Secret-Stamp: Document Leakage Prevention via Steganographic Marking

Sang-Hoon Choi, Sung-Kyu Ahn, Dongmin Yang, Ki-Woong Park†
System Security Lab. Kongju National University
System Security Lab. Daejeon University
Dept. of Informtaion Communication
Dept. of Computer and Information Security Sejong University
csh0052@gmail.com, yiimfn@gmail.com, dmyang@dju.kr, woongbak@sejong.ac.kr

***Abstract*—** The leakage of secret documents through a network or portable disk is constantly increasing. Though various ways to prevent such accidents have been proposed, most schemes have an operational dependency on its application. This paper proposes a scheme, termed *Secret-Stamp*, to prevent a leakage of secret documents without imposing dependency on its application. To achieve it, *Secret-Stamp* inserts a steganographic pattern into an unused region of the designated document file, and traces the designated document file. If *Secret-Stamp* detects a leak of the files through the network or via a portable disk, it blocks the related operations. Since the proposed scheme does not have any dependency on applications, it can be used on various file formats and platforms.

Keywords : File Monitoring, Data protection, Data exfiltration

Ⅰ. Introduction

Information leakage in enterprises is constantly increasing. The leakage of secret documents is expected to cause financial and technical damages in terms of enterprise operations and national infrastructure [1]. Most of these secret information leakage accidents are generated via the network such as browsers, e-mail, etc. or via portable disks such as USB.

There have been various schemes to prevent such accidents. One of them is to use DRM (Digital rights management) [2, 3] technology to prevent the leakage of secret documents. However, applying DRM technology into document leakage prevention systems has challenging issues because it cannot be used on various file formats and platforms. Another scheme is to apply the content filtering technology [4, 5] into document leakage prevention systems to analyze the overall contents of the file and to control the access for the files. However, it has a critical limitation because it cannot monitor the access transactions for the files which are encrypted or compressed.

As a remedy to the problems, we suggest *Secret-Stamp* which is to protect the document safely and to solve the existing challenging issues above mentioned. *Secret-Stamp* inserts a steganographic pattern into an unused region of the designated document file. *Secret-Stamp* then monitors the inserted pattern and if it detects the leaking of documents with the inserted pattern on the network or portable disk, it blocks the related operation. Since the proposed scheme does not have any dependency on its application, *Secret-Stamp* can be applied into various file formats and platforms.

This paper is organized as follows. In Section 2 presents limitations of existing schemes for documentation leakage preventions, and Section 3 proposes a document leakage prevention system based on steganographic marking. In Section 4, our proposed scheme is verified that blocking leakage of document can be applicable to the network and the portable disk. Finally, in section 5, we describe our conclusions and future work.

Ⅱ. Related Work

A. Prevention of document leakage using DRM

DRM is a technique for preventing illegal copying of content. Since DRM has dependency on its applications, it cannot be applied into various file formats and platforms. In addition, applying DRM technology into document leakage prevention system provides inconvenience for users accessing files because it requires authentication process whenever a document is accessed.

† Corresponding Author
This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (2016R1A4A1011761) and by the Ministry of Education (2015H1C1A1035859).

B. *Detectoion of document leakage using file content filtering*

Managing secret documents based on file content filtering technology is a technique that defines and detects specific patterns such as personal information (resident registration numbers, contacts, etc.) [4, 5]. However, it causes user-obstructive latency because of performing pattern analysis for overall contents of all documents. It has also challenging issue to find specific patterns from documents that are encrypted or edited by a specific document editor (Microsoft Office [6]).

Ⅲ. Design and Implementation of Secret-stamp

This paper suggests *Secret-Stamp* to prevent the leakage of secret document. Conventional schemes have dependency on its application. As a remedy to the problem, we propose a scheme which allows the secret documents to be accessed without any dependency with our schemes. Consequently, our proposed schemes can prevent leakage accidents. Our proposed scheme is to insert a steganographic pattern into an unused region, monitor the inserted pattern, and block the leakage detected. To achieve it, we analyzed the file formats of Microsoft Office programs such as PPT, XLS and etc.

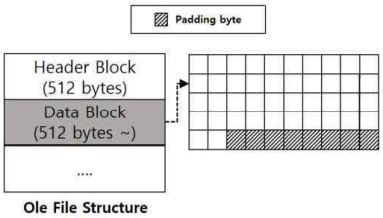

Fig 1. The Structure of OLE File

Microsoft's Office Program has OLE [7] as a file format. As shown in Fig 1, OLE is divided into a header block with 512 bytes and a data block larger than 512 bytes. We found one problem that inserting a false pattern into an OLE file may damage the file. Thus, we devised a scheme to insert a specific pattern into multiple data blocks. We found the region of a padding byte used to standardize size of blocks in data blocks. However, there was the other problem. If the contents of the file were modified, the region of the inserted pattern was filled with padding byte again. To solve this problem, when the modification in the file is finished, we figured out the scheme that inserts a specific pattern into the region that would have been filed with padding byte, monitoring the file in real time. This scheme that we propose has independency on applications and because part of the file is monitored, it causes low overhead.

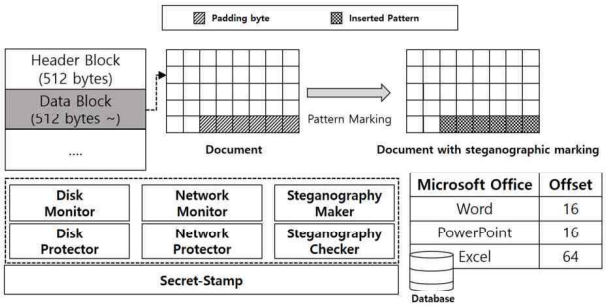

Fig 2. The Structure of *Secret-stamp*

We define and design this scheme as *Secret-Stamp*. As shown in Fig 2, the region where is filled with padding byte was extracted. The offset that contains information about the region was stored in the database each file formats. We implemented a module called steganography marker that inserts a specific pattern into the extracted region. And we also designed a module called steganography checker that detects a specific pattern and the time when the file is modified. We implemented modules that block the related operations if a leakage of the file is detected by monitoring the disk and the network and that are called each disk monitor module and network monitor module.

Ⅳ. Experiment

We establish a hypothesis about two situations that leaked documents and verify the fact that *Secret-Stamp* can protect them through the experiment by inserting the steganography marker. The first situation is that a file whose steganographic is marked is leaking through file copying. The second situation is that a file is leaking through e-mail or file transfers.

A. *A leakage of secret documents through portable storage device*

Disk monitor module is implemented inside a *Secret-Stamp* to prevent secret documents from being copied through a portable disk. In this module, when user's computer recognizes the disk, if a file that has steganographic mark is created or copied on the disk, the file is securely erased by using P.Gutmann [8] algorithm to prevent the file from file-carving [9]. Additionally, log information about secret documents access is recorded on the database. As a result of copying secret documents to the portable disk, it was impossible to copy and file-carving these documents.

B. *A leakage of secret documents over the network*

The network monitor module is implemented inside a *Secret-Stamp* to prevent the leakage of secret documents through the Internet or e-mail. This module is designed by using the WinDivert library [10]. It can filter, capture, modify, and block network packets. The network monitor module performs filtering on the file transfer and

determines whether or not the contents of the filtered packet have steganographic mark. If the file that is sent has steganographic mark, the packet is blocked and is recorded on the database. As a result of sending secret documents through e-mail, it was impossible to send these documents.

Ⅴ. Conclusion

In this paper, we present a steganographic marking to the region where a specific pattern is inserted of the file, and suggestions on how to prevent a leak of secret documents through file access monitoring. Our proposed scheme, *Secret-Stamp,* can be one of the solution solve the problems. Through two experiments, we found that our steganographic marking scheme could be useful to prevent the leakage of secret documents. As a further work, we will revise techniques to track and block the path of the file in a computationally efficient manner.

References

[1] GLOBAL DATA LEAKAGE REPORT 2015, https://infowatch.com/report2015

[2] Subramanya, S. R., and Byung K. Yi. "Digital rights management." IEEE Potentials 25.2 (2006): 31-34.

[3] Liu, Qiong, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard. "Digital rights management for content distribution." Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21. Australian Computer Society, Inc., 2003.

[4] Shabtai, Asaf, Yuval Elovici, and Lior Rokach. A survey of data leakage detection and prevention solutions. Springer Science & Business Media, 2012.

[5] Katz, Gilad, Yuval Elovici, and Bracha Shapira. "CoBAn: A context based model for data leakage prevention." Information Sciences 262 (2014): 137-158.

[6] Microsoft Office, https://www.office.com/

[7] OLE, Brockschmidt, Kraig. Inside Ole. Microsoft Press, 1995.

[8] Gutmann, Peter. "Secure deletion of data from magnetic and solid-state memory." Proceedings of the Sixth USENIX Security Symposium, San Jose, CA. Vol. 14. 1996.

[9] Pal, Anandabrata, and Nasir Memon. "The evolution of file carving." IEEE signal processing magazine 26.2 (2009): 59-71.

[10] WinDivert, https://reqrypt.org/windivert.html