Safe Authentication Protocol for Secure USB Memories

Kyungroul Lee[1], Hyeungjun Yeuk[1], Youngtae Choi[1], Sitha Pho[1], Ilsun You[2] and Kangbin Yim[1]
[1] *Soonchunhyang Univeristy, Dept. of Information Security Engineering,*
646 Eupnae, Shinchang, Asan, Korea
{*carpedm, goodyug, cocoip, schpho, yim*}*@sch.ac.kr*
[2] *Korea Bible Univeristy, School.of Computer Science,*
205 Sanggye7, Nowon, Seoul, Korea
isyou@gmail.com

Due to its portability and accesibility, the USB memory has become one of the most popular storage devices. However, if the device is lost, stolen or hacked, it can lead to critical information leakage. It is natural that malicious insiders would try to thieve their colleagues' USB memories. Consequently, various security-incorporated USB products have been developed. To our best knowledge, there is lack of security analysis and comparison on them. In this paper, we explore the authentication protocols for the secure USB memory while analyzing their vulnerabilities. Also, we classify the vulnerabilities into 12 categories, based on which the protocols are then compared. In addition, some recommendations are given to address the vulnerabilities. It is expected for commercial secure USB products to provide enhanced security after a thorough revise on the authentication protocols of their software based on the introduced vulnerability categorization.

Keyword : Authentication protocol, Identification, USB Flash Memory, Reverse engineering

1 Introduction

Due to the evolution in computer science and technologies, digitalized identity information or private data for individuals and companies have been rapidly increasing in this modern society. Accordingly, there has been a requirement to plan in advance to prevent many problems related to the disclosure of the information.

Meanwhile, the USB memories have become prevalent as personal storages for individuals, enterprises and governments thanks to their portability and accessibility. The portability of the memories sometimes can result in critical information leakage when they are lost, stolen or hacked because they usually store many kinds of important data[1]. Due to this reason, various security-incorporated USB memories have been developed. In particular, many governments (e.g., South Korea) have released a regulation to mandatorily use these USB memories especially for public organizations. This regulation encouraged manufacturers to produce and drove users to adopt secure USB memories more than before. However, most of the commercial secure USB memories and their application softwares are proved through a sequence of analytical tests to have considerable vulnerabilities that result in exposing important information. These vulnerabilities can cause the secure USB memories to be easily compromised even though they are equipped with their own user and device authentication protocol[2]. It is worth noting that the authentication protocols for the devices are executed and handled by the host computer[3]. This means that it is possible to understand and attack the mechanisms of the protocols through the reverse engineering during transactions.

To our best knowledge, there is lack of security analysis and comparison on secure USB memories and their authentication protocols[4]. Motivated by this, in this paper, we explore the authentication protocols for secure USB memories. At the same time, we analyze their vulnerabilities, which are then

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 1, number: 1, pp. 46-55

classfied into nine categories. Based on them, the authentication protocols are compared, followed by some recommendations for addressing the founded vulnerabilities.

The rest of the paper is structured as follows. Section 2 firstly discusses technologies and approaches that are used for commercial secure USB memories. Section 3 analyses the existing authentication protocols as an aim to demonstrate vulnerabilities. Section 4 discusses several considerations on the vulnerabilities and their causes by categorizing them. Section 5 concludes with related future works.

2 Technological anatomy and classification of the secure USB memories

2.1 Technical aspects

Secure storage is technologically engaged with the user authentication and the data encryption regarding to the authenticity and the confidentiality[5].

- User authentication is required to register an oracle to be matched with candidates that are provided by users. Most important job for user authentication is to safely store and treat the oracle out of unauthorized accesses[6].
- Data encryption is possibly not required to have an oracle according to the implementation policy. Just checking the integrity of the data after decrypting the encrypted one would be enough, which means that secure integrity checking is most important job for data encryption.

2.2 Three approaches to the Secure USB

Commercial secure USBs are divided into three categories including software-only approach, hardware-supported partitioning approach, and hardware-based encryption approach according to how they implement the mentioned technologies.

- Software-only approach:
  This approach utilizes normal USB memory to provide secure storage. Because the USB memory itself has no security mechanism, only the managing software has responsibility on implementing user authentication and data encryption. In this approach, anti-reversing of the software should be most considered for security assurance.
- Hardware-supported partitioning approach:
  This approach uses a hardware device to support multiple partitions physically divided on the USB memory. One of the partitions is used for installing managing software or as a normal storage and the other is used for user work area that is to be encrypted and decrypted by the managing software.
- Hardware-based encryption approach:
  This approach incorporates a specific security hardware chip into the USB memory and delegates the user authentication and data encryption to the security hardware[7]. The managing software has a role only to control the transactions as an agent. However, Most of the USB memories in this category normally do not implement the data encryption in hardware and support only the user authentication.

3 Analysis on the Authentication Protocols for Commercial Products

In terms of security, there are many security problems need to be considered to protect the privacy information that could be eavesdropped by malicious users. To investigate these problems, we selected several security-certified USB memories popularly used in society these days in Korea and analyzed the security vulnerability to them. Because the commercial products were too various in their supported security criteria, we selected and analyzed three typical product types using different approaches based on the categorization above. We anonymize the names or manufactures of these commercial products because of privacy or social side-effects to the manufacturers.

3.1 Type A : software-only approach

The registration process for the product type A is very simple and includes communication of ID and PW in plaintext between the host and the device as depicted in figure 1. These host-device transactions are intended to memorize and recognize the device for the host side by storing and identifying the user-provided device information for verifying access permission. Because the transactions are easily captured by malware through sniffing the USB traffic[**?**], this approach looks expecting that the host has never been compromised and is still offline during the registration process. Furthermore, the authentication information stored in the memory is permanently in plaintext, it is possibly drawn back from the memory when it is missing or stolen.

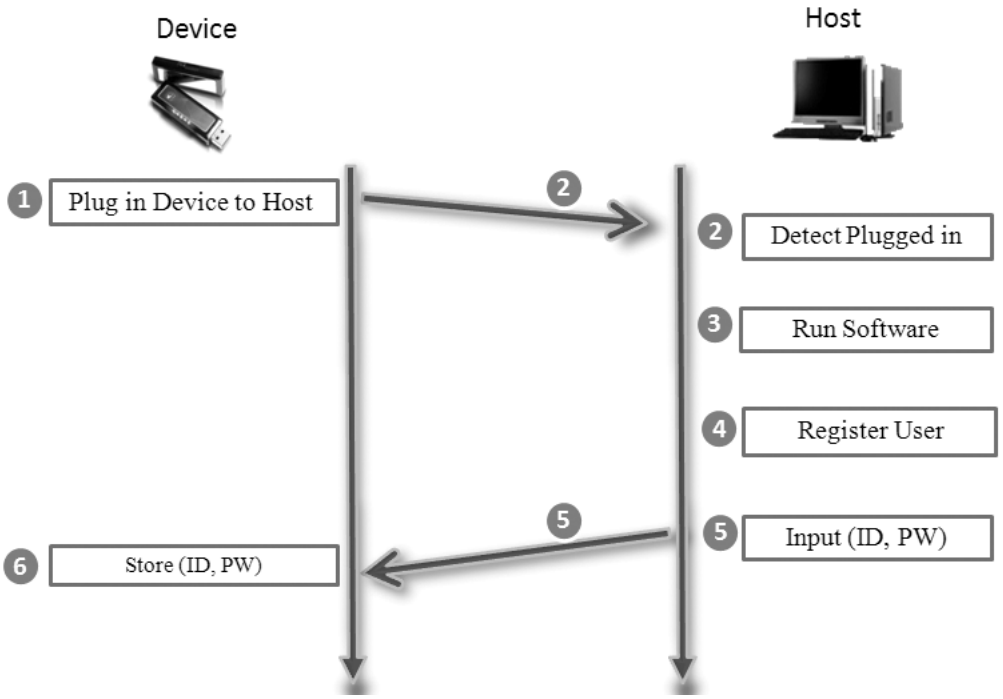

Figure 1: Registration process for type A

The demonstration at figure 2 describes the authentication protocol for the commercial product type A above mentioned. The host requests the authentication information stored into the memory to compare with the ID and PW acquired from user. And then, the device will retrieve and send the stored authentication information to the host. Because the authentication messages for this type also have no encryption, the information transacted between the host and the device is easily exposed. During the authentication,

the process runs on the host and it was really easy to capture the ID and PW in this research through sniffing on the work area for the USB host controller in the primary memory of the PC. Furthermore, the ID and PW that were stored into the memory at the registration step were not encrypted they were easily retrieved and transferred to the host through the standard USB commands.

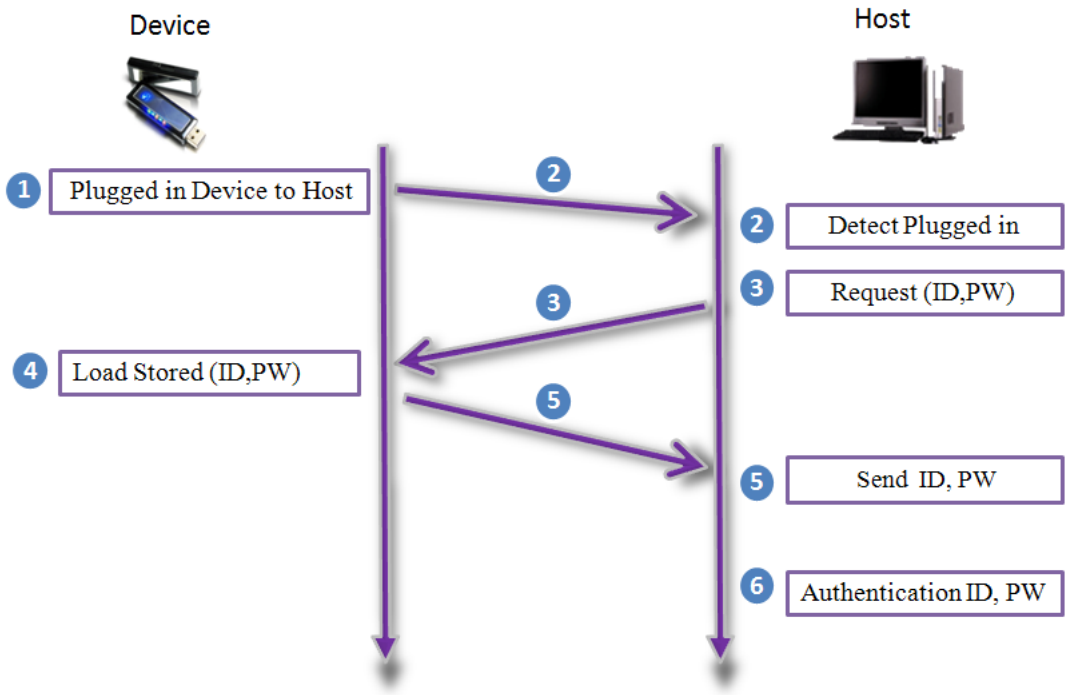

Figure 2: Authentication process for type A

3.2 Type B : hardware-supported partitioning approach

Product type B incorporates a hardware chip for dividing the memory storage into partitions. One of the partitions is initially mounted on the host and provides it as a work area from which the managing software is installed. The other partition is encrypted by the software after registration. Registration process requires user to input ID and PW to be used for both authentication and encryption. Some products using similar approach require two pairs of ID and PW to be used for authentication and encryption separately or support hardware-based encryption. Because it is reasonable for data encryption functions only to check the integrity of the data after decryption, it is not required to store ID and PW for future matching. However, the authentication process requires storing ID and PW in a secure form in a secure area though the product type B stores ID and PW as a decrypted plaintext.

During the registration process for type B as depicted in figure 3, the host software encrypts ID and PW by a unique session key and stores the session key and the encrypted ID and PW into the memory. However, the output buffer of the USB host controller is non-volatile as well as the input buffer, and the session key transferred to the memory is easy to be stolen.

For authentication, the host software of the type B shares a session key and the device encrypts the stored ID and PW during the host encrypts candidates of ID and PW that are acquired from user. The host then receives the encrypted ID and PW for matching. Figure 4 shows the sequence. In this case, even though an attacker can sniff the session key, it is difficult for the attacker to guess the relations

between the session key and the encrypted ID and PW. However, a brute-force trial is possible to find the relations. Furthermore, it is easy to deceive the host software into successfully going on with the authentication through the reverse engineering.

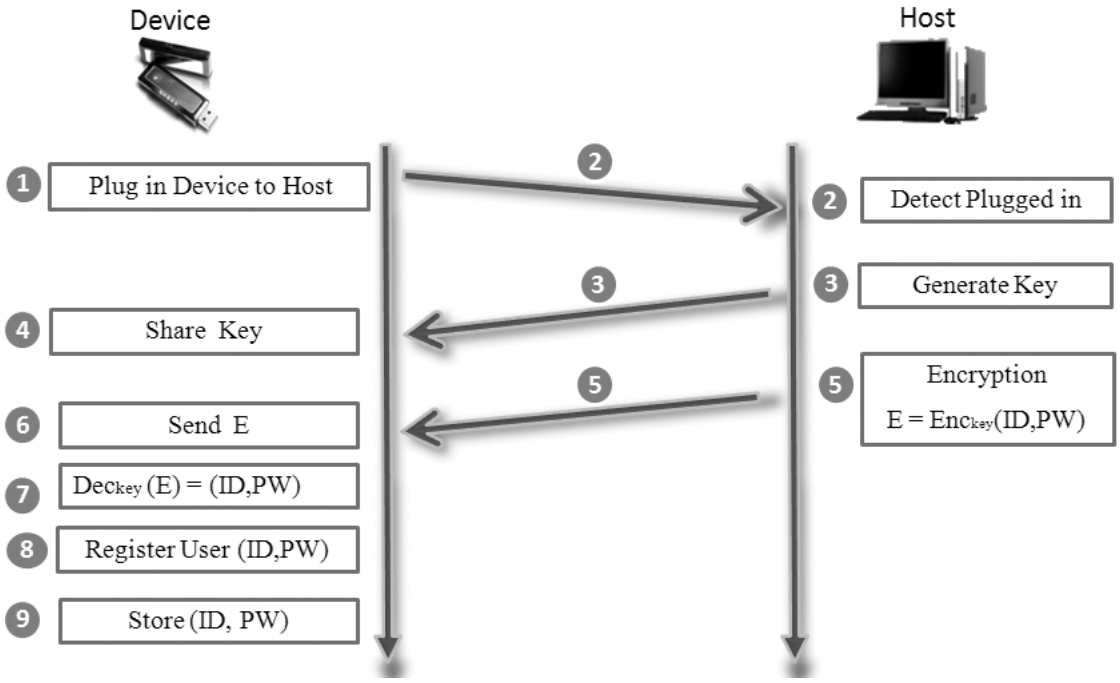

Figure 3: Registration process for type B

3.3 Type C : hardware-based encryption approach

One of major characteristics of product type C is that it uses off-line registration and challenge-response authentication. Off-line registration is an important step for authentication to make the security mechanism a fully-linked chain. During the off-line registration process, the intrinsic shared key key1is generated from user's ID and PW and it is implanted into the memory. The key is shared offline between the registration server and the device and the host retrieves it through the secure network from the server when the device needs to be authenticated. After authentication gets successful, the authentication key can be changed.

After issuing the memory through the off-line registration, on-line registration is required. On the on-line registration, a globally unique serial number is given and another shared key key2 is generated using a composite of ID, PW and the serial number for data encryption.

This encryption key for user data block is practically a hash of the truncated value of the authentication key key1and a fingerprint of the data block. Because the integrity checking mechanism is used to authorize the encrypted data area, key2 is securely transferred to the memory after encrypting it by key1 and stored into a secure block for matching with the signatures that are abstracted from the header of the decrypted data block.

During the authentication, host receives user ID and PW, retrieves key1 and the unique serial number from the server and the memory respectively and generates key2'. Thereafter, host authenticates the memory device using a nonce Rnd. Because this approach uses challenge-response authentication, the initial shared key key1 is not revealed out of the memory.

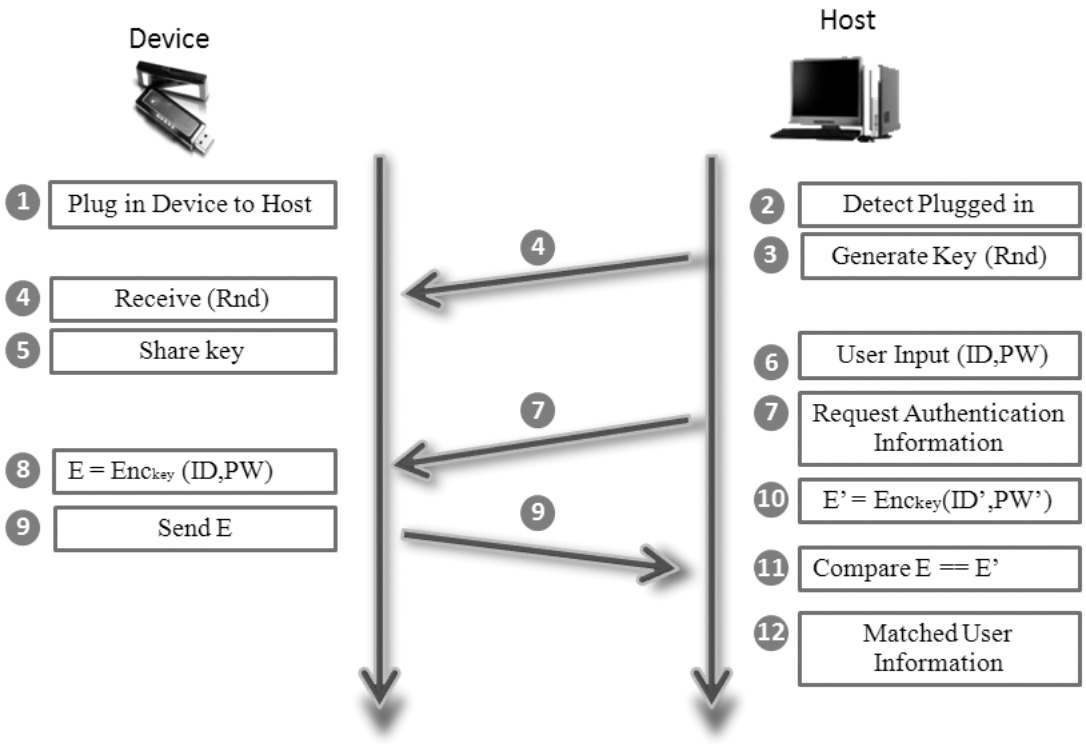

Figure 4: Authentication process for type B

If the authentication process successfully proceeds, a candidate for key2is generated, encrypted and transferred to the memory. In the memory, the candidate is decrypted and accordingly the encrypted user data block is partially decrypted using key2' and key2. If the results from the two different paths are matching, decryption on the rest of the data block proceeds and the block is given to the host.

The approach for type C gives better security compared to the others. However, comparison operation for matching on the host software is also easily compromised and some possible attackers can pass through the authentication process[9, 10].

4 Considerations on the Secure Authentication Solutions

The description on three different commercial product types A, B and C demonstrates much considerable vulnerability especially to the authentication protocols. We identified the found vulnerabilities to twelve specific ones and evaluate and compare the security levels of the product types based on them as shown in figure 6.

To generalize the vulnerabilities according to the differentiated attack points or techniques possibly available for malicious users, we classified them into mainly four categories as follows.

- *Whether the authentic information such as password, authentication key or encryption key is found on the USB transaction packets or in the host memory*: As a most severe insider threat called Memory Hacking, this vulnerability is easily targeted by the kernel mode sniffing malwares.
- *Whether the above authentic information is found on the USB memory itself*: In this case, finding or stolen memories can be investigated, analyzed or de-capped and the authentic information is collected at remote places by some hacking tools available on the Internet.

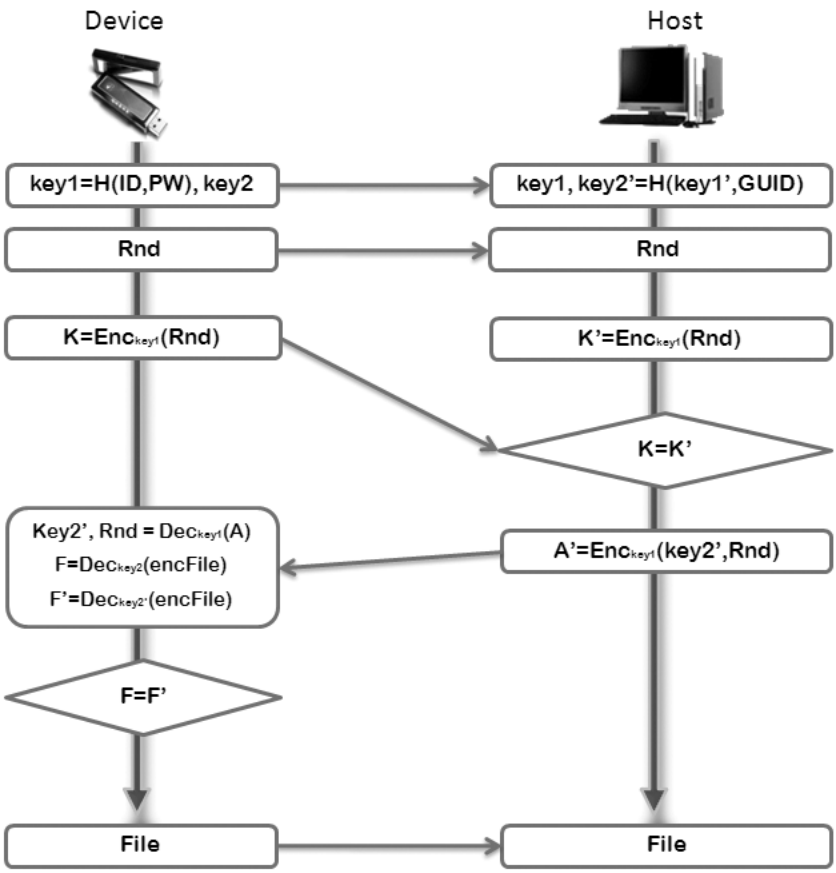

Figure 5: Authentication process for type C

Vulnerability	Product A	Product B	Product C
Password found in the USB packet	O	O	X
Authentication key found in the USB packet	X	O	X
Encryption key found in the USB packet	O	O	X
Password found in the host memory	O	O	O
Authentication key found in the host memory	X	O	O
Encryption key found in the host memory	O	O	O
Password found in the USB memory	O	O	X
Authentication key found in the USB memory	X	O	O
Encryption key found in the USB memory	O	O	O
Authentication process compromised	O	O	O
Decryption process compromised	O	O	O
Standard requests possible without authentication	O	O	X

Figure 6: Considerable vulnerabilities to the authentication of the USB memory

- *Whether the authentication process is perverted through the reverse engineering*: Decision or branch operations are simply compromised and register values or stack contents are easily distorted by malicious users who arms with disassemblers, debuggers or many other hooking utilities.
- *Whether it is possible to attack the USB memory through the standard USB request commands*: Usual hardware chip in the USB memory supports standard requests for mass storage class devices. For many commercial products that implement their authentication process at the kernel mode driver level without any support from the memory itself, the process can be easily bypassed and the standard requests are issued by a kernel mode malicious code to collect the internal information of the memory.

Secure USB memory requires USB interface of course. USB interface is intrinsically unsecure because the USB data are transferred through USB host controller's communication area (HCCA) which is mapped into the primary memory of the host computer. The primary memory of the computer is basically non-volatile across processes, it is impossible to protect USB packets from sniffing. It is not enough to encrypt the authentic information when they are transferred in the sniff-unsafe environment because the affirmative response message after a successful authentication can be replayed by an attacker. Because of the reason, status of the transaction sequences should not be recognized as well as the meaning of the messages transferred. It is notable that an acceptable solution to this problem is to use onetime-pass key scheme, which raises too much overhead. Instead, it would be a reasonable scheme to minimize transactions or oppositely maximize them by mixing lots of random noisy messages. In many cases related to the vulnerabilities, the weakness is due to the failure to protect software from reverse engineering. Even though the authentication protocol is securely designed, everything is highly encrypted and messages and resources are all confidential, it is impossible to keep the memory securely if the managing software loses control on the operations that are related to comparison, branch or dispatch. To escape from this problem, the software needs to have a sophisticatedly designed code level obfuscation scheme even in virtue of heavy additional workload. An alternative approach to the obfuscation is to move the decision functions in the host software away into the USB memory itself as pretty as possible. It partially means that the authentication process requires to be executed by the remote device itself, which also means that it is desirable the device authenticates the host or user instead of vice versa.

5 Conclusion

This paper describes different authentication protocols for three different types of commercial secure USB memories. Even though these products tried to make themselves secure, the dedicated authentication protocols and the implemented software are not complete and many vulnerabilities can be utilized at some possible attacks. Especially, device and user authentication protocol is more important than data encryption for the devices including secure USB memories. However, manufacturers try to give the security focus onto data encryption rather than authentication. The considerable vulnerabilities shown in figure 6 should be evaluated to make security considerations balanced and fully investigated on both sides.

References

[1] Hanjae Jeong, ”Vulnerability Analysis of Secure USB Flash Drives,” Journal of the KIISC, vol. 17, No. 6, pp.99-118, December 2007

[2] Kangbin Yim, ”A fix to the HCI specification to evade ID and password exposure by USB sniff,” Proceedings of APIC-IST 2008, pp.191-194, December 2008

[3] Kangbin Yim, ”A new noise mingling approach to protect the authentication password,” IEEE, proceedings of CISIS 2010 Conference, pp. 839-842, February 2010

[4] O’Gorman L., ”Comparing Passwords, Tokens, and Biometrics for User Authentication,” Proceedings of the IEEE, Vol. 91, No.12, December 2003

[5] Amit Vasudevan, Bryan Parno, Ning Qu, Adrian Perrig, ”Lockdown: A Safe and Practical Environment for Security Applications,” Technical Report CMU-CyLab-09-011, CyLab Carnegie Mellon, July 2009

[6] Taeyoung Jung, Kangbin Yim, ”Countermeasures to the Vulnerability of the Keyboard Hardware,” Journal of the Korea Information Security and Cryptology, Vol.18, No.4, pp.187-194, August 2008

[7] Kingpin, ”Attacks on and Countermeasures for USB Hardware Token Devices,” proceedings of the Fifth Nordic Workshop on Secure IT Systems Encouraging Co-operation, pp.35-57, October, 2000

[8] Kyungroul Lee, Kwangjin Bae, Kangbin Yim, ”Hardware Approach to Solving Password Exposure Problem through Keyboard Sniff,” ACADEMIC SCIENCE RESEARCH, proceedings of WASET2009, pp.23-25, October 2009

[9] Kwangjin Bae, Kangbin Yim, ”Analysis of an Intrinsic Vulnerability on Keyboard Security,” Journal of the KISC, Vol.18, No.3, pp.89-95, June 2008

[10] Kangbin Yim, ”Keyboard Security,” Workshop on Ubiquitous Information Security, May 2008

Kyungroul Lee received his B.S. from Soonchunhyang University, Asan, Korea in 2008 and he is currently towards Master’s degree. His research interests include vulnerability analysis, kernel mode root kit, obfuscation, systems security, access control and insider threats.

Hyeungjun Yeuk received the B.S. from Dongyang University, Yeongju, Korea in 2010. He is currently a graduate student of Soonchunhyang University. His research interests include vulnerability analysis, kernel mode hooking and secure hardware.

Youngtae Choi received the B.S. from Soonchunhyang University, Asan, Korea in 2010 and he is currently a graduate student of Soonchunhyang University. His research interests include vulnerability analysis, systems security, and insider threats.

Sitha Pho received the B.S. from Handong University, Pohang, Korea in 2009 and he started his Master course at Soonchunhyang University in 2010. His research interests include vulnerability analysis and code obfuscation.

Ilsun You received his M.S. and Ph.D. degrees in Computer Science from Dankook University, Seoul, Korea in 1997 and 2002, respectively. Since March 2005, he has been an Assistant Professor in the School of InformationScience at the Korean Bible University, South Korea. Prof. You served or is currently serving on the organizing or program committees of international conferences and workshops including CISIS'10-11, IMIS'07-11, MIST'09-11, MobiWorld'08-11, BWCCA'10-11 and so forth. Also, he has served as a guest editor for more than 10 international journals. He is on the editorial boards of International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), Computing and Informatics (CAI), and Journal of Korean Society for Internet Information (KSII). His main research interests include mobile Internet security and formal security verification.

Kangbin Yim received his B.S. M.S. and Ph.D. from Ajou University, Suwon, Korea in 1992, 1994 and 2001, respectively. He is currently an associate professor as he has joined Dept. of Information Security Engineering, Soonchunhyang University since 2003. His research interests include vulnerability analysis, code obfuscation, metamorphic malware, insider threats, access control and secure hardware and systems security.