Ransomware-Prevention Technique Using Key Backup

Kyungroul Lee[1], Insu Oh[2], and Kangbin Yim[2(✉)]

[1] R&BD Center for Security and Safety Industries (SSI), Soonchunhyang University, Asan, South Korea
carpedm@sch.ac.kr

[2] Department of Information Security Engineering, Soonchunhyang University, Asan, South Korea
{catalyst32,yim}@sch.ac.kr

Abstract. In this paper, a key-backup technique for the recovery of files that have been encrypted by ransomware is proposed. Ransomware interferes with the victim's system through the enactment of abnormal behavior, which is the locking of the victim's system or the encryption of the system or files. Ransomware writers require money from victims as a condition for the recovery of the encrypted files and systems that have been seized; accordingly, systems that are infected by ransomware cannot be repaired without a decryption key, making the employment of detection and recovery methods urgent. In this paper, a prevention technique for the backing up of encryption keys in a secure repository, and that can enable the recovery of ransomware-infected systems and ransomware-encrypted files. The proposed technique can be used to repair systems that have been infected by ransomware, thereby ensuring safety regarding such malicious codes.

Keywords: Ransomware · Key backup · Prevention · Big-data security

1 Introduction

As modern society is changing into an information society, a variety of the corresponding information are utilized and stored. In the past, users stored their personal information onto storage media such as hard disks, floppy disks, and CDs. While the computing environment and the network environment continually develop, remote-storage technologies such as Web hard drives and cloud services, whereby data as well as the computing environment such as software are used remotely, have emerged. With the development of the storage environment, the remote techniques are able to process mass data, and big data has appeared as a result. Big data is the derivation of the value of data through the collection, storage, management, and analysis of mass data.

Similar to other research fields, the most important element of big data is the data itself. In this respect, the analyzed results are valuable when data must be preserved without the need for an alteration; unfortunately, though, there is a problem regarding the insurance of the reliability of the data. Because many problems are caused by forgery data, a critical issue is the vulnerability of the system data that are stored, and a representative example is cyber-crime.

J.J. Jung and P. Kim (Eds.): BDTA 2016, LNICST 194, pp. 105–114, 2017.
DOI: 10.1007/978-3-319-58967-1_12

Past cyber-crimes originated from curiosity, but it has developed into a tool for revenge, monetary purposes, and cyber warfare, and cyber-crime has become a kind of service; this is called "CaaS" (Crime as a Service) [1]. These services are being sold as products, and they are classified by consulting services as either botnet setups, infection and spreading services, botnet and rentals, and crimeware-upgrade modules. Concretely, a consulting service for a typical botnet setup is approximately 350 dollars to 400 dollars, while an infection and spreading service is approximately 100 dollars for every 1,000 that are installed. Big-data systems that comprise one of the various service-related data have therefore been exposed to threats, and the salience of this issue is further highlighted by the recent emergence of ransomware. In February 2016, one of the most notorious ransomware attacks occurred; due to this, Hollywood Presbyterian Medical Center in the U.S. did not access patient data because of the ransomware-encrypted files that had infected the medical organization's electronic medical-record system [2].

In this paper, a prevention technique for user PCs as well as a variety of systems such as big-data systems is therefore proposed to provide protection from ransomware-based cyber-crime. For the proposed technique, the encryption key is monitored and backed up in a secure repository. It has been confirmed that the use of the proposed technique provides safety from the threats that are caused by ransomware for data-storage systems.

2 Related Works

Related Works describes an overview of ransomware and previous countermeasures.

2.1 Overview of Ransomware

"Ransomware" is the name of the phenomenon here, and it is a new compound word comprised of "ransom" and "malware." The origin of the name is the act of demanding money from users whose information is being held "hostage" on a computer [5]. Ransomware is a kind of Trojan horse, as it penetrates systems so that it can seize or encrypts the files and resources on the system. Money is typically required by the cyber-criminal to release the held-hostage data [3, 4].

The ultimate purpose of ransomware is a monetary one. While users typically do not want to pay malicious attackers, they are left with no choice because their systems have been seized. The payment elements are classified by the user's education, the complexity level of the malware, and the urgency of the recovery, and users pay according to the correlation between these elements [4].

Regarding the domination of a system, data is one of the important factors of ransomware, while another important factor is the payment method. In terms of the payment methods, the following classifications apply: online monetary, irregular funding, and online shopping. Online monetary is a transmission method for which users transfer money to a designated online account using an online payment service such as PayPal, E-Gold, Bitcoin, or Webmoney [4, 6]. For irregular funding, users transfer illegal funds using legitimate businesses and services [4]. The online-shopping payment

method involves an inducement by the attacker to the user for the purchase of goods from designated websites or web services. All of these payment methods are also utilized for money laundering because legitimate payments are used to provide camouflage for illicit activities [4].

Ransomware writers request either of the above-described payment methods by storing a text file such as a Readme.txt, or by displaying messages in pop-up windows. Examples of the payment-request methods are as follows [6]:

Ransomware first emerged in "PC CYBORG/AIDS information Trojan" in 1989, but this technique was not frequently used by malware writers, leading to its abandonment for almost 15 years. Nevertheless, through the widespread increase of Internet usage and e-commerce activities, ransomware was reborn in 2005 [4] (Table 1).

Table 1. Examples of payment-request methods

Ransomware name	Seizure type	Payment method
Trojan.Pluder.a	Hiding	Remit designated bank
Arhiveus	Encryption	Purchase pharmaceutical products from Russian websites
Trojan.Randsom.A	Deletion	Remit Western union
Trojan.Cryzip	Compression	Remit designated E-Gold account
Trojan.PGPCode	Encryption	Remit designated E-Gold account

The ransomware revival was mainly active in the U.S., but it gradually became widespread throughout the world. In the case of South Korea, ransomware emerged in April 2015. Notably, CryptoLocker was a new type of ransomware that was mostly active in the U.S. in 2013, and is now known as the most notorious and influential type of ransomware. According to ZDnet, ransomware writers have extracted approximately 27 million dollars in revenue from victims [8]. During the second quarter of 2013, 350,000 samples were found [9], and 14 kinds of ransomware appeared from January 2014 until September 2015 [7]. Regarding recent ransomware trends, 600,000 cases of ransomware were detected during the fourth quarter of 2015 [10].

The ransomware-infection process consists of five steps. The first step is the seeking out of the victim, for which ransomware writers propagate their ransomwares using spam mail or other propagation methods. The second step is the execution step, whereby the propagated ransomware is executed using social-engineering methods without the targeted user's knowledge; for example, CryptoLocker is disguised as a PDF icon when it is actually an executable file, so users execute ransomware by confusing the PDF file because the Windows operating system does not display file extensions by default. When ransomware is running, it generates a session key and an IV for communication with the writer. The third step is the generation of the key that encrypts the files of the system. In the case of public-key encryption, ransomware generates a secret key, so that the generated secret key is encrypted and sent to an attacker based on the received public key. Afterward, ransomware encrypts the files on the victim's system based on the generated secret key. In the case of secret-key encryption, ransomware generates a secret key for the encryption, so that the generated secret key is encrypted and sent to the attacker based on the generated session key. Afterward, the ransomware encrypts the

files on the victim's system based on the generated secret key. The fourth step is the actual encryption, for which the ransomware actually encrypts the files based on the generated encryption key. The last step is the display of the message requesting payment that is either a stored text file or is displayed on the victim's screen [9].

According to Kaspersky, when it is operated in the above way, ransomware is exceedingly dangerous because the encrypted files cannot be recovered [6]. Numerous researchers have therefore engaged in studies to counteract ransomware threats, and the surveyed results of previous countermeasures are subsequently described.

2.2 Previous Countermeasures

Traditional anti-virus products generally detect and cure malicious codes based on signatures; for this reason, it is difficult to detect newly produced malicious codes. In particular, ransomware is very difficult to repair due to the post-infection encryption. In this section, the countermeasures that have been previously researched to solve these problems are described.

File-based detection method: This method detects the specific signatures of malicious actions in the particular format that is used by an operating system; for example, a PE (Portable Executable) file. The advantage here is a fast detection time; however, false negatives can sometimes occur, and it is difficult to detect new formats or unknown malicious codes [11].

System-based-behavior detection method: This method detects the malicious behaviors on a computer system, and integrity checking and behavior blocking are also carried out. Integrity checking is the checking of a system or files periodically to verify the integrity of the files based on the hash value of the execution files and the directories of a clean system. Behavior blocking is the monitoring of the entire behavior of a system; therefore, when malicious behaviors are detected, they are blocked by an anti-virus product according to the checked result after the tracing process [7].

Resource-based-behavior detection method: This method is the monitoring of specific resources for the detection of malicious behaviors. The targets of the monitored resources include CPU usage and I/O usage, among others. To detect the malicious behaviors, an anti-virus product collects the information of these resources on a clean system over a long duration. The product then determines malicious behaviors when exceptions are detected based on the collected information [7].

Connection-based-behavior detection method: This method is the monitoring of connection statuses. When connections are required, a user accepts or rejects connections. In the case of public-key-based ransomware, the ransomware receives a secret encryption key from the attacker's server such as the C&C server; accordingly, the installed ransomware tries to connect the server during this process. If an anti-virus product blocks the connections, ransomware does not encrypt the files or the system because it does not receive the key for the encryption. This method is therefore able to preserve the safety of a user's data and system [9].

Reverse engineering: This method is the recovery of encrypted files or an encrypted system based on the discovery of the key that is stored in the ransomware using reverse engineering. The advantage of this method is the repairing capability of the encrypted

files when an anti-virus product does not detect the ransomware. Nevertheless, the drawback is the inability of the method to apply ransomware or an encrypted file that does not contain a decryption key [7].

3 Proposed Prevention Technique

As described in Sect. 2, an attacker requests a monetary sum while he/she is using a variety of ransomwares to hold a victim's system or files hostage; however, the prevention and recovery methods for the solving of this type of threat that have been introduced by security experts are limited in reality. One reason for this status is the need for a decryption key to recover a system or files, while another reason is the dependence on the payment of the ransom for the receipt of the decryption key that is stored in the attacker's system or server; for these reasons, victims suffer significant damages [6]. A countermeasure is therefore proposed in this paper, for which the key is backed up to recover the encrypted system and files.

3.1 Concept and Structure

Ransomware writers need the following requirements to run their ransomware successfully. First, a way to penetrate the victim's system successfully is needed. Users commonly install anti-virus programs to protect their systems; for this reason, writers penetrate victims' systems by using unknown vulnerabilities or known vulnerabilities such as the zero-day attack to avoid detection from anti-virus programs. Second, after penetration, ransomware should not be detected by an anti-virus program during the performance of malicious features such as the encryption and rootkit. After penetration, if ransomware does not implement the encryption, writers are unable to request money; therefore, writers want to stealthily execute features such as the encryption, and this technique is rootkit [12]. Third, writers use an implemented code that has been written by them or a library that is provided by the operating system to perform the encryption feature successfully. The encryption feature is the most important part of ransomware, because if it does not work or is not designed properly, a number of problems arise for the ransomware writers. One example here is the exposure of the encryption and decryption keys, while another example is the recovery capability when the ransomware does not implement the encryption properly. In the case of the implemented encryption feature, ransomware occasionally does not work on a specific system, so writers use cryptography libraries for reliability. Lastly, the number of keys that are generated for the encryption is equivalent to the number of the infected systems; in this case, writers generate one key equally for reliability because it is difficult for writers to manage the generated keys.

3.2 Assumption

In this paper, both the concept and the structure of the proposed countermeasure are described based on the above-defined requirements; here, an especial focus is placed on

the third and fourth requirements. In the case of the encryption feature, it is assumed that ransomware writers use cryptography libraries due to a reliability that is greater than that from the implementation of the codes by the writers themselves. For the Windows operating system, a variety of commercial cryptography libraries are provided, and the newly adopted CNG library that was adopted during the development of Windows Vista, Windows 7, and Windows 8 is especially notable [13, 14]. Accordingly, it is assumed that ransomware uses the CNG library or downloads encryption codes from outside of the server, and then the codes encrypt the victim's files. In the case of the encryption key, when ransomware generates the key inside or receives it from outside of the server, the ransomware must generate or import the key. For this feature, ransomware uses specific functions of the CNG library; therefore, the proposed concept serves as the basis for the defining of the utilized functions with respect to the encryption on the CNG library.

Generation and import functions for the key: The BCryptGenerateSymmetricKey and BCryptGenerateKeyPair functions for the generating keys, and the BCryptImportKey and BCryptImportKeyPair functions for the importing keys.

Encryption and Decryption: The BCryptEncrypt function for the encryption and the BCryptDecrypt function for the decryption.

For this paper, it is assumed that ransomware encrypts the victim's files using the CNG library; therefore, the prevention program hooks both the key-generation and the key-import functions for the backing up of the keys, and then the obtained keys are stored in a safe repository of the victim's system, or they are sent to an authentication server or a certificate server. Nevertheless, if the steps for the key generation and import

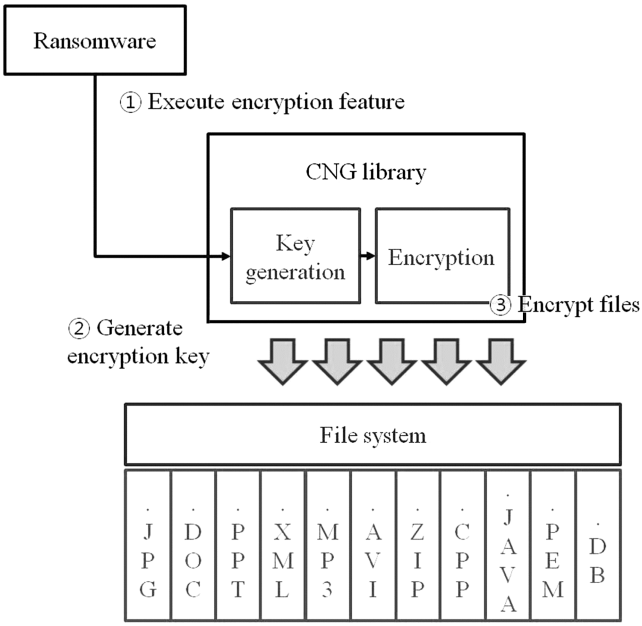

Fig. 1. The whole structure of the ransomware-encryption step for which the CNG library is used

are not executed by the ransomware, the encryption is based on the inside key that is in the ransomware file, and the prevention program backs the keys up during the encryption step. Figure 1 shows the entire structure of the ransomware-encryption step for which the CNG library is used.

Step 1. Ransomware penetrate the victim's system and executes the encryption feature to encrypt the files of the victim's system.

Step 2. Ransomware loads the CNG library to execute the encryption, followed by its generation of the encryption key. When the key is generated successfully, the generated key is sent to the attacker.

Step 3. Ransomware encrypts files such as .JPG and .DOC that are on the victim's system.

3.3 Prevention Technique

The structure of ransomware is described above, whereby it calls the BCryptGenerate-SymmetricKey function of the CNG library to generate the secret key, or it calls the BCryptGenerateKeyPair function of the CNG library to generate the public/private key-pair. The prevention program searches all of the programs that are loaded on the CNG library, and then hooks the BCryptGenerateSymmetricKey function to back up the secret key and the BCryptGenerateKeyPair function to back up the public/private key-pair; then, when the ransomware generates the key, the hooked codes are executed by the prevention program. The codes extract the encryption key and store it in the safe repository. If the ransomware does not generate the key due to its inclusion inside, it uses the key during the encryption step so that the prevention program stores it at that time. To import the key, the ransomware calls the BCryptImportKey function in the case of secret-key cryptography, or it calls the BCryptImportKeyPair function in the case of public-key cryptography; for this reason, the prevention program hooks these functions when they are called by the ransomware, and then the program stores them in the safe repository. Nevertheless, if the prevention program does not obtain the encryption key during the above steps, the program extracts the key during the encryption step. Concretely, to encrypt the files, ransomware calls the BCryptEncrypt function so that the prevention program hooks the function and stores it in the safe repository. Figure 2 shows the entire structure of the proposed prevention technique.

Step 1. Ransomware penetrates the victim's system and executes the encryption feature to encrypt the files of the victim's system; for this concept, the prevention program passes the encryption feature instead of blocking it. In terms of the prevention program, it does not determine whether the running program is a clean program such as the IE (Internet Explorer) and Outlook program or ransomware.

Step 2. When ransomware calls the key-generation and key-import functions of the CNG library, the hooking code passes the key onto the prevention program, and then the code passes the execution control onto the ransomware. The delivered key is stored in the safe repository by the prevention program that

can be one module of an anti-virus program. The extracted keys are stored in the victim's safe repository or sent to the outside of an authentication server or a certification server. The received or stored keys are stored securely using password-based authentication or the certificate-based access-control technique to prevent exposure to unauthorized persons. Additionally, to improve security, the keys can be re-encrypted based on the shared keys, which are received from reliable agencies such as certification agencies and investigative agencies. When the files of the victim's system are encrypted by ransomware, the system is therefore able to recover through a decryption for which the extracted keys are used by commissioning investigation agencies or certificate agencies.

Step 3. If the prevention program does not obtain the encryption key, the program extracts the key during the encryption step. The key-storage and key-recovery processes are equal to those of Step 2.

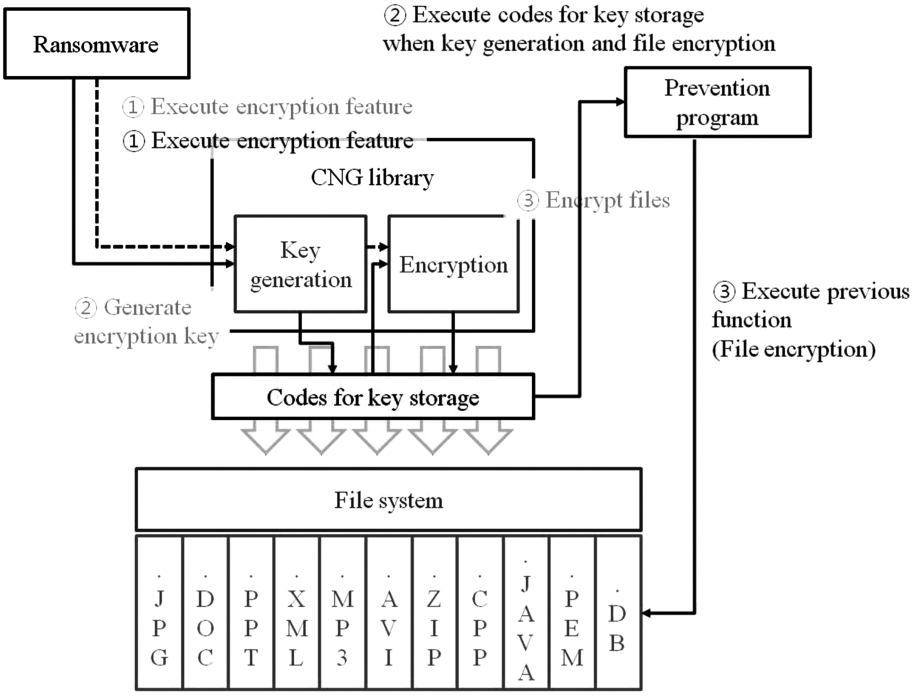

Fig. 2. The entire structure of the proposed prevention technique

4 Conclusion

In this paper, the proposed ransomware-prevention technique uses a key-backup process to recover the encrypted files of a system that has been infected by ransomware. Ransomware uses abnormal behavior, which is the locking of the victim's system or the encryption of system or files, to interfere with a victim's system; therefore, a key-backup technique for which the encryption key is stored in a safe repository is proposed in this paper. For reliability, ransomware writers use the cryptography libraries that are provided by operating systems to execute the encryption feature; these codes call functions for the generation and importation of keys, and encryption of the files. When ransomware calls these functions, the prevention programs back the keys up in the safe repository, and the program subsequently recovers the encrypted files and the system using the stored keys. This study has confirmed that the proposed countermeasure can be used for the recovery of files and systems that have been infected by ransomware.

Acknowledgment. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) that is funded by the Ministry of Education (NRF-2015R1D1A1A01057300) and the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2016-R0992-16-1006) supervised by the IITP (Institute for Information & communications Technology Promotion).

References

1. Manky, D.: Cybercrime as a Service: a very modern business. J. Comput. Fraud Secur. **2013**(6), 9–13 (2013)
2. Everett, C.: Ransomware: to pay or not to pay? J. Comput. Fraud Secur. **2016**(4), 8–12 (2016)
3. Xin, L., Qinyu, L.: Ransomware: a new cyber hijacking threat to enterprises. In: Handbook of Research on Information Security and Assurance, IGI (2009)
4. Giri, B.N., Jyoti, N.: McAfee AVERT, The Emergence of Ransomware, AVAR (2006)
5. Gazet, A.: Comparative analysis of various ransomware virii. J. Comput. Virol. **6**(1), 77–90 (2008)
6. Liao, Q.: Ransomware: a growing threat to SMEs. In: Conference Southwest Decision Science Institutes (2008)
7. Song, S., Kim, B., Lee, S.: The effective ransomware prevention technique using process monitoring on android platform. J. Mobile Inform. Syst. **2016**, 9 (2016)
8. Violet, B.: CryotoLocker's crimeware: a trail of millions in laundered Bitcoin, ZDNet, December 2013
9. Ahmadian, M.M., Shahriari, H.R., Ghaffarian, S.M.: Connection-monitor & connection-breaker: a novel approach for prevention and detection of high survivable ransomwares. In: Conference Information Security and Cryptology, pp. 79–84, September 2015
10. McAfee, McAfee Labs Threats report. http://www.mcafee.com/us/resources/reports/rp-quarterly-threats-mar-2016.pdf
11. Kim, D., Kim, S.: Design of quantification model for ransom ware prevent. World J. Eng. Technol. **3**, 203–207 (2015)
12. Kim, S., Park, J., Lee, K., You, I., Yim, K.: A brief survey on rootkit techniques in malicious codes. J. Internet Serv. Inform. Secur. **2**(3/4), 134–147 (2012)

13. Lee, K., Lee, Y., Park, J., You, I., Yim, K.: Security issues on the CNG cryptography library (Cryptography API: Next Generation). In: Conference Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 709–713, July 2013
14. Lee, K., You, I., Yim, K.: Vulnerability analysis on the CNG crypto library. In: Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 221–224, July 2015