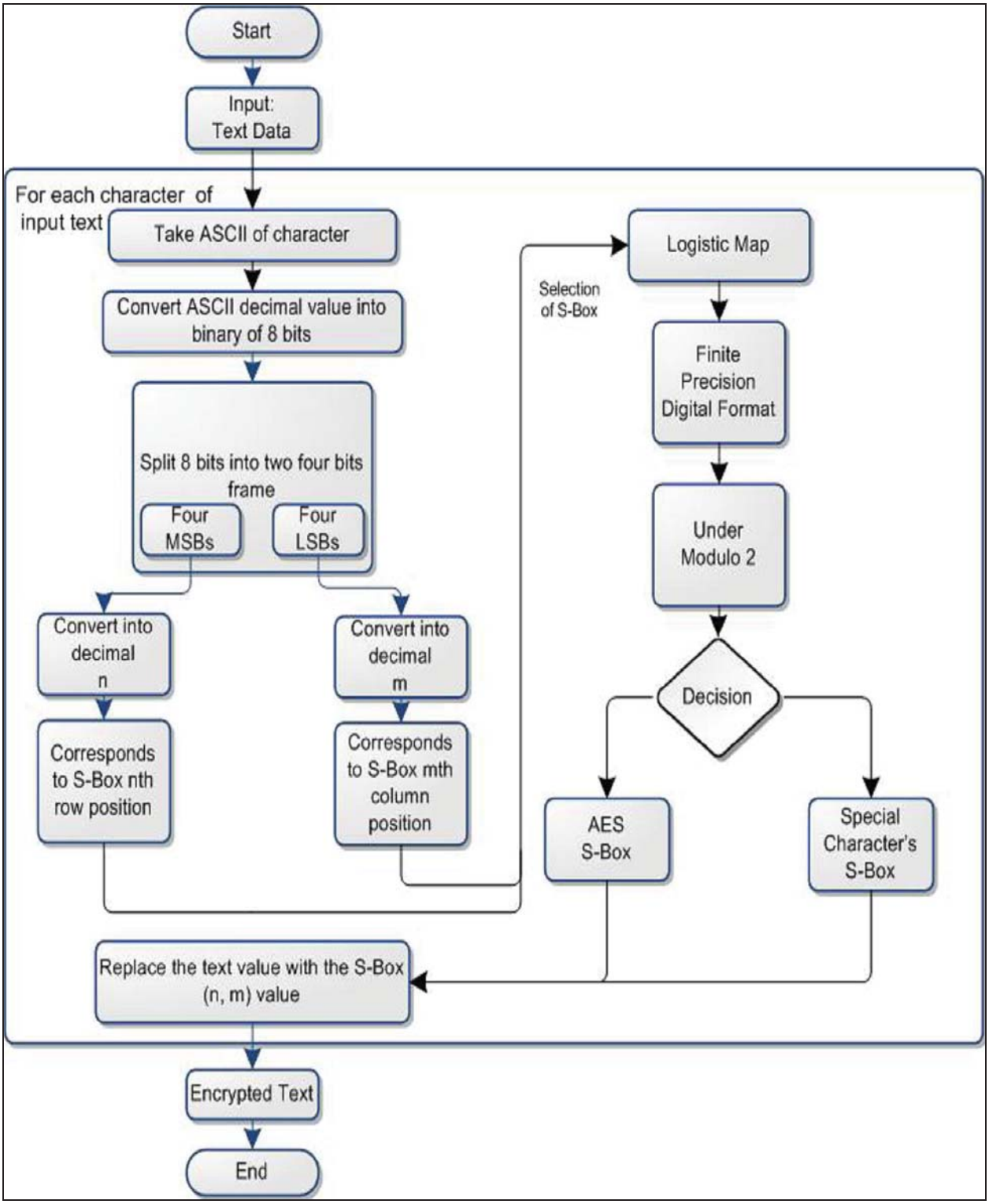

Figure 5. Proposed Substitution algorithm

After the encryption is applied on the login information, the username and password of the each agent is encrypted as shown in figure 3. Similar to password file, policy file which includes the rules / rights of the agents is also encrypted using the same algorithm; therefore, it becomes really hard for the unauthorized agent to understand which right is given to what particular agent. Once these configuration files become encrypted, entire application becomes more secure than before.

3.1 Encryption Algorithm

The encryption algorithm uses multiple S-box instead of single S-Box in substitution method.

Input: Simple text that need to encrypt.

Output: Encrypted Text.

While characterarray[]! = null do

Ascii number = characterarray[i]Asciivalue

Binarynumber = convert Ascii number to binary

Binaryarray[] = Binarynumbersplitby4digits:

Rowvalue = Binaryarray[0]

Columnvalue = Binaryarray[1]

Sum = Rowvalue + Columnvalue

Mod = Sum/2

If Mod == 0 then

Encrpytedvalue = S0[Rowvalue][Columnvalue]

characterarray[i] = Encrpytedvalue

else

Encrpytedvalue = S1[Rowvalue][Columnvalue]

characterarray[i] = Encrpytedvalue

end if

End While

3.2 Decryption Algorithm

Decryption Algorithm follows the same step in reverse order to convert the encrypted data into original text.

Input: Encrypted Text Data.

Output: Original Simple text data.

While decryptedlettersarray[]! = nulldo

Find the location of decrypted character in both S-boxes. Once the letter found save the row and column index of that character.

Row = decryptedlettersarrayrownumber:

Col = decryptedlettersarraycolnumber

Rowbinarynumber = Convert Row number to binary number

Colbinarynumber = Convert Col number to binary number

Binary = Add both Rowbinarynumber and Colbinarynumber as a string to become 8 or 16 digit numbers.

ASCII = Convert Binary to decimal number.

Orignaltext = Get the character value against the ASCII number.

End While

4 Experimental Analysis

eJade-S is design to secure the Multi Agent System, the end user can upload the multi-agent based application as input and the proposed framework yields secure MAS application as output. The GUI of application is shown in Figure 6. The end users select the MAS project that needs to be secured by clicking on the Upload Project. All the agents in the uploaded project are listed in the list box which is on the left side of the window. The user must select the Agent from the agent list and assign the roles / right by clicking on the check box which are on the right side of the window. The users must repeat the same step to assign the roles to each and every agent individually. Once the roles are assigned, the user needs to click the secure project Button which yields secure multi-agent application as output.

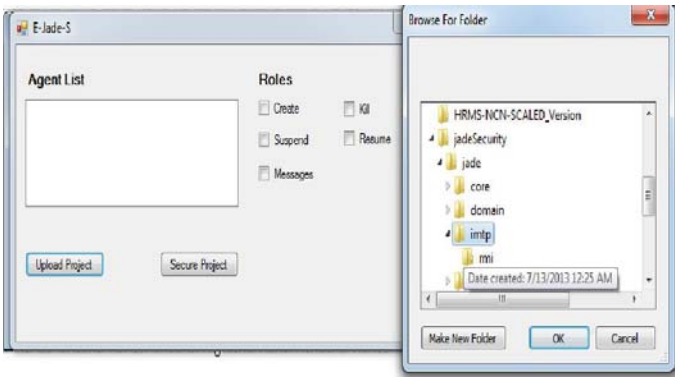

Figure 6. GUI of Encrypted JADE-S

We have evaluated the proposed solution on large number of test cases; in this section we have discussed one test case

which is multi-agent based application proposed by Manzoor et al [12] for activity or resources monitoring over the networks. An agent based system for activity monitoring on network (ABSAMN) is fully autonomous and manage the resources with the help of mobile agents. The ABSAMN uses XML configuration files for managing / monitoring the network, these configuration files are unsecure as any agent can amend or modify them. Furthermore, no authentication / authorization feature is included in the system which means any agent can access any application module and perform unauthorized operation. Seven agents are being used in ABSAMN and the administrator has to assign rights / permission to each one using eJADE-S inferface.

No	Agent Name	Rights / Permissions
1	Master Controller Agent	Create, Kill, Send Messages, Suspends, File Sharing
2	Controller Agent	Create, Send Messages, File Sharing
3	Monitor Agent	Task performing, File Sharing
4	Action Agent	Start Or Kill Process, Receive Messages
5	Statistical Agent	Task Performing
6	Information Agent	Store Information and Send Messages
7	Messaging Agent	Send Receive Messages

Table 1. Shows the rights / permissions assigned to ABSAMN agents.

Once the administrator assigns the rights to each ABSAMN agent, eJADE-S configuration files (policy and password) are generated which contains encrypted contents. After that eJADE-S library files are added to ABSAMN project and secured ABSAMN jar file is generated. After securing the ABSAMN, agents needs to authenticate / authorize with the system and can only perform the actions which are assigned to them by the administrator.

5 Conclusion

Java Agent Development Framework (JADE) is one of the better and most often used frameworks for the implementation of MAS. The architecture layer of JADE is unsecure which implies that any application developed with JADE is also unsecure and can easily get infected by different attacks. JADE-S [2] has been design to overcome the security weakness of JADE; however, it still has many security weaknesses. In this paper, we have proposed Encrypted JADE-S (eJADE-S) for securing Multi-agent based applications. Cryptography new algorithm using two S-box for encryption / decryption is integrated in JADE-S to overcome the existing security issues. The proposed framework has been tested on larger number of applications, the experimental results shows the efficiency and effectiveness of the same.

6 References

[1] Salvatore Vitabile, Vincenzo Conti, Carmelo Militello, Filippo Sorbello "An extended JADE-S based framework for developing secure Multi-Agent Systems", Computer Standards & Interfaces, Volume 31, Issue 5, September 2009, Pages 913-930.

[2] X. Vila, A. Schuster, A. Riera "Security for a Multi-Agent System based on JADE", Computers & Security, Volume 26, Issue 5, August 2007, Pages 391-400.

[3] S. Venkatesan, C. Chellappan, T. Vengattaraman, P. Dhavachelvan, Anurika Vaish "Advanced mobile agent security models for code integrity and malicious availability check" Journal of Network and Computer Applications, Volume 33, Issue 6, November 2010, Pages 661-671.

[4] T. Sander and C. F. Tschudin, “Protecting mobile agents against malicious hosts” Mobile Agents and Security, Lecture Notes in Computer Science, Vol. 1419, pp. 44-60, 1998.

[5] M. Loulou, Sfax ENIS , M. Tounsi, A.H. Kacem, M. Jmaiel, "A Formal Approach to prevent Attacks on Mobile Agent Systems" The International Conference on Emerging Security Information, Systems, and Technologies (SecureWare), Pages 42-47, 2007.

[6] Umar Manzoor, Samia Nefti, Yacine Rezgui "Categorization of malicious behaviors using ontology-based cognitive agents" Data & Knowledge Engineering, Volume 85, May 2013, Pages 40-56

[7] Umar Manzoor, Samia Nefti "iDetect: Content Based Monitoring of Complex Networks using Mobile Agents", Applied Soft Computing, Volume 12, Issue 5, May 2012, Pages 1607-1619

[8] Java Agent DEvelopment Framework (JADE), http://jade.tilab.com/

[9] Foundation for Intelligent Physical Agents (FIPA), http://www.fipa.org/

[10] K.A Karasavvas, R Baldock, A Burger "Bioinformatics integration and agent technology" Journal of Biomedical Informatics, Volume 37, Issue 3, June 2004, Pages 205-219

[11] Chuan-Jun Su, Chia-Ying Wu "JADE implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring" Applied Soft Computing, Volume 11, Issue 1, January 2011, Pages 315-325

[12] Umar Manzoor, Samia Nefti "An agent based system for activity monitoring on network – ABSAMN" Expert Systems

with Applications, Volume 36, Issue 8, October 2009, Pages 10987-10994

[13] Umar Manzoor, Samia Nefti "QUIET: A Methodology for Autonomous Software Deployment using Mobile Agents" Journal of Network and Computer Applications, Volume 33, Issue 6, November 2010, Pages 696-706

[14] David Isern, David Sánchez, Antonio Moreno "Agents applied in health care: A review" International Journal of Medical Informatics, Volume 79, Issue 3, March 2010, Pages 145-166

[15] Jiannong Cao, Yudong Sun, Xianbin Wang, Sajal K. Das "Scalable load balancing on distributed web servers using mobile agents" Journal of Parallel and Distributed Computing, Volume 63, Issue 10, October 2003, Pages 996-1005.