Survey on password recovery methods for forensic purpose

Sang Su Lee[1,2], Sung Kyong Un[1], and Soon-Ja Kim[2]
[1]Cyber Security-Convergence Research Laboratory, ETRI, Daejeon, Korea
[2]School of Electrical Engineering and Computer Science, KNU, Daegu, Korea

Abstract - *In this article, we introduce the password recovery methods against encrypted files like Microsoft Office, Adobe PDF, and other things. Of course, the password recovery may be recognized as finding cryptographic key with a cipher text through cryptanalysis. However, the forensic password recovery process would be more complicate than traditional cryptanalysis in general. We will explain the basic process of the forensic password recovery, and also introduce various methods to achieve it. In addition, commercial products including S/W and H/W for the same purpose are compared, too. Finally, we summarize the recent problems it is facing and suggest a brief description on countermeasures against the problems..*

Keywords: computer forensics, password recovery, password cracking, FPGA, GPU, CELL

1 Introduction

Digital forensics (sometimes known as digital forensic science) is a branch of forensic science encompassing the recovery and investigation of material found in digital devices, often in relation to computer crime[1-2]. According to Wikipedia(http://en.wikipedia.org/wiki/Digital_forensics), it has several sub-branches according to target devices, media or artifacts. Computer forensics for a computer system, storage media or electronic document, mobile device forensics focused on cell/SMS/Email data, network forensics concerned with the monitoring and analysis of computer network traffic, and DB forensics for DB contents and log files. Thus, digital forensics investigations requires a variety of IT technologies to deal actual cases.

In general, digital forensics is done with 3 stages: acquisition or imaging of data from digital evidences, analysis of data, and reporting of investigation [3-4]. In acquisition stage, the forensic duplicate of the digital data stored in evidence digital devices is created in forensically sound manner. The forensic investigators do analysis with the copy. The evidence is analyzed to reconstruct events or actions and to reach conclusions, work that can often be performed by less specialized staff[1]. When an investigation is complete the data is presented, usually in the form of a written report, in lay persons' terms[1]. The famous commercial forensic tools like EnCase and FTK can handle the entire stages of investigation.

Anti-forensic techniques are to prevent proper forensic investigation or make it much harder[5]. They include data destruction such as wiping data, data contraception not to create any evidences, and data hiding such as encryption and/or steganography. Actually, the experts to computers or cryptography could use the techniques years ago. Recently, various anti-forensic tools with easy user interface can be obtained from internet. That means time and efforts required for forensic investigations would be longer and more.

One of the greatest challenges faced by a forensic examiner must be an encrypted file or data. Recovering original data from encrypted one may be not an uncommon case in forensic investigations. The impact of encryption on a digital forensic investigation is largely determined by the type of data being encrypted and how. The extent of what is encrypted combined with the strength of encryption methodology will have the greatest impact on the level of difficulty imposed on the investigator[7].

In this paper, we describe the password recovery for forensic investigation. In section 2, the password verification procedure and issues related to it are described. For easy understand, we show an example with encrypted Microsoft Office Word 2007 file found in forensic investigation. In section 3, we proposed future issues for more effective password recovery and to come up with the future trend. Finally, conclusions of this article is given in section 4.

2 Password recovery for forensics

2.1 How to verify

The forensic investigators may handle various file programs. Let's imagine that a forensic examiner who found encrypted file of Microsoft Office Word 2007. The first thing forensic investigator must know is file encryption procedure target program is applying. Most commercial programs apply the same procedure for file encryption as shown in figure 1.

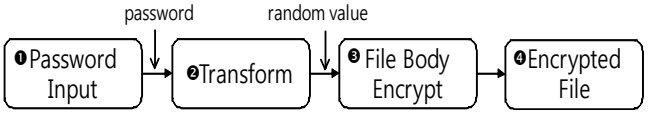

Figure 1. The general procedure of file encryption in the commercial products

The password a user selected is transformed into a string of random values, and it is used for real encryption key to the file body. The file encryption algorithm used in third step may be not useful because understanding the transform mechanism for the second step would give way to find password put into the first step. Thus, the examiner must know where the reference value is in the file, too. Each vendor applies a different transformation mechanism based on safe-proof algorithms in cryptography. While most vendors open their own transform mechanism and the position of the reference value in their public documents, some keep it in secret. If failing to find the documents, reverse engineering of the target program would be only the choice.

According to [8], Microsoft Office 2007 derives the encryption key from PKCS#5[17], which specifies the secure password generation mechanism, as described in the algorithm 1. As of now, H( ) denotes SHA-1, E( ) does AES, and '+' does concatenation.

Algorithm 1 *Password-transform*

1: *Salt* <= 16 bytes random value
2: Input password(unicode) load
3: H_0 = H(salt + password)
4: For integer *i* from 0 to 49999
$H_n = H(i + H_{n-1})$
5: $H_{50000} = H(H_{49999} + 0x00000000)$
6: *DataBlock* = H_{50000} *XOR* 64 bytes data stream consisted of 0x36
7: *EncryptionKey* = H(DataBlock)

Table 1 summarizes cryptographic algorithms used for transforming user password in famous programs. Fortunately, the programs on the table give the detailed document which gives the algorithms and transform mechanism.

Table 1. Cryptographic algorithms used for some applications' password transform

Programs	Cryptographic algorithm(s)
WinRar (v3.62)	SHA1
PDF(v7.0 - v9.0)	MD5, RC4
MS-Office 2007/2010	SHA1, AES

In Unix system, each registered user's login password is hashed and the result is stored. When a user wants to log in the system, he would be asked to put the password. The way to prove he is an authorized user is to show the hash value of the password exists in the system. In this case, the hash value of user's password is the reference value for future verifications. If readers consider the hashing process of passwords in Unix system's user authentication as transform process shown in figure 1, it can be said that forensic password verification procedure is basically the same as Unix's.

Microsoft Office Word 2007 also stores the reference value in the file as metadata[8]. For authentication of user passwords, Microsoft Office 2007 derives two values named EncryptedVerifier and EncryptedVerifierHash through the way describes in algorithm 2. The Salt value used in algorithm 1 and this two values are stored in the encrypted file's header field.

Algorithm 2 *Reference value generation for verification*

1: Verifier <= 16 bytes random value
2: *EncryptedVerifier* = E(*EncryptionKey*, Verifier)
3: *EncryptedVerifierHash* = E(H(Verifier))

The investigator who found an encrypted Microsoft Office WORD 2007 file was required to extract the three values of (Salt, EncryptedVerifier, EncryptedVerifierHash) from the file for password recovery. After obtaining the EncryptionKey according to the word through algorithm 1, he must decrypt EncryptedVerifier and EncryptedVerifierHash, respectively. Finally, if the hashed value of decrypted EncryptedVerifier is the same as the value of decrypted EncryptedVerifierHash, the word might be the correct password.

Now, imagine again the examiner knows how to mix SHA-1 and AES algorithms to transform user password in Microsoft Office Word 2007 and where to find reference value for the verification. He can try verification with any word. If he gets the same as reference value after transforming a word he chooses, the word must be a password. The problem is how fast he can process the verification. Under the blind situation about the password, his available choice would be a bruteforce verification. Figure 2 shows a flow chart for password verification of encrypted files.

2.2 How fast to do

Because most investigators are assigned to multiple cases and have a finite amount of time to allocate to each investigation, they are often unable to devote adequate time to each examination[7]. However, just one encrypted file in which the critical evidence may be stored can make all investigators involved in a case must to be stuck. The reason is the password recovery is very time consuming process in

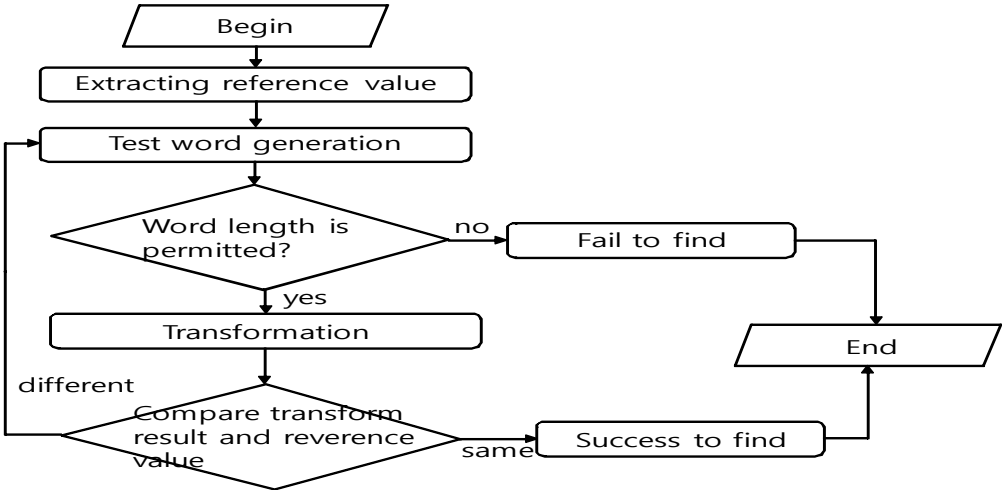

Figure 2. The flow chart for password verification of encrypted files

general. As mentioned before, vendors use safety-proof cryptoalgorithms for password transform. Since the algorithms were designed to survive the bruteforce attack, transform mechanisms also have the resistance to the same attack. Thus, the time consumed for password recovery relies on the performance of verification systems.

Many research groups have studied various methods to increase the speed of cryptographic algorithm processing for long time. In the early 1990s, the researchers paid their attentions to the software implementation techniques for the purpose. Due to the great advances in semiconductor technology, the cost for hardware solutions kept down. By the late 1990s, many researchers focused on specific hardware solution like FPGAs. It gave researchers not only high feasibility like software implementation but also high speed data processing like ASIC. Especially, they had great interested in built-in characteristics of parallel processing. By the middle of 2000s, many researchers and engineers have studied on applying multicore processors for parallel processing of cryptoalgorithms.

In Black Hat Europe 2008, CrackStation[9], an excellent accelerator for MD5 processing, was introduced. Taking advantage of the Cell's vector architecture[9-10], it showed up to 1.4 billion MD5 calculations per second. It was 90-100 times faster than performance of Intel-based architecture at that time. Nick Breese, the developer of CrackStation, implemented it on Sony's Play Station 3. An alternative to CELL for the cryptographic accelerator might be graphic processing units(GPUs). Actually, GPU and Cell are close cousins from a hardware architecture point of view. They are flexible and easy to program using high level languages and APIs. Two major differences are the number of multicore processors and major usage. While CELL has 8 internal core processors, The NVIDIA GeForce 8800 GTX GPU is comprised of 16 streaming multiprocessors (SMs). Each SM has 8 streaming processors (SPs), with each group of 8 SPs sharing 16 kB of per-block shared memory[12-13]. Thus, GPUs can be regarded as massively parallel processors with 10 times faster computation and 10 times higher memory bandwidth than CPUs[11]. And, while CELL is general purpose microprocessor, GPUs are graphic processing accelerators mainly. Recently, NVIDIA introduced CUDA[13-14] to support developers who want to use GPUs for general purposes. Recently, most commercial products for password recovery of encrypted files use GPUs for fast processing. Figure 3 shows password recovery speed for Microsoft Office 2007/2010 supported by ElcomSoft AOPR. As one can see, NVIDIA GeForce 480 gives about 10 times faster recovery speed than Intel Core i7 processor.

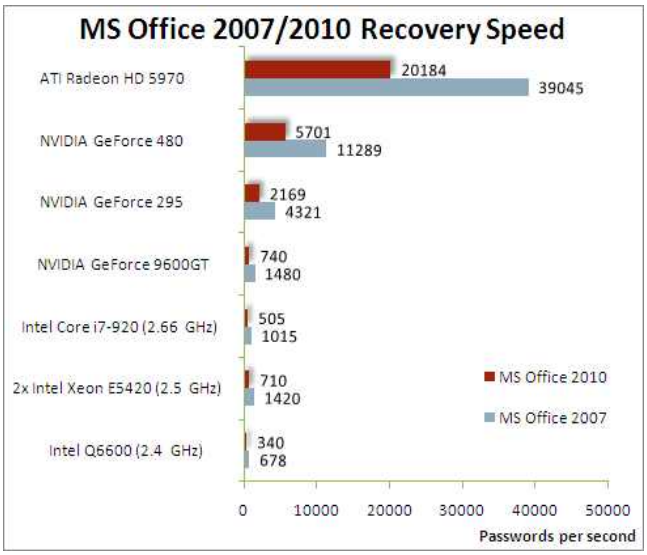

Figure 3. MS Office 2007/2010 password recovery speed by ElcomSoft AOPR[15]

2.3 How to reduce the number of words

In computer systems, the number of all printable characters in English language is 95. One can easily

understand the number would be increased in exponential manner according to increase of word length. Thus, possible 5-length combinations with all printable characters are 955 = 7,737,809,375. Let's look over the figure 3, again. In the beginning of section 2, we assume an examiner found an encrypted Microsoft Office Word 2007 file. If he uses Elcomsoft AOPR program run on a computer system with a NVIDIA GeForce 480 board, he must have more than 190 hours to test all 5-length words. An worse thing is there in no guarantee the correct password is in the words. Who knows the password is 6-, 7-, or more than 7- length word. The time to be required for testing all combination of characters is increased in exponential manner, too.

As you can see in figure 3, password recovery speed for Microsoft Office Word 2010 is almost 0.5 times slower than the speed for Microsoft Office Word 2007. Thus, one can predict Microsoft would make transform mechanism more complicate to come up with increase of password recovery speed. In this point of view, password recovery for forensic investigation must not absolutely depend on the performance of any solution.

If one can reduce the number of words to be tested, then estimated time of verification will be reduced, too. For example, people tend to use words combined with lower case English characters and digit numbers as their passwords. That means the possibility to find the password would be very high from the verification with those characters. In this case, possible 5 length combinations with those characters is drastically reduced to (26 + 10)5 = 60,466,176, and estimated time of verification for the combinations about Microsoft Office Word 2007 would be less than 1.5 hour.

Another way to reduce the words to be tested is using a dictionary. In contrast with a bruteforce attack, where a large proportion key space is searched systematically, a dictionary attack tries only those possibilities, which are most likely to succeed, typically derived from a list of words for example a dictionary or a holy bible etc. Dictionary attacks succeed because many people have a tendency to choose passwords which are short (7 characters or fewer), single words found in dictionaries or simple, easily-predicted variations on words, such as appending a digit. John the Ripper, the famous password cracking tool, supports the dictionary attack. It takes text string samples (usually from a file, called a wordlist, containing words found in a dictionary), encrypting it in the same format as the password being examined (including both the encryption algorithm and key), and comparing the output to the encrypted string[16].

There is no doubt in that the success possibility of dictionary attacks depends on the dictionary used for the attacks. Thus, a dictionary file for the attacks doesn't need to be targeted for unspecific people. For the forensic investigation, the dictionary had better contain selected words derived from information about a suspect. For example, his/her name concatenated his/her birth year can be a choice. This dictionary has much smaller size than the general dictionary file included in John the Ripper tool package.left.

3 Future Issues

While finding vulnerability in transform mechanism might be the ideal case for password verification, it is actually impossible. As described before, the encryption key in Microsoft Office Word 2007 is derived from PKCS#5 through the transform process. The current version of PKCS#5 is announced in 1999, but the weakness or vulnerability of it has not been reported, yet.

Rather, to scale up the computing resources can be practical. In the example, the estimated time of Microsoft Office Word 2007 5-length password verification with 95 characters 190 hours under one NVIDIA GeForece 480 board. If a hundred of boards are applied for verification, the estimated time would be reduced to 1.9 hour. Famous password recovery solutions of Elcomsoft, Passware, and Passcovery provide distributed password recovery in client-server model. In this case, parallel processing by GPU in PC inside and parallel processing in distributed nodes are combined. However, the cost to maintain the massive system is a problem. And new version of commercial programs can easily mix the transform mechanism up so that the verification time is increased drastically. Thus, the computer systems of each police division or law enforcement organization are required to interoperate. For this purpose, integrated framework to guarantee the reliable and secure operation of linked systems must be designed.

And, to reduce the number of words for password verification, it is required to share the information about the crimes and criminals with each law enforcement organization. The profiling pattern from one crime case can be helpful to make a profile for the other cases.

4 Conclusions

In computer forensics, One of the greatest challenges faced by a forensic examiner must be finding the key used for encrypt a file or data. To handle this effectively, a great computing power and techniques to reduce the number of words to be verified for password recovery are required.

For the former, the specific hardware system like GPU, CELL, or FPGA, which inherits the parallel processing characteristics, is used. In addition, distributed computing via system aggregation or networking gives more powerful performance. For the later, dictionary based verification is in common.

Unfortunately, the word length people use as their passwords gets longer. And, program vendors make the password transform mechanism more complicate. To come up

with the situation, a reliable and secure framework for interoperation of systems belonging to each law enforcement organization must be proposed to aggregate the computing resources.

And, information reported from investigation of crimes must be shared between law enforcements. For example, the password pattern found from investigation of crime case A might be helpful to make password dictionary for investigating a similar crime.

5 References

[1] M Reith, C Carr, G Gunsch, "An examination of digital forensic models," International Journal of Digital Evidence, 2002.

[2] Carrier, B, "Defining digital forensic examination and analysis tools," Digital Research Workshop II, 2001.

[3] Casey, Eoghan, Digital Evidence and Computer Crime, Second Edition, Elsevier, ISBN 0-12-163104-4. 2004.

[4] "Electronic Crime Scene Investigation Guide: A Guide for First Responders," National Institute of Justice, 2001.

[5] Vincent Liu and Francis Brown, "Bleeding-Edge Anti-Forensics," Infosec World Conference & Expo, April 3, 2006.

[6] Simson Garfinkel, "Anti-Forensics: Techniques, Detection and Countermeasures," 2nd International, Conference in i-Warefare and Security, pp 77, 2007.

[7] Mark Whitteker, "Anti-Forensics : Breaking the Forensic Process," ISSA Journal, pp 10, 2008..

[8] MS-OFFICE, "[MS-OFFCRYPTO]:Office Document Cryptography Structure Specification", available from http://msdn.microsoft.com/en-us/library/cc313071(v=office.12).aspx

[9] CrackStation, available from http://www.security-assessment.com/files/presentations/crackstation-njb-bheu08-v2.pdf

[10] CELL, http://en.wikipedia.org/wiki/Cell_(microprocessor)

[11] A. Ailamaki, N. K. Govindaraju, S. Harizopoulos, and D. Manocha, "Query co-processing on commodity processors," Proceedings of the 32nd international conference on Very large data bases, pages 1267–1267. VLDB Endowment, 2006.

[12] Shuai Che, Jie Li, Jeremy W. Sheaffer, Kevin Skadron, John Lach, "Accelerating Compute-Intensive Applications with GPUs and FPGAs,", 2008 Symposium on Application Specific Processors, pp.101-107, 2008.

[13] E. Lindholm, J. Nickolls, S. Oberman, and J. Montrym, "NVIDIA Tesla: A unified graphics and computing architecture," IEEE Micro, 28(2), pp39-55, 2008.

[14] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with CUDA," ACM Queue, 6(2):pp40-53, 2008.

[15] ElcomSoft, http://www.elcomsoft.com/aopr.html

[16] John the ripper, http://en.wikipedia.org/wiki/John_the_Ripper

[17]] PKCS #5: Password-Based Cryptography Specification, available from http://www.ietf.org/rfc/rfc2898.txt