Advanced Science and Technology Letters
Vol.30 (ICCA 2013), pp.45-50
http://dx.doi.org/10.14257/astl.2013.30.09

Secure USB using Reconstructed File Structure

Jae-Hong Youn[1] and Yoo-Kang Ji[2]

[1]Digital contents cooperative research center,
Dongshin University, Korea
[2](Corresponding Author)Dept. of Information & Communication
Engineering, Dongshin Univ., Korea
[1]jhyoun@dsu.ac.kr, [2]neobacje@gmail.com

Abstract: This paper we proposed advanced methods of USB security software using file system recomposition and unassigned arbitary disk area. It provides data confidentiality of the USB storage device in the Windows OS using forced disk partition division by offering the real time data password in secure area. The confidentiality is also served by using user login verification and saving encoded the hash value at the unassigned arbiraty disk partition.

Keywords: USB, memory security, non-partition

1 Introduction

USB memory has simple internals and no mechanical working so it fails less than storage devices like hard disk. On the other hand there are some problems like unrecognized partition from unusual seperation when in use and data loss and damage by viral infection.

Nevertheless USB memory is generalized as a portable storage device by virtue of its convenience and mass storage. Loss and theft can cause personal information leaks and there are many plans for these accidents.

The recent USB secure system causes security problems like encoding method, storage of the key and indirect user verification and gets user hostile according to encryption time and key management. Therefore research for more convenient and safer portable storage device security become emphasized the need.

This paper proposes the USB memory security using file system recomposition at the unassigned arbirary disk partition.

2 USB Security Requirements

USB security system offers security function into stages by the purpose and used place, that should be designed for usability and efficiency. Basically security solution for portable storage device should fulfill these requirements as following

Access control: In the USB memory secure area all actions like reading and modifying data should be controlled establishing lines of authority about them in order to block in advance unapproved attempts to approach.

Verification: Only verified user should be approached to the secure area in USB memory secure area, and indirect user verification not be possible.

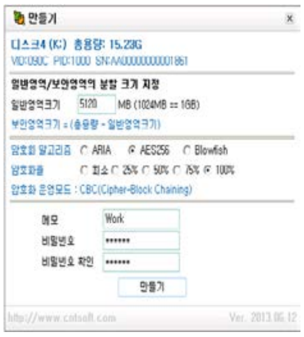
Confidentiality: Data of USB storage should be checked only by the right object and not by hackers.

Efficiency: Service should be easy, and quality be cost-efficient.

3 Secure USB Generating and Configuration of Encryption Environment

Fig.1 shows Non Secure Volume Size Difime, Cipher Algorithm Method (Choose Cipher Algorithm(ARIA, AES256, Blowfish) and Cipher Rate (Minimum, 25%, 50%, 100%)), Secure USB Volume Labeling and initialize User Login Password.

The reconstructed file system structure used in secure USB is presented in Fig.3 as FAT32 VBR, and it's used through secure USB Driver. By using reconstructed file system structure, data cannot be accessed properly through existing file system structure. If an unverified user wants to access to the file system in secure area, he should analyze the reconstructed file system structure first and then access is possible. That's the additional access control to the USB Storage.


만들기

디스크4 (K:) 총용량: 15.23G
VID:090C PID:1000 SN:AA00000000001861

일반영역/보안영역의 분할 크기 지정
일반영역크기 5120 MB (1024MB == 1GB)
보안영역크기 = (총용량 - 일반영역크기)

암호화 알고리즘 ( ) ARIA [✓ AES256] ( ) Blowfish
암호화율 ( ) 최소 ( ) 25% ( ) 50% ( ) 75% [✓ 100%]
암호화 운영모드 : CBC(Cipher-Block Chaining)

메모 Work
비밀번호 ••••••
비밀번호 확인 ••••••

Fig. 1. Secure USB Partitioning Configuration

3.1 Secure USB Storage Processing mechanism

USB storage processing mechanism for user verification and data encoding through Logical disk volume partitioning and file system reconstructing as shown in Fig.5 is composed of the stage creating the key and hash code for user verification and data encoding and saving at unused space, the stage for user verification, and the stage data coding and encoding by the verified user.

Level 1. Creation process of the message and the key for user verification and data coding.

(1) Create the hash code for user password verification and the 32 byte random key (K) for data encoding.
(2) Create 32 Byte Random Message (M) for Hash function.
(9) Insert the initial user password.
(3) Create hash code (hc1) using hash function from M and PWu of (2) and (9).
(4) Create hash code (hc1) using 32 byte message digest.
(5) Encrypt K from (1) and hc1 from (4) by encoding algorthm.
(6) Save Ehc1[K] from (5) and M from (2) to the unused space.

Through this process by saving Ehc1[K] created from hash code and key encryption and the message used for hash function to unused space, key for encryption and user password aren’t exposed to the unverified user directly.
In the second stage the any 32 byte text of (7) is saved at the encrypted Ek[T] Unused space of (8) using created key of (1), and in the user verification request the user verification is performed by checking the coincidence of Text String (T).
The third stage is the user verification process when a user wants to access to the secure USB Storage, and it is as in the following.

(9)’ Insert the user password (PW1)
(10) Create the 32 byte message digest from inserted user password (PW1) and M saved
in (6) by using hash function used in (3).
(11) Create Hash Code (hc2) using 32 byte message digest.
(12) Read Ehc1[K] created in (5) from (6) and decode to Hash Code (hc2) created in (11)
and then create KD.
(13) If user password inserted in (9)’ and the one set in (9) are the same, K from (1) and
KD would be the same.
(14) Ek[T] created in (5) and saved in (6) is decoded to KD created in (13) and create (15)
TD.
(16) If T of (7) and TD of (15) are the same, verify the valid user.
(17) The user verified in (16) uses KD created in (13), and data coding-decoding is performed in real time.

Provided is the secure area which can offer the safe user verification method blocking indirect user verification from memory dump attack and unusal seperation, the compatibility and the safety.

4.4 Execution Mount of Secure USB Storage

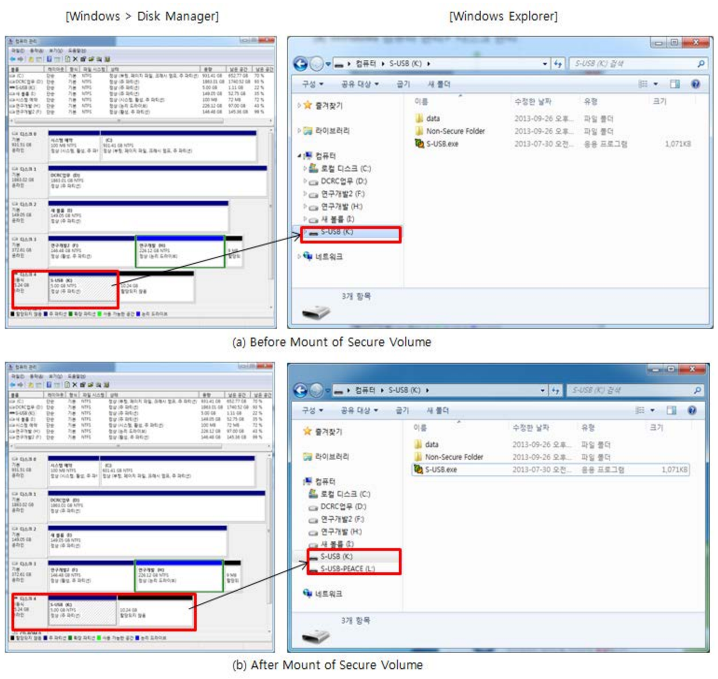

[Windows > Disk Manager]
[Windows Explorer]
(a) Before Mount of Secure Volume
(b) After Mount of Secure Volume

Fig. 2. Secure USB Storage Mount

In the case after secure USB runs a user doesn’t login the non secure area is mounted as in the Fig.6(a), and the case the secure area is mounted after login through the secure USB S/W is shown in Fig 6(b).

Partitioning cannot be performed by Windows general function, therefore secure volume guarantees the safety and even after format the secure volume remains safely.

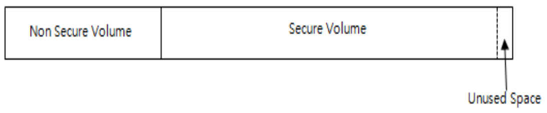


Fig. 3. Secure USB Storage at Disk Manager.

Even when mounted secure area is recognized as disk volume, it’s still shown as not allocated area to the disk administrator. Also the area shown as not allocated includes unused space for saving the key, though internally the secure area and unused space are separated as shown in Fig.7.

5 Conclusion

The secure USB this paper proposed using Logical Volume Partition makes secure and non- secure area possible to be used in one USB. In non-secure area the Window File I/O Driver makes user access, and in secure area recomposing the file system structure the access is possible by user verification and encoding-decoding process through the secure USB Driver.

Disk volume of secure area gets mounted by active user demand, and thereby firstly visible access control is performed. Also the file system structure in secure area recomposes the structure Windows offers, and the structure about the volume in secure area which will be mounted in Windows hooks disk I/O messege and gets mounted actively, that is secondary user access control.

Data confidentiality is served by real time data code about secure area.

By encoding at the unassigned arbitary area in order to save hash code used in user verification the security problems like missed key caused by Boot Record damage and leaked key by memory dump can be solved.

This paper develops the security solution research for movable storage device when loss and theft of USB memory lest personal information is disclosed by malicious third party, and utilizes the structure and characteristic of file system for the safe user verification and offer of secure area.

References

1. Dung Vu Pham, Ali Syed, Malka N. Halgamuge, "Universal serial bus based software attacks and protection solutions", Digital Investigation, Vol 7, pp.172-184, Feb (2011).
2. An Wang, Zheng Li, Xianwen Yang, Yanyan Yu, “New attacks and security model of the secure flash disk”, Mathematical and Computer Modelling,Vol 57, Issues11-12, pp.2605–2612, June (2013).
3. A. Menezes, P. Van Oorschot, S. Vanstone Handbook of Applied Cryptography CRC Press (1997).
4. FreeOTFE Develop Group, “FreeOTFE user manual”, January 30, (2008).
5. Sophos PLC. “Data sheet of safe guard removable media”, November 1, (2009)
6. CE-Infosys, “Data sheet of compuSec mobile hardware security for notebooks”, May 20, (2009).
7. T. Espiner, “Kingston flash drives suffer password flaw”, ZDNet UK, (2010).
8. Syss, “Cryptographically Secure SySS Cracks a USB Flash Drive”, (2010).
9. L. Lamport “Password authentication with insecure communication Communications of the ACM”, 24 (11), pp.770–772, (1981).
10. Z. Li, L. Zhang, Y. Liu, “Foundations of Cryptographic Engineering Information Science and Technology” Institute Press (2008).
11. Minho Kim, Hyunuk Hwang, Kibom Kim, Taejoo Chang, Minsu Kim, Bongnam Noh1, “Vulnerability Analysis Method of Software-based Secure USB”, Vol 22, No.6, pp.1345-1353, (2012).
12. Jewan Bang, Byeongyeong Yoo, and Sanjin Lee, “Secure USB Bypassing Tool,” DFRWS 2010, pp. 114-120, Aug. (2010).

13. Robin Snyder, “Some Security Alternatives for Encrypting Information on Storage Devices,” InfoSecCD '06 Proceedings of the 3rd annual conference on Information security curriculum development, pp.79-84, (2006).
14. Stefan Balogh and Matej Pondelik, “Capturing Encryption Keys for Digital Analysis”, IEEE:International Conference on Intelligent Data Acquisition and Advanced Computing Systems, vol 2, pp.759-763. Sep. (2011).