IEICE TRANS. INF. & SYST., VOL.E100–D, NO.9 SEPTEMBER 2017

LETTER

CLDSafe: An Efficient File Backup System in Cloud Storage against Ransomware

Joobeom YUN[†a)], ***Member***, **Junbeom HUR**[††], **Youngjoo SHIN**[†††b)], ***and*** **Dongyoung KOO**[††††c)], ***Nonmembers***

SUMMARY Ransomware becomes more and more threatening nowadays. In this paper, we propose CLDSafe, a novel and efficient file backup system against ransomware. It keeps shadow copies of files and provides secure restoration using cloud storage when a computer is infected by ransomware. After our system measures file similarities between a new file on the client and an old file on the server, the old file on the server is backed up securely when the new file is changed substantially. And then, only authenticated users can restore the backup files by using challenge-response mechanism. As a result, our proposed solution will be helpful in recovering systems from ransomware damage.

key words: *ransomware, cloud storage system, fuzzy hashing*

1. Introduction

Over the past few years, ransomware has become popular among cybercriminals [1]. Ransomware locks the victim's computer until he makes a payment to re-gain access to his data. Although the first ransomware appeared in the world almost 10 years ago, the number of ransomwares was not significant before 2013. However, the ransomware threat made headlines as the most notable malware trend after targeted attacks in 2013 [2]. For example, the *Cryptolocker* ransomware alone infected approximately 250,000 computers globally in its first 100 days [3].

Given the significant growth of ransomware attacks, it is important to develop a protection technique against this type of malware. Some researches [2], [4] have emphasized that backing up data to a location where attackers cannot write to or erase is important. However, most users feel annoyed at regular backup of their data. Other research [1] proposed a ransomware detection method monitoring file system, which monitors repetitive reading-writing-deleting patterns in a short period of time. However, this approach has drawback in terms of false-positive rate because a normal program can show repetitive reading-writing-deleting patterns in a short time. In addition, file monitoring technique using filter driver results in decreasing stability and performance overhead. Other ransomware detections [5], [6] are also complicated because a file monitoring is needed at least and there are a lot of file operations in real system. Thus, as far as we known, most of the previous ransomware detection schemes suffer from detection accuracy or efficiency. In this paper, we propose CLDSafe, an efficient and ransomware-proof backup system, using cloud storage systems. CLDSafe is not about ransomware detection but about ransomware damage prevention.

Cloud storage systems store user data or files to remote cloud storage locations such as Dropbox [7] or Google Drive [8]. These systems automatically back up user configured folder files, but they have limitations of storage time and space. Whenever a file is changed, commercial cloud storage systems should save the old version file. Thus, frequent file changes result in the storage flooding. This is a denial of service condition. We call this attack a storage-consuming attack.

Meanwhile, CLDSafe backs up files only when they are changed substantially. That is, it copies old version of the file to a separate place only when a new version of the file has low similarity level compared with the old version. CLDSafe is implemented with fuzzy hashing (also known as context triggered piecewise hashing [12]). When a user identifies his files encrypted by ransomware, he can restore original files from the remote cloud storage.

The advantage of CLDSafe is described in detail as follows.

- CLDSafe backs up files more efficiently than commercial cloud storages.
- CLDSafe is resistant to storage-consuming attacks. It can detect and block storage-consuming attacks by a simple algorithm.
- CLDSafe is light-weighted. It does not require lots of resources such as memory or CPU of the cloud system.

2. Background

2.1 Ransomware

Ransomware is a type of malware that restricts access to the infected computer system, and demands that the user pays a ransom to the attacker to remove the restriction. Some forms of ransomware encrypt files on the system's local storage,

Manuscript received March 7, 2017.
Manuscript revised May 10, 2017.
Manuscript publicized June 12, 2017.

[†]The author is with Dept. of Computer and Information Security, Sejong University, Seoul, Korea.
[††]The author is with Dept. of Computer Science and Engineering, Korea University, Seoul, Korea.
[†††]The author is with School of Computer and Information Engineering, Kwangwoon University, Seoul, Korea.
[††††]The author is with Dept. of Electronics and Information Engineering, Hansung University, Seoul, Korea.
a) E-mail: jbyun@sejong.ac.kr
b) E-mail: yjshin@kw.ac.kr (Corresponding author)
c) E-mail: dykoo@hansung.ac.kr (Corresponding author)

DOI: 10.1587/transinf.2017EDL8052

which become almost impossible to decrypt without paying the ransom for the decryption key; while some may simply lock the system and display messages intended to coax the user into paying [10].

Ransomware can be classified into two classes: Class A ransomware overwrites the contents of the original file, so the deleted file cannot be restored. In this class, encrypted filenames are the same as the original ones. Class B ransomware makes another encrypted file (for example, *B.txt.crypt*) using the original file then deletes the original file after encryption. The encrypted files have other file extensions and sometimes the original files can be restored because they are not overwritten. Class B ransomware is easy to detect because files having a specific file extension are created. Whereas, class A is more serious because the original files cannot be restored in practice, so we will concentrate on class A ransomware in this study.

2.2 Cloud Storage System against Ransomware

Utilization of cloud storage systems to prevent ransomware is an effective method. However, if a user's computer is infected with ransomwares such as *CryptoLocker*, files on his local storage will be held to a ransom and their copies in the cloud storage are overwritten when his computer is synchronized with the cloud storage system [11]. Then, his backup storage is also now held as a hostage. In order to prevent this problem caused by the autonomous synchronization, it is needed to keep data files to separate and secure cloud storage areas, which users cannot access. Unfortunately, however, commercial cloud storages do not provide separate places. Thus, we propose a novel cloud storage system that supports separate cloud storage for secure file backups through challenge-response authentication.

Dropbox keeps all backups in the cloud storage until the amount of stored data reaches its limit; most of the free users have 2GB limitation. However, keeping all backups in the cloud storage is a storage-consuming mechanism. Our system identifies almost identical files in order to back up files efficiently. If a new version is significantly different from a previous version, our system keeps a previous version file in another separate place. This place is called *secret area* because ordinary cloud storage programs cannot access this area, but only an authorized user can access here through user authentication. Although several cloud storages such as Google Drive [8] or MS OneDrive [9] provide file versioning services, they are also vulnerable to storage-consuming attacks.

2.3 Identifying Almost Identical Files

CLDSafe identifies almost identical files using fuzzy hashing, also called context triggered piecewise hashing [12]. This can match a file up with another having similarity. Similarity in two files means that the two files have almost same order bytes although they have the different length. *ssdeep* [12] is a program to compute context triggered piecewise hashes. It can compute the similarity of files. File similarity information is stored in metadata.

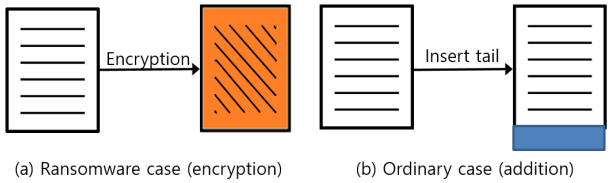


Fig. 1 File update

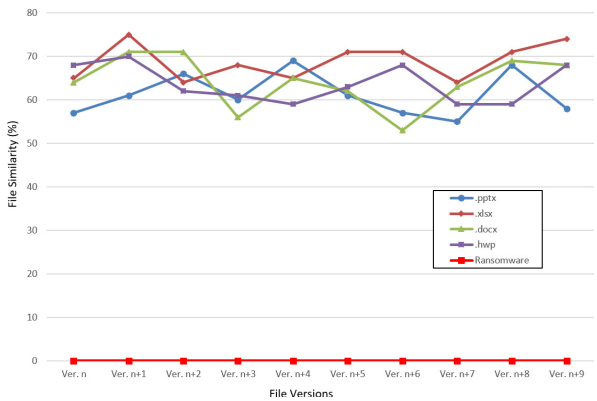


Fig. 2 Average file similarity flow

Figure 1 shows file changes when a data file is updated by ransomware and ordinary cases, respectively. Figure 1 (b) shows an ordinary case when a user updates a file, which represents a data addition or modification in the tail, head, or body. Meanwhile, file encryption by ransomware results in completely different contents, which is the case in Fig. 1 (a). The file similarity can be measured by using *ssdeep* program.

Figure 2 shows the experimental results of 210 file modification cases using *ssdeep*. The first 200 cases were ordinary cases. We selected 4 kinds of data file: *.pptx, .xlsx, .docx,* and *.hwp*. Then, we got 5 samples per each kind of file, respectively. We measured the similarities 10 times per one data file, respectively. The second 10 cases were ransomware cases; 10 files were infected by *Cryptolocker*. *Cryptolocker* used RSA encryption algorithm with 1024-bit RSA public key. As shown in Fig. 2, ordinary cases show high similarity scores; they are above than 60. This means that above 60% blocks of a file are the same as blocks of the other files. Encrypted files have low similarity score, which is almost 0 compared with the original file.

3. CLDSafe

3.1 System Architecture

CLDSafe is executed on cloud storage servers. Cloud storage servers automatically back up user files, and they overwrite old files when the files are updated. CLDSafe examines whether the file is changed substantially or not. If the similarity of the new file is above threshold θ compared with

Table 1 Stored file list

Blocksize	Hash1	Hash2	Filename
24576	XI9COtZW1zXJ8muRZWkzI1Pi436BqY0wH2fwz+/gXvwZWBzYbPsqx	XI9COtZW1z58muRZWkzazNc2fwq/gXvi	A.pdf
393216	IA89gDcZ7xl4H2rJO6pvdDzjikQM/MRM4HH	IfVxl4lfvd+9kM1HH	B.pptx
49152	XFsPc/57m9b5ytsieDrkYupPT2sYba5Y3a0nPbXw2	XFsk/m5YcOuuBEV	C.zip
...	...	...	...

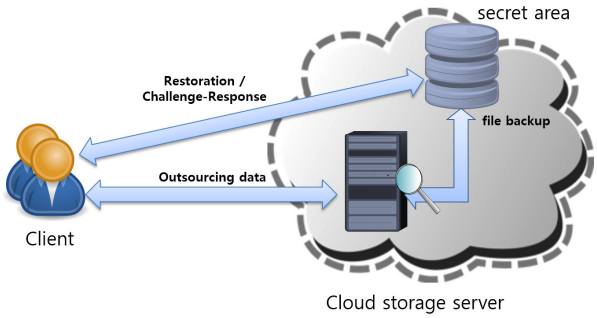


Fig. 3 System architecture of CLDSafe

the old file, it is considered as a usual modification and not saved in the secret area. On the other hand, if it is below θ, which might be the case of ransomware infection, the old file is restored from the secret area. Figure 3 illustrates this backup and restoration process.

In CLDSafe, a synchronized folder is set at the client side. Then, files on the folder are automatically backed up with the cloud storage server. On every file update at the client side, CLDSafe on the server examines whether the updated file is significantly different from the previous version of the file in the cloud server or not. If its similarity is below threshold θ, the program stores the previous version to a secret area in the cloud storage server, because it might be the case of ransomware infection. Afterwards, then, users can get the original files from the secret area after authentication. Clients can be authenticated by challenge-response method using their credentials such as password.

3.2 Proposed Scheme

Data Upload. For a set of uploaded files π on a cloud client $C_i, \pi = \{F_1, F_2, \ldots, F_n\}$, if $\pi \subset C_i$ then $\pi \subset S$, where S represents the cloud storage server.
Data Backup. For a new uploaded file F' on a cloud client C_i, where $F' \in C_i$ and $F \in S$, if SIM$(F, F') < \theta$ then execute SECRET_SAVE(F) in secret area.

- SIM$(F1, F2)$: This is a similarity computing function. It receives two filenames as inputs and produces a calculated similarity score as an output using fuzzy hash. The score is from 0 to 100; 0 means that two files are completely different and 100 means that two files are the same. In the similarity check procedure, CLDSafe keeps metadata for better performance, which is a file list having information about the stored files in the cloud storage server. Table 1 shows a stored file list. It has four columns: block size, hash1, hash2, and filename. The block size indicates the size of the input block. Hash1 is computed by using block size, and hash2 is computed by using twice the block size. The reason to use two block sizes is that it is possible to compare twice using two block sizes. The final component, filename, stores the file name. The metadata is used to compare two files, those are an old version and a new version of a file, when a file is updated. Because CLDSafe uses this metadata, it has better speed than reading the file itself.
- SECRET_SAVE(F): This function receives a filename as an input and copies the file to a user-inaccessible cloud storage area. Saved files can be restored after a user is authenticated via challenge-response mechanism (for example, ID and password).

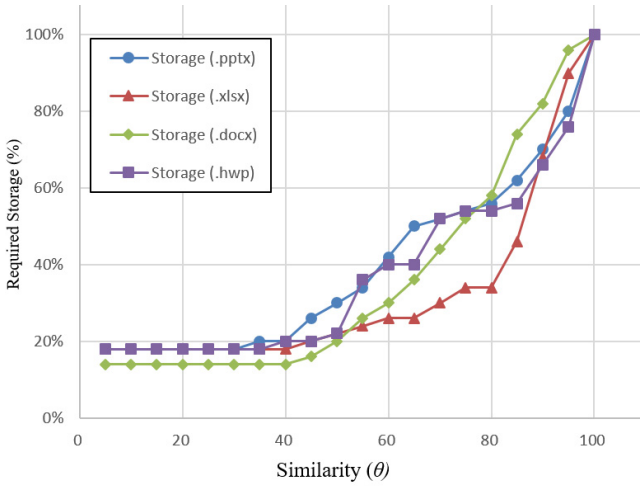
Attack Detection. For successive files $F1, F2, F3, F4$ on a cloud server S, if SIM$(F1, F2) < \theta$ and SIM$(F2, F3) < \theta$ and SIM$(F3, F4) < \theta$ then execute ALERT(F).

- ALERT(F): This is alerting function. It alerts a storage-consuming attack warning to a user with file information.

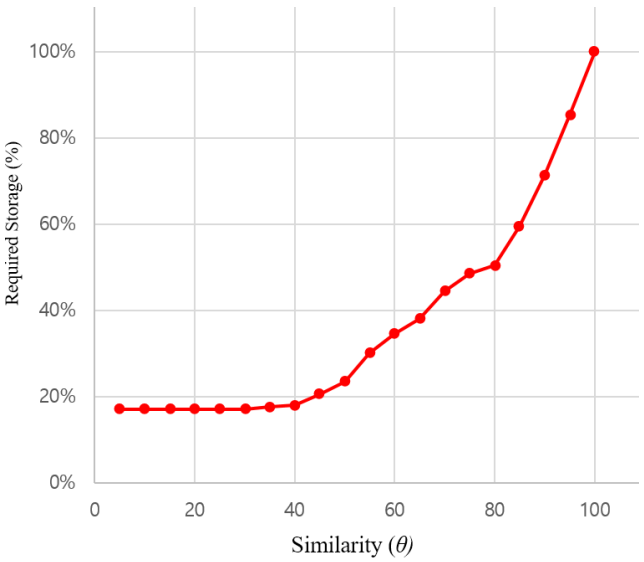
Restoration. When a user is notified of the ransomware infection, he can restore backup files from a secret area of the cloud storage server after additional authentication procedure.

4. Evaluation

We evaluated CLDSafe with regard to the storage overhead. Figure 4 shows the storage overhead with different file similarity θ. Figure 4 (a) shows the storage overhead with different file types, which are pptx, xlsx, docx, and hwp. First, we measured the required space when θ is 100. This needs full storage space for all file versions. Other storage overheads are relative storage requirement compared with a full storage requirement. As the similarity increases, the storage requirement also increases. This phenomenon is commonly observed in all of the four different file types. Figure 4 (b) shows the storage overhead on average among all



(a) Storage overhead with different file types



(b) Storage overhead on average

Fig. 4 Storage overhead

of the required storages in Fig. 4 (a). When θ is 90, it backs up 72% of all versions on average. This means that files having below 90% similarity are 72% of all versions on average. When θ is 80, it backs up 51% of all versions on average. In CLDSafe, we set θ as 80 because it stores almost 51% of all file versions, and each of them is only 20% different, which is the maximum cost-effective point. As a result, CLDSafe can save 41% storage space compared with full versions backup mechanism, which is a typical backup strategy of many commercial cloud storage services. Therefore, CLDSafe is more efficient than commercial cloud storages against storage resource-consuming attacks.

5. Conclusion

Ransomware becomes more and more threatening nowadays. In order to prevent ransomware, we proposed CLDSafe, a novel and efficient file backup system. It keeps shadow copies of files and provides the safe restoration when a computer is infected by ransomware. After CLDSafe measured the file similarity between an original file and a new file, it stored safely the original file when the new file was changed substantially. Only authenticated users can restore the original file by using a challenge-response mechanism. CLDSafe will be very helpful in preventing ransomware.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No.2015R1C1A1A02036511, No.2017R1C1B5015045). This work was supported by the research fund of Signal Intelligence Research Center supervised by the Defense Acquisition Program Administration and Agency for Defense Development of Korea.

References

[1] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda, "Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks," DIMVA 2015, pp.3–24, 2015.
[2] Internet Security Threat Report, http://www.symantec.com/security_response/publications/threatreport.jsp, 2014.
[3] Cryptolocker Ransomware, https://www.secureworks.com/cyber-threat-intelligence/threats/cryptolocker-ransomware/, 2014.
[4] How to Protect Yourself from Ransomware, http://www.howtogeek.com/174343/ransomware-why-this-new-malware-is-so-dangerous-and-how-to-protect-yourself/
[5] N. Scaife, H. Carter, P. Traynor, and K.R.B. Butler, "Cryptolock (and drop it): Stopping ransomware attacks on user data," 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), pp.303–312, 2016.
[6] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, E. Kirda, "Unveil: A large-scale, automated approach to detecting ransomware," USENIX Security Symposium, 2016.
[7] Dropbox, https://www.dropbox.com/
[8] Google Drive, https://www.google.com/drive/
[9] OneDrive, https://onedrive.live.com
[10] Ransomware Malware: Everything You Need To Know About It, http://beebom.com/2015/01/ransomware
[11] CryptoLocker Virus Affects Google Drive Files — Is Your Cloud Safe?, http://www.datto.com/blog/cryptolocker-virus-affects-google-drive-files
[12] J. Kornblum, "Identifying almost identical files using context triggered piecewise hashing," Digital investigation, vol.3, pp.91–97, 2006.