Data Protection Technology in Classified Networks

CHEN Xun, HAN Zhen, and LIU Ji-qiang

(School of Computer and Information Technology, Beijing Jiaotong University Haidian Beijing 100044)

Abstract Management of cryptography keys for data encapsulation in classified network has been troubling people. The aim of this article is to present a new mechanism for protecting classified data by using trusted cryptography module. The solution contains both runtime data protection and static file data protection. The key technique is using the feature of cryptography keys hidden technology of trusted cryptography module. The result of a simple performance test of the trusted cryptography module is provided while the solution for its insufficiency is also presented. By using the trusted cryptography module, an implementation with experiment result of a network control is presented.

Key words classified network; data encapsulation; network security; trusted cryptography module

涉密网络中的数据保护技术

陈 勋，韩 臻，刘吉强

(北京交通大学计算机与信息技术学院 北京 海淀区 100044)

【摘要】提出了通过使用可信密码模块保护涉密数据的一种新的机制，该解决方案包括运行时的数据保护与静态文件数据保护，使用的关键技术是可信密码模块的密钥保护特性。给出了可信密码模块加解密的性能测试，以及弥补性能不足的解决方案。通过使用可信密码模块的数据封装技术，提出了一个网络控制的实现方案，给出了网络控制的实验测试结果。

关 键 词 涉密网络; 数据封装; 网络安全; 可信密码模块
中图分类号 TP309 文献标志码 A doi:10.3969/j.issn.1001-0548.2013.01.029

1 Introduction

In these years, information leak often happens in enterprises, organizations, governments units, and etc. People set up classified network in their inner network structure, in which data can be plaintext for convenience when transmitting through network. Under this situation, staff can find ways to access internet intentionally or unintentionally. Classified data is very dangerous when exposed to unsecured environment[1-2]. When data come to public area, such as VPN where data have to be transmitted out of classified network, data must be encrypted as ciphertext before transmission[3]. Even though, cryptography keys lost or stolen still make classified data in danger. If cryptography keys are managed by machine not by human, the situation can be less complicated.

In this article, we provide a solution to help preventing information leak by using Trusted Cryptography Module, TCM for short. Data are encapsulated by the chip in the computer. Data will be automatically encrypted or decrypted using TCM when needed. Only authorized applications can use the encapsulated data. Because cryptography keys are always in TCM, nobody can obtain them. If the data are stolen by hackers or staffs, what they get is just the cypher text.

In this article, An attack model is given and the protection method is provided. The corresponding implementation is given.

Received date：2012 – 08 – 23； Revised date：2012 – 11 – 15
收稿日期：2012 – 08 – 23； 修回日期：2012 – 11 – 15
Foundation item：Supported by the National Natural Science Foundation of China under Grant(60973112)；Program for New Century Excellent Talents in University(NCET-11-0565)；Fundamental Research Funds for the Central Universities under Grant(2011JBM221)
基金项目：国家自然科学基金(60973112)；新世纪优秀人才支持计划(NCET-11-0565)；中央高校基本科研业务费专项资金(2011JBM221)
Biography：CHEN Xun was born in 1984, and his research interests include trust computing.
作者简介：陈勋(1984 –)，男，博士生，主要从事可信计算方面的研究.

2 Preliminaries

2.1 Trusted Cryptography Module

Trusted Cryptography Module is a hardware module of the trusted computing platform. It provides cryptography features and protected storage space. With the basic technology of cryptography, TCM can be used to achieve platform integrity, identification credibility, and data security. Instead of software method, TCM can help to protect critical data, such as cryptography keys, in the hardware level, which enforces cryptography calculation inside the chip without reveal of the keys.

The Trusted Computing Group is an international industry standards group which using Trusted Platform Module, TPM for short, to achieve trusted computing. Several Chinese IT industry corporations adopt Chinese developed cryptography algorithm and engine to create the TCM. TCM, which is quite similar to TPM, can be used to build a more secure and reliable system environment.

2.2 Winsock LSP

Layered Service Provider is a feature of Microsoft Windows Winsock Service Provider Interface. A dll library can be built according to Winsock LSP interface standard to achieve modifying all the inbound and outbound data through the network, which means all the information of network request can be easily obtained, including network address and content. This can be used for a wide range of practical usage, such as parent control, Web content filtering, and etc. This article uses the feature to achieve classified data protection through the network access.

3 Attack Model

Here is the possible attack model. Alice and Bob are coordinated attackers in the inside and outside of the classified network. Alice has classified data in her computer and wants to give them to Bob. Below are their potential methods to obtain Alice's classified data. We assume that they do not use paper and a pen or a camera to copy the data.

1) Alice copies the classified data file directly through a removable storage and gives it to Bob.

2) Alice plants malware into her own computer. The malware sends the classified information to Bob by communication tools when Alice uses the legal software to deal with the classified data.

3) Alice writes some illegal websites in the policy file for allowing uploading classified data to Bob.

Facing the attacks mentioned above, we devote to protect the classified information in this paper.

4 Data Encapsulation Model

4.1 Cryptography Model

This article has given a model for encapsulating classified data, as shown in Figure 1. All the data are encrypted using cryptographic keys which are stored in TCM. Authorized applications, which often are specified, have the exclusive universally unique identifier, UUID in abbreviate, to load the keys in TCM for encryption and decryption. The data, which are decrypted in TCM, will be used by authorized applications. Because unauthorized applications do not have the appropriate UUID to load the correct keys, they can not get the plaintext of the data.

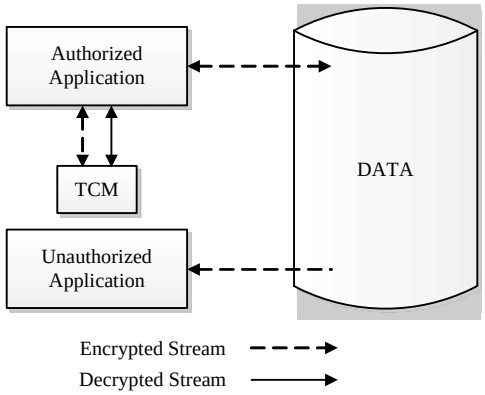

Figure 1 Data cryptography model

Applications use the opposite way to encrypt data. After plaintext of the data are loaded into TCM, TCM outputs encrypted data to the applications. Thus applications can save the data to database.

The cryptography keys are always in TCM and nobody can get the keys. The only one who can use the keys for cryptography is TCM itself. Thus the encrypted data is safely stored.

4.2 Encapsulation Model

Although authorized applications can obtain the decrypted data, the data are still unsafe. Without some protection mechanism, the decrypted data stored in the

memory space are exposed to malicious codes[4-6]. Making the calculation environment safe is a must for protecting classified data. In this paper, we provide a validation mechanism to achieve a trusted platform[7-8].

DEFINITION 1 We give P for a set of properties defining requirements of the trusted platform, defined as PF_T. Properties of the calculation platform, defined as PF_C, are $p_1, p_2, \cdots, p_n$.

THEOREM 1 PF_C is PF_T iff properties $p_1, p_2, \cdots, p_n$ satisfying $\{p_i \in P \mid 1 \leqslant i \leqslant n\}$.

If every property of the calculation environment meets the requirements of the trusted platform, we believe that the calculation environment becomes a trusted platform. Only in the trusted platform does the TCM Core Service[9-10] start to function. Thus, we need to define properties of the trusted platform, such as process white list, anti-virus service, and etc., which will be discussed in the follows of the article. Under this mechanism, we believe the decrypted data is safely kept in the memory space.

5 Implementation

5.1 Data Cryptography Implementation

The speed of cryptography calculation of the TCM chips is low. Figure 2 has shown the speed of the TCM when we use TCM Service Module Interface for cryptography calculation. The maximum speed is about 750 bytes per second. So we cannot use TCM to do large data cryptography.

From Figure 2, we can notice that the data size of every peak point is the multiple of 512, which means the maximum data size is 512 bytes when TCM Service Module transmits data.

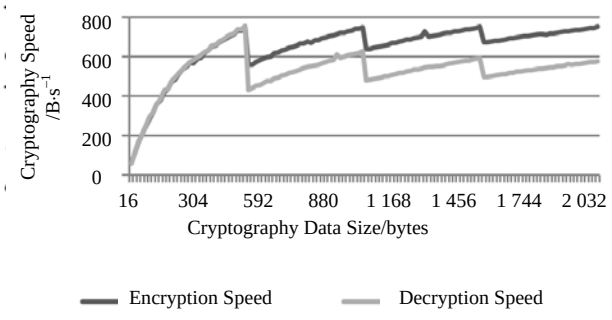

Figure 2 Cryptography speed of TCM process scheme

Since TCM is not designed for large data cryptography, we use it for cryptography keys management instead. The cryptography calculation is taken place in CPU for the large size classified data. The keys used in data cryptography is encrypted or decrypted in TCM. With the technology of process white list control[12] and process isolation mechanism[13-15], keys for cryptography of classified data can be well protected.

We also give the test result of cryptography speed in CPU process scheme, as shown in Figure 3. The maximum speed is about 2.2 MB per second, which is much higher than the TCM process scheme.

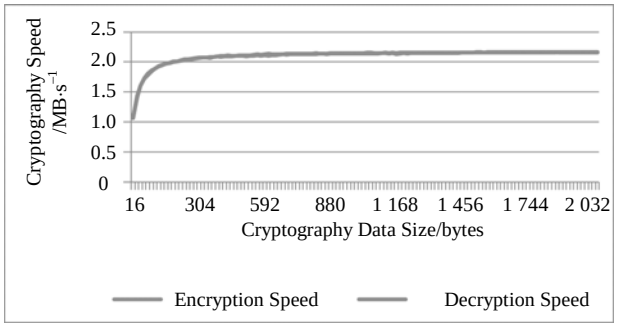

Figure 3 Cryptography speed of CPU process scheme

In our implementation, we use a USB key for additional protection. Every user has a USB key. Data are encapsulated with the USB key and TCM. Every cryptography key in TCM has a UUID, which is stored in the USB key. The cryptography key used in the CPU process scheme is encrypted by TCM and stored in the USB key. When the UUID and the encrypted cryptography key are loaded, password should be provided. When we need to encrypt or decrypt classified data, we first load the UUID to load the cryptography key in TCM. With the key in TCM, we decrypt the cryptography key stored in the USB key. Then we can use the decrypted cryptography key to do any data encapsulation. The plaintext and ciphertext shown in Figure 4 are the encapsulation source and result in our implementation.

TCM is the key for data encapsulation. In our implementation, our client agent validates the calculation environment before the TCM starts to work. If the validation fails, TCM Core Service will be stopped. Data cannot be used by any application. The client agent validates whether the process whitelist module and process isolation module are working, whether anti-virus system is on, whether anti-virus

database is up to date, whether firewall module is started, and etc. These are properties for the requirement to achieve trusted platform.

Figure 4 Classified data before and after encapsulation

In order to protect the data, one more property is necessary for the trusted platform. Think about that there is an authorized application which uses network to send the decrypted data to internet. Thus in our implementation, we provide a network control module design, which will be discussed later in this article. If the network control module is not working, the TCS will be stopped to protect the classified data. Our client agent, including the network control module, is started early as the operating system starts to ensure the data protection.

5.2 Network Control Module

In order to control the network access, some operating systems provide interfaces to developers for additional functions[16]. We adopt Windows Layer Service Provider, short for WinLSP, to achieve network control for convenience.

Our network control module consists of 3 parts as follows, Whitelist Loader, Network Request Filter and Server Connection Monitor, as described in Figure 5.

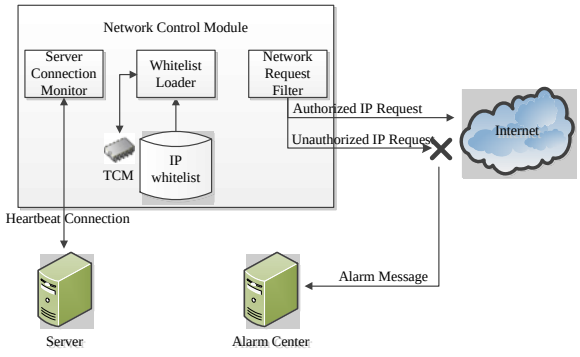

Figure 5 Structure of network control module

WinLSP loads our library which functions as network IP whitelist module. The module will load the IP whitelist at startup. Administrator customizes the whitelist in which applications obey the rules which network address can access. If one network address is not in the IP whitelist, WinLSP will stop this network request. Thus if the application is going to send the decrypted classified data to unauthorized address in the internet, the request will be denied. Also our system will send the alarm message describing the activity to the alarm center.

The client agent keeps the connection with the server from the beginning of operating system starts. Once the connection is closed, which means the computer may be moved to other network environment to avoid the server monitoring, then the network control module will stop all the network requests in order to keep the classified data safe.

To maintain the integrity and the confidentiality of the IP whitelist, whitelist is protected using our data encapsulation mechanism described in section 4. The only difference is that the whitelist data can be encapsulated directly by TCM because the data size is small.

Figure 6 shows the result when a user want to access the website whose IP address is not listed in the IP white list policy file. The network request will be blocked and the result will be displayed to user. If the website is allowed be accessed in classified network, administrators can add it into the policy file.

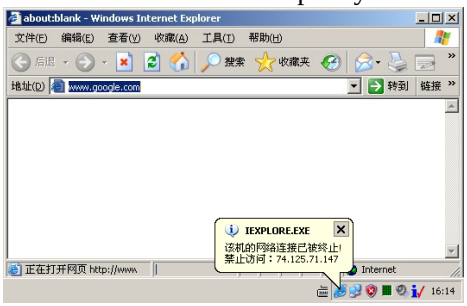

Figure 6 Websites not listed in white list are blocked

We have also tested the network control module for the success rate of the blocking function through over 50 sample software. The result is 100% of the applications are under control. We also test for data transmit efficiency. The average transmission speed is reduced by 0.02% after use of network control module, which can be ignored.

Overall, by using network control module data can be well protected without the abuse of network

access. It can avoid the harm even if staffs attempt to betray classified data from the inside of intranet.

6 Conclusion

We have shown how to use TCM to build a mechanism for protecting classified data. By having a simple test of the performance of the TCM, we use CPU for large size data cryptography while using TCM for cryptography keys protection. We also give a design of network control module for protecting decrypted classified data from stealing by authorized applications, which is a good reference for companies to solve the problem of staff stealing data by network.

Data protection using TCM also prompts interesting new research questions, such as level based data protection, access control using TCM, process isolation base on TCM, and so forth.

7 Epilogue

The solution given by this article can help to prevent from the attack method mentioned in the attack model of section 3. When Alice copies the development document directly from her computer, which is classified data of the company, Bob can not see the plaintext of the file because both Bob and Alice do not have the cryptography key. Alice refuses to give up. She tries to send the classified data as the attachment of the email to Bob. But the administrator of the company does not give the permission for staffs to access any email service provider. Then Alice finds out the location of the policy file and plans to insert the IP address of the email website. At last she feels great disappointment because she cannot understand the encrypted policy content. Once she inserts something to the policy file forcibly, the network control module will detect the changes of the policy file. All the network connections of Alice's computer will stop and the administrator will receive the alarm message. Thus, the data can be well protected in the classified network.

Reference

[1] LIANG Ying-bin, POOR H V, SHAMAI S. Information theoretic security[J]. Foundations and Trends in Communications and Information Theory, 2009, 5(4-5): 355-580.

[2] ZHAO Yong, LIU Ji-qiang, HAN Zhen, et al. The application of information leakage defendable model in enterprise intranet security[J]. Journal of Computer Research and Development, 2007, 44(5): 761-767.

[3] RAHIMI S, ZARGHAM M. Analysis of the security of VPN configurations in industrial control environments[J]. International Journal of Critical Infrastructure Protection, 2012, 5(1): 3-13.

[4] DAVI L, SADEGHI A R, WINANDY M. Dynamic integrity measurement and attestation: towards defense against return-oriented programming attacks[C]//Proceedings of the 16th ACM Conference on Computer and Communications Security. New York, USA: ACM, 2009: 49-54.

[5] CHECKOWAY S, DAVI L, DMITRIENKO A, et al. Return-oriented programming without returns[C]// Proceedings of the 17th ACM Conference on Computer and Communications Security. New York, USA: ACM, 2010.

[6] BLETSCH T, JIANG Xu-xian, FREEH V W, et al. Jump-oriented programming: a new class of code-reuse attack[C]//Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security. New York, USA: ACM, 2011: 30-40.

[7] GRAWROCK D. Dynamics of a trusted platform: a building block approach[M]. Santa Clara, CA, USA: Intel Press, 2009.

[8] ZHANG Ting, JIANG Hui-ping, GUI Xin-kai, et al. Design principles for trusted platform modules protected with power analysis[C]//2012 Second International Conference on Intelligent System Design and Engineering Application. Los Alamitos, USA: IEEE Computer Society, 2012: 1409-1412.

[9] TOEGL R, WINKLER T, NAUMAN M, et al. Towards platform-independent trusted computing[C]//Proceedings of the 2009 ACM Workshop on Scalable Trusted Computing. New York, USA: ACM, 2009: 61-66.

[10] Trusted Computing Group. TCG software stack specification[EB/OL]. [2007-11-09] http://www. Trusted-computinggroup. org/resources/tcg_software_stack_ tss_ specification.

[11] Trusted Computing Group. TCG TPM specification [EB/OL]. [2011-03-20] http://www.trustedcomputinggroup. org/resources/ tpm_main_specification.

[12] WU Yun-long, CUI Dong, ZHANG Qiang. A malicious software evaluation system based on behavior association [C]//International Conference on Optics, Photonics and Energy Engineering. New York, USA: IEEE, 2010: 258-260.

[13] YU Li, WEI Jiang, LI Lin, et al. Research on user permission isolation for multi-users service-oriented program[J]. Int'l J of Communications, Network and System Sciences, 2012, 5(S2): 105-110.

[14] GREY J A. Test executive with external process isolation for user code modules[P]. US7480826B2. Austin, Texas, USA: [s.n.], 2009.

[15] FOCKE M W, KNOKE J E, BARBIERI P A, et al. Trusted operating system with emulation and process isolation[P]. US7549165B2. Wilson Boulevard, Arlington, USA: [s.n.], 2009.

[16] RUSSINOVICH M E, SOLOMON D A, IONESCU A. Windows internals[M]. 6th edition. Louisville, KY, USA: Microsoft Press, 2012.

编 辑 黄 莘