# Prototype Design of Self-securing Portable Storage Device

ZHU Li-jue
School of Computer Science
National University of Defense Technology
Changsha, China
E-mail: hoyozlj@163.com

ZHAO Wen-tao
School of Computer Science
National University of Defense Technology
Changsha, China
E-mail: zhao_bruce@sina.com

WU Hui-jun
School of Computer Science
National University of Defense Technology
Changsha, China
E-mail: brightwhj@gmail.com nick_season@163.com

LIU Yong
School of Computer Science
National University of Defense Technology
Changsha, China
E-mail: nick_season@163.com

HU Zhao-ming
School of Computer Science
National University of Defense Technology
Changsha, China
E-mail: hulahzm@gmail.com

***Abstract*—Embedding the security mechanism into the device is the core of self-securing, on purpose to provide the device with the ability of self-defense. Currently, security of portable storage device is commonly depends on the host, which makes the data easy to be attacked or stolen. Researches on self-securing mechanism on portable device are few. In order to improve the safety of those devices, we proposed the prototype design of self-securing portable storage device based on the framework of ARM+Linux. Combined with access control, intrusion detection based on storage and data encryption/decryption, it can protect data availability, integrity and confidentiality effectively. Since we have expanded researches on self-securing mechanism of portable storage devices, this prototype design is promising to be one kind of trend of intelligent storage.**

***Keywords-self-securing; storage security; access control; intrusion detection***

## I. INTRODUCTION

With the role of electronic information being more and more important in today's human social life, the security of storage have become the focus issue. Traditional security mechanisms, represented by data encryption and access control, are mostly established on the host, therefore, their realizing of security are forced to depend on how security the host being. As a matter of fact, system of host is not indestructible, once the security of operating system or protective mechanism on the host is broken through, there would be no sense of storage security. Self-securing, which exactly based on no confidence of the host at all, embeds the security functions into the device on purpose to enhance its ability to withstand attack. In recent years, self-securing storage has raised wide focus, and self-securing mechanism have promisingly recognized the typical application of intelligent storage. However, technical problems of implementation still need further research, and studies and researches on self-securing realization on portable storage device are few.

Currently, portable storage devices, especially USB ones, have been widely used in various information systems as the advantage of mass storage, tiny size and fast speed. In important institutions as government and enterprises, USB devices are usually storing and transferring sensitive data and information. Those, once leaked or missed, will cost a lot for people, business and even the whole nation. Meanwhile, portable storage devices are marching ahead to diversification. Electronic products, such as cell phones, digital cameras and tablets, have indeed become significant storage devices via connecting with the computer. These products usually carry vital information of users, and cases of information stolen through cell phones are really common.

Differ from hard disks, portable storage devices face kinds of environments much more than one computer. Since the environment, the host, is not guaranteed safe, so requirement for equipment safety is higher. That proves building a self-securing system onto the device is urgently necessary. In this paper, prototype design of self-securing portable storage device based on architecture of ARM+Linux is rendered, with vital functional modules realized and performance tested.

## II. PREVIOUS STUDIES

Self-securing technology can provide the ability of self-defense to the device, whose core is to embed the security mechanism into the storage device. With this method, safety of the device can no more depend on the host, which can largely enhance the storage security.

Researches on self-securing are not common nationally or internationally, existed research results are mostly focused on intelligence disk and self-securing disk. Domestic researches have studied about the self-securing model

Published by Atlantis Press, Paris, France.

implementation, S4, and the typical structure of self-securing mechanism. Gu Dawu and Zeng Mengqi from Shanghai Jiaotong University have designed the solution of self-securing disk. Chen Yunliang from China University of Geosciences has worked fruitfully on intrusion detection based on storage.

Overseas researches, illustrated by Carnegie Mellon University and University of Pennsylvania, have done some in-depth study on Self-securing[6] and Autonomously Secure Disk[5]. Research results from Parallel Data Lab of Carnegie Mellon University are mostly outstanding. They have delivered numerous papers on self-securing and active disks (as reference [10~19]). Also the self-securing origin implementation, S4 was implemented by PDL.

However, researches focus on portable storage device is relatively few.

## III. PROTOTYPE DESIGN OF SELF-SECURING PORTABLE DEVICE

Self-securing portable storage device need to response to R/W requests from the host and handle the data going to store. Therefore, besides the memory chip, software and hardware system of the device itself is needed to support the self-securing mechanism.

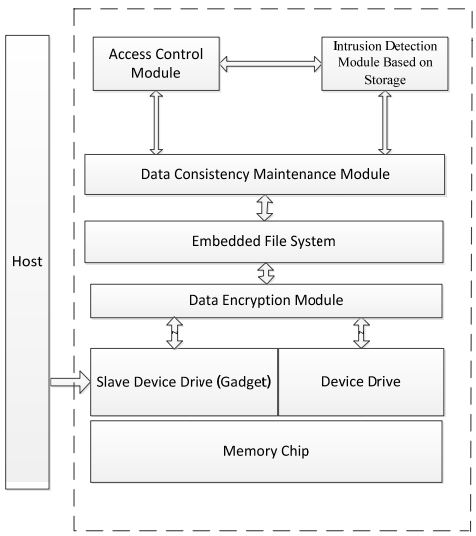


Figure1. Frame of self-securing portable storage device

As shown in Figure1, the underlying framework is memory chip, which connect with the upper-layer file system through Gadget and device drive. Data encryption module between them can defense the physical attack from cheating. Three modules on the top is application ones.

*Memory Chip*

The memory chip is SD card of device connected and the device read/write data from/onto the chip through device drive.

*Slave Device Drive (Gadget)*

In Linux, USB devices can be treated as host as well as slave, which two drives corresponded. In Linux kernel, slave device drive fall into 3 layers: USB device control drive, Gadget API and Gadget drive. The device control drive visits the hardware directly to control the underlying communication and provided call-back functions to upper layer. Gadget drive calls API to realize USB device functions. Basically, with slave device drive, the memory chip can manage and lay out related applications based on Linux.

*Device Drive*

Device drive is responsible for the interaction between device and storage card.

*Data Encryption Module*

Encryption is the vital solution to data confidentiality. We encrypt data stored in the device to avoid no-right users acquire data directly from the memory chip. Common encryptions include application layer encryption, operating system encryption and full disk encryption. Considering the limitation of system resources, we accept application layer encryption. Taking public-key cryptosystem and public/private keys separate way can bring more effective protection to the private key and system when attackers try to break the code. Further, choice of encryption algorithms and modes holds key effect on safety. Self-securing device's real-time performance considered, we accept RC6-XTS-CBC+Elphant diffuser after studied multiple candidate AES algorithms.

*Intrusion Detection Module Based on Storage*

This module is a hosting agent of device actually. Same as network storage system, self-securing storage device as well mount the memory chip on system via USB Gadget slave device drive, and it can provide the overall package of self-securing mechanism and the memory chip to users. The intrusion detection module monitors the requests send from the host side and model the illegal behaviors like an autorun associating executable files. Take virus detection based on autorun files as an example, when the device finds an autorun file, it searches all the storage space, if a related executable file exits, Clam Amtivirus (a Linux open source anti-virus software) can scan and detect if it is a virus.

*Access Control Module*

Besides encryption, self-securing device should provide good usability to users. Access control module authenticates users by key file or certification instead of host-side software. The ACM verify the key file or certification to explode, views, different memory content to users. Since the difference between operating systems on the host and the device, R/W attributes of files will be controlled by the file system. Moreover, the user views have time limits, once time is out, the memory content will de encrypted again.

Published by Atlantis Press, Paris, France.

*Data Consistency Maintenance Module*

Self-securing device should owe a recovery mechanism to prevent data corruption. The device records every request from users in the log, which defined permissions on the file system layer, and stores history files in a specific storage space. After data damaged, the administrator can analyze the log and recover the history edition. This module cooperates with the intrusion detection module to improve data regeneration of real-time.

## IV. SYSTEM PROCESS AND SECURITY

According to the prototype design, the system process is shown in Figure2.

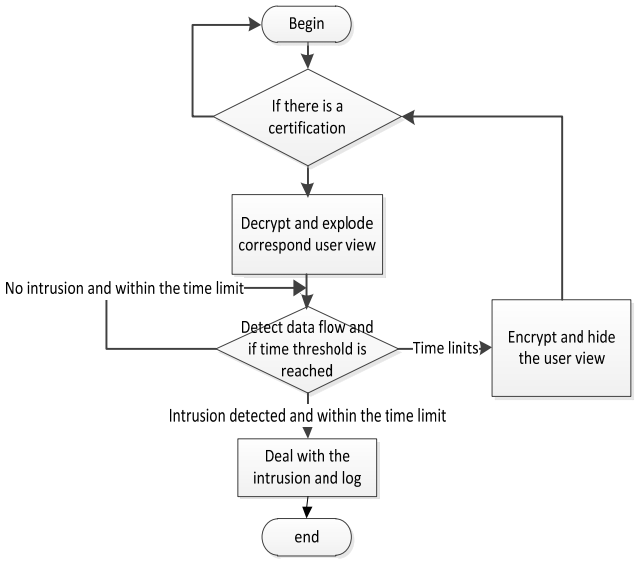


Figure2. System process

The device keep scanning the memory to detect the certification, once it written, the device will decrypt the data encryption key $Key$ with private key $SK_A$ stored in the certification, thus $Key = SK_A[PK_A[Key]]$, where $PK_A$ means the public key. The data encryption key is encrypted with the public key mainly to prevent from physical attacks and memory content analysis. After achieving the encrypted key, the device will decrypt the data and explode correspond user view. Once the time threshold is reached, data will be encrypted again and user view will hide. The device detects data flows, if there exits intrusion, it will deal with it and log.

## V. FURTHER WORK

In this paper, we proposed a prototype of self-securing portable storage device based on the framework of ARM_Linux. It includes memory chip, slave device drive, device drive, data encryption module, intrusion detection module based on storage, access control module and data consistency maintenance module. This prototype design can ensure the data availability, integrity and confidentiality effectively.

Further work we mainly need to do is to improve the scanning algorithm of detecting malicious executable programs and viruses, promote efficiency and reduce time consumption of intrusion detection. As well, enhance the access control module with PKI public key mechanism to make it more usable and natural.

## REFERENCES

[1] Christopher Lum, Jiri Schindler, Gregory R.Ganger, David F.Nagle, and Erik Riedel. Towards higher disk head utilization: Extracting \free" bandwidth from busy disk drives. Symposium on Operating Systems Design and Implementation (San Deigo, CA, 23-25 October 2000). ACM, October 2000.

[2] Josh MacDonald. File system support for delta compression. Masters thesis. Department of Electrical Engineering and Computer Science, University of California at Berkeley, 2000.

[3] John D. Strunk, Garth R. Goodson, Adam G. Pennington, Craig A.N. Soules, Gregory R. Ganger. Intrusion Detection, Diagnosis, and Recovery with Self-Securing Storage. CMU-CS-02-140. May 2002.

[4] Craig A. N. Soules, Garth R. Goodson, John D. Strunk, and Gregory R. Ganger. Metadata efficiency in a comprehensive versioning file system. Technical report CMU–CS–02–145. Carnegie Mellon University, 2002.

[5] Butler K.McLanghlin S, McDaniel P. Non-volatile Memory and Disks: Avenues for Policy Architectures[C]. Proc of the 1st Computer Security Architecture Workshop. Alexandria, VA, USA: IEEE Computer Society, 2007: 51-57.

[6] Strunk J D. Self-securing Storage: Protecting Data in Compromised Systems[C]. Proc of the 4th Symposium on Operating Systems Design and Implementation[S.1.]: IEEE Computer Society 2000: 195-209.

[7] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. on Neural Networks*, vol. 4, pp. 570-578, July 1993.

[8] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility," IEEE Trans. Electron Devices, vol. ED-11, pp. 34-39, Jan. 1959.

[9] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, and M. Miller, "Rotation, scale, and translation resilient public watermarking for images," IEEE Trans. Image Process., vol. 10, no. 5, pp. 767-782, May 2001.

[10] WYLIE J J, BIGRIGGMW, STRUNK JD, et al. Survivable information storage systems[ J].IEEE Computer Society, 2000,33(8): 61-68.

[11] WYLIE J J, BAKKALOGLUM, PANDURANGAN V, et al. Selecting the right data distribution scheme for a survivable storage system,CMU-CS-01-120 [ R ]. Pittsburgh: Carnegie Mellon University,2001: 1-10.

[12] PENNINGTON A, STRUNK J, GRIFFIN J, et al. Storage-based intrusion detection: watching storage activity for suspicious behavior, CMU-CS-02-179[R]. Pittsburgh: Carnegie Mellon University, 2002:1-11.

[13] STRUNK JD, GOODSONG R, PENNINGTONAG, et al. Intrusion detection, diagnosis, and recovery with self-securing storage, CMU-CS-02-140[R]. Pittsburgh: Carnegie Mellon University, 2002: 1-9.

[14] RIEDEL E, FALOUTSOS C, GIBSON G A, et al. Active disks for large-scale data processing[J].Computer, 2001,34(6): 68-74.

[15] RIEDEL E, FALOUTSOS C, NAGLE D. Active disk architecture for databases, CMU-CS-00-145 [R]. Pittsburgh: Carnegie Mellon University, 2002: 1-12.

[16] RUEDEL E, FALOUTSOS C, GANGER G R,etal. Data mining on an OLTP system (nearly) for free[ J].ACM S IGMOD Record,2000,29(2): 13-21.

Published by Atlantis Press, Paris, France.

[17] RIEDEL E, GIBSON G. Active disks-remote execution for network-attached storage, CMU-CS-99-177[R]. Pittsburgh: Carnegie Mellon University, 1999: 1-11.

[18] RIEDEL E, GIBSON G. Active storage for large-scale data mining and multimedia[C] //Proc of the 24th International Conference on Very Large Databases. San Francisco: Morgan Kaufmann Publishers,1998: 62-73.

[19] RIEDEL E, GIBSON G. Active disks: remote execution for network-attached storage, CMU-CS-97-198[R]. Pittsburgh: Carnegie Mellon University, 1997: 1-9.

Published by Atlantis Press, Paris, France.