A Design of Trusted Computing Supporting Software based on Security Function

LENG Jing[1, a], HE Fan[2*,b]

[1] Department of Information Technology, Hubei University of Police, Wuhan 430034, China
[2*] Corresponding author, School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China

Abstracts. Most of existing TCSS have defined the functional interface of software in accordance with TCG specification. However, there is no clear definition of the security functional requirements and security targets that TCSS needs to meet in TCG specification. An implementation scheme for TCSS was proposed, and a TCSS prototype on the basis of Common Criteria (CC), combining with TCSS security function division and practical application demands was implemented. The experimental results show that the proposed software prototype can support the interface call of the TPM 2.0 and the SM4 cipher algorithm of China.

1 Introduction

At present, network security threats and risks are becoming increasingly prominent all over the world. Academician Shen Changxiang proposed to realize network security by Trusted Computing (TC) which guarantees security protection during calculating operation so that the results can be consistent with the expected and the calculation can be controlled during the whole process without interference [1]. The research and development of Trusted Computing are in the ascendant. It is clearly pointed out in the 13th Five-year National Strategic Emerging Industrial Development Plan that we should actively promote the R&D and industrialization of Trusted Computing, data and network security, and other information technology products.

The basic idea of Trusted Computing is embodied in the design of the Trusted computing platform (TCP) as follows: First of all, create a roots of trust on the platform, i.e. TPM (Trusted Platform Module). And then create a chain of trust. Carry out measurement and authentication from the root of trust, and expanding level by level to hardware platform, trusted computing support software, operating system and application software to extend trust to the entire TCP and ensure the credibility of the entire platform [2].

TCSS (Trusted computing supporting software) is an essential part of TCP. It works as a communication bridge between application software and TPM. The application can use the security features provided by TPM through the interface call of TCSS [3].

Most TCSS at home and abroad are based on the trusted software stack specification of TCG (Trusted Computing Group). Representative products include the OSS released by IBM TrouSerS[4], Trusted Computing API for Java[5] developed by IAIK of Graz University of Technology, Austria and Client Security Solution8.3[6] of Lenovo.

Most of existing TCSS have defined the functional interface of software in accordance with TCG specification. However, there is no clear definition of the security functional requirements and security targets that TCSS needs to meet in TCG specification. This paper proposed an implementation scheme for TCSS, and a TCSS prototype on the basis of Common Criteria (CC), combining with TCSS security function division and practical application demands.

2 Background Knowledge

2.1 TCSS

TCSS defined in TCG specification is also known as TCG Software Stack (TSS) which is a software system provided by upper level applications to access to TPM interface. It is an indispensable part of TCP system [7].

TSS is a multi-level architecture which provides service and support to both local and remote trusted computing platforms. Fig. 1 presented TSS architecture which is specifically divided into TCG Service Provider (TSP), TCG Core Service (TCS) and TCG Device Driver Library (TDDL). Each layer has defined a normalized function interface. TSP is primarily used as a trusted proxy for local and remote applications, TCS is used to provide a collection of public services, while TDDL is responsible for interacting with TPM.

* Corresponding author: [a]*email:daleng0127@sina.com*, [b]*email: hefan@whut.edu.cn*

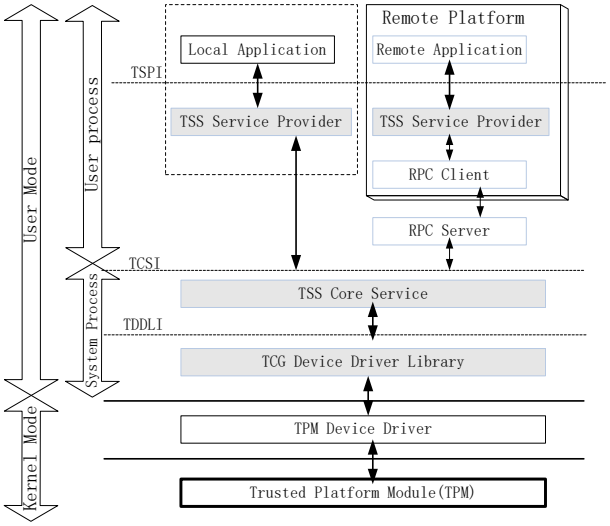

Figure 1. Architecture diagram of TCG software stack

2.2 Common Criteria

Common Criteria (CC) are internationally recognized information security evaluation criteria. It provides common standards for security function requirements and security assurance measures of information technology products[8]. CC can also guide the development and evaluation process of security-related IT products.

In CC system, PP (Protection Profile) is the basis for security assessment. It is the description of a set of security requirements of IT security products. It is a security requirements specification with higher abstract level. CC development committee also proposed a security assessment framework based on cooperative protection profile [9]. PP is conducive to improve the standardization of IT product development and evaluation process. TOE security requirements are to be determined through describing TOE (Target of Evaluation). TOE is generally defined as the software, firmware, and hardware realized by referring to relevant specification. TOE studied in the paper is TCSS.

Security requirements determined during development stage are of great significance to meet the user's security targets. It is not a mandatory requirement for CC to adopt a specific development method and model. CC guides the development process of TOE through gradually breaking down abstract security targets to ultimate realization. The basis of this process is to break down the security requirements to the TOE profile specification of security targets to make each low-level of detailed representatives possess design decomposition of more detailed design. The lowest abstraction expression indicates TOE realization. Breakdown of each level is the instantiation of a higher level [10]. Literature [11] expounded the basic assessment model of CC from the perspective of design decomposition, and presented a semi-formal method for security function requirements.

3 Security function analysis of TCSS

The definition of security functional requirements is used to describe the security features that a product should provide. CC formulated standard class description mode for security functional requirements to ensure the standardization of product development.

All TOE security functional requirements are based on considerations of TOE applications and environment. TCSS design requirements and standard export processes can be divided into multiple tiers. Firstly, determine the security environment of TCSS; secondly determine the security targets of TCSS; thirdly export TCSS security requirements, including functional requirements, assurance requirements and environmental requirements; and finally export security specifications. Complete the design and test of TCSS referring to security function specifications.

CC defined security functional requirements, according to the functional classes, families and component levels. A functional class contains one or more functional families. A functional family contains one or more components. And each component provides a set of security functional elements. TCSS standardized and defined by TCG is set as the research object. Security functions extracted from TCG Service Provider (TSP), TCG Core Service (TCS) and TCG Device Driver Library (TDDL) are function class; each security function is divided into different sub-functional modules corresponding to functional families; each module has different interface functions, namely corresponding components. And interdependent relationships exist between interface functions. Fig. 2 presented the hierarchy of security features of TCSS.

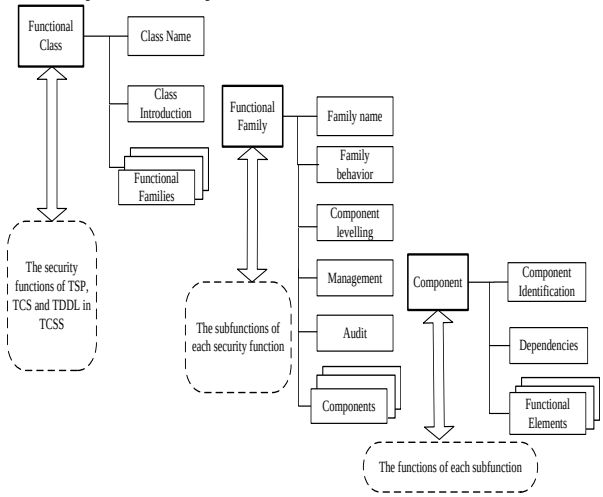

Figure2. Hierarchical structure diagram of TCSS security function

Referring to TCG trusted software stack specification as presented in 1.1, TCSS can be divided into TSP, TCS and TDDL. The modules of TSP layer include: HASH function, TPM management, context management, policy management, PCR management, secret key management, and data encryption management; modules of TCS layer include: encryption computation, key certificate management, context management, event management, and parameter block generation; modules

of TDDL layer include TDD interactive function, TPM information state setting and TDD command transferring function.

In order to meet the security requirements of the trusted computing platform and achieve the design functional objectives of TCSS, TCSS analysis and design are implemented according to different specifications. The paper extracted Trusted Computing standards and defined security functions of TCSS as shown in Fig. 3. Functional interfaces defined in diversified standards are different and the implementation mechanisms of TCSS will be different.But as long as TCSS can provide corresponding security features, design objectives can be achieved as well.

The relationship between security functions and related sub-functions of TSP layer of TCSS is analyzed as follows. TSP is divided to three security functions, i.e. integrity protection, trusted authentication, and cryptographic support. Security function of TSP can be decomposed to correlated subfunctions as well. Integrity protection includes HASH function and TPM management; trusted authentication includes context management, policy management, and PCR management; cryptographic support includes cryptogrammic key management and data encryption.

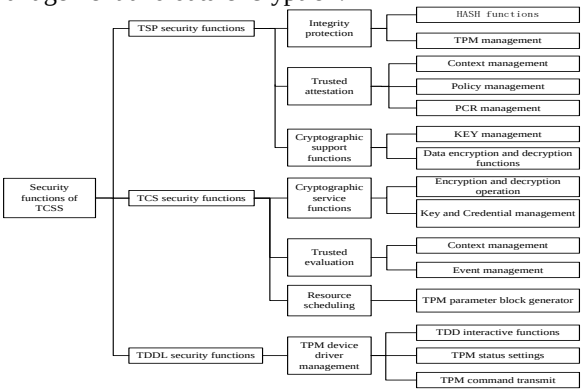

Figure3. Division diagram of TCSS security function

TCSS design adopts the method from top to bottom to identify functional classes and corresponding functional families, and progressively refine components related to design. TCSS design process is described below in detail by setting cryptographic support as an example.

Cryptographic support class (FCS class) in CC is composed of two functional families [12]: cryptogrammic key management (FCS_CKM) and cryptogrammic operation (FCS_COP). Referring to division of functional class, families, and components in CC, cryptographic support security function of TSP are analyzed. Subfunctions of cryptogrammic key management correspond to FCS_CKM functional families to solve cryptogrammic key management issues; data decryption function corresponds to FCS_COP functional families. It is related to service conditions of cryptogrammic key in operation. FCS_CKM functional families include four components: cryptogrammic key generation (FCS_CKM.1), cryptogrammic key distribution (FCS_CKM.2), cryptogrammic key using (FCS_CKM.3) and cryptogrammic key destroy (FCS_CKM.4); FCS_COP functional family's component is cryptogrammic operation (FCS_COP.1). The dependency relationship among components of cryptographic support functions is as shown in Fig. 4. It can be seen that in the process of completing cryptographic support security functions, cryptogrammic key generation component is executed first and cryptogrammic key destruction component is finally executed.

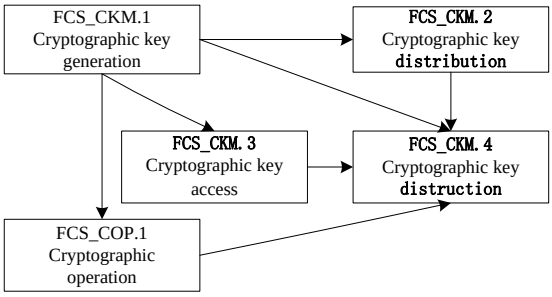

Figure 4. Dependency diagram of cryptographic support function

Cryptogrammic key generation, cryptogrammic key distribution, cryptogrammic key using, cryptogrammic key destroy, cryptogrammic operation and other components are designed with specific functional functions respectively. Table 1 presented division of TSP cryptographic support class.

Table 1. Division of cryptographic support class of TSP layer

Function class	Function family	Component	TSP layer corresponding to functions of the component	Function description
Cryptographic support (FCS class)	Cryptogrammic key management (FCS_CKM)	Cryptogrammic key generation FCS_CKM.1	Tspi_TPM2_ CreatePrimaryKey	Generate a primary cryptogrammic key
			Tspi_TPM2_CreateKey	Generate an asymmetric cryptogrammic-key
			Tspi_TPM2_CreateMigration Blob	Generate data packet of for the migration of cryptogrammic-key
		Cryptogrammic key distribution FCS_CKM.2	Tspi_TPM2_ConvertMigratio nBlob	Transfer Data packet for the migration of cryptogrammic-key
		Cryptogrammic key Using FCS_CKM.3	Tspi_TPM2_LoadKey	Load the generated cryptogrammic-key to TPM
			Tspi_TPM2_CertifyKey	Use a cryptogrammic key for authentication of the public key of the other cryptogrammic key
			Tspi_TPM2_WrapKey	Encapsulate a specific cryptogrammic key
			Tspi_TPM2_GetPubKey	Obtain the public key of asymmetric cryptographic-key loaded into TPM
		Cryptogrammic key destroy FCS_CKM.4	Tspi_TPM2_UnloadKey	Uninstall the specified cryptogrammic key in TPM
	Data encrypt and decrypt (FCS_COP)	Cryptogrammic operation FCS_COP.1	Tspi_TPM2_Bind	Bind a data packet
			Tspi_TPM2_Unbind	Unbind the data packet
			Tspi_TPM2_Seal	Encrypt a paragraph of plaintext with the public key of a cryptogrammic key
			Tspi_TPM2_Unseal	Decrypt the plaintext by using private key of the cryptogrammic key

4 Experiment

The development platform of the TCSS is an Intel minicomputer supporting TPM2.0. The operating system is Ubuntu10.04. And the compiler version is GCC3.3. 31 functional functions are designed and realized according to integrity protection, trusted authentication, cryptographic support and other security features. Functional function design of TCSS is a subset of software stack defined in TCG specification. The core security features are realized, and the call support for the Chinese standard encryption algorithm SM4 is added.

All functional requirements of the specification are expected to meet through performing functional test of all functional interfaces realized by TCSS. The workflow of TCSS prototype is introduced as follows by setting the test of SM4 algorithm invocation as an example. First of all, create a PrimaryKey. And then generate SM4 cryptogrammic key pair after authorization. Sign with a private key and verify the signature with a public key. Fig. 5 presented the whole test process. Where, Tspi_TPM2_ CreatePrimaryKey is to generate a primary cryptogrammic key. Tspi_TPM2_ CreateKey function can generate an asymmetric cryptogrammic key. Its function test interface is as shown in Fig. 6 and Fig. 7.

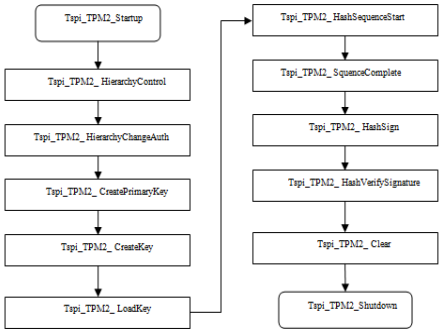

Figure 5. Process of SM4 key signing

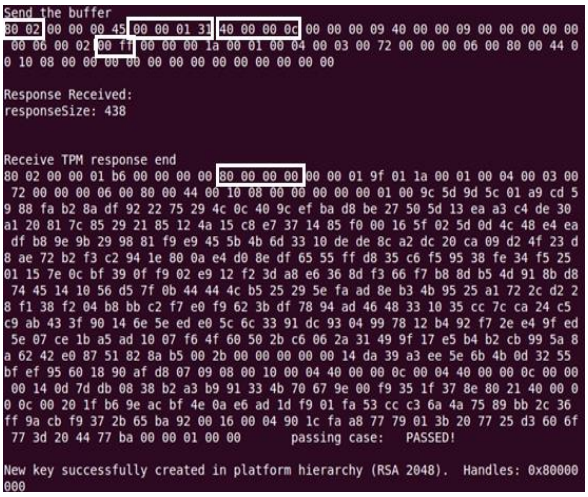

Figure 6. Test interface of Tspi_TPM2_ CreatePrimaryKey function

Parameter description for related bytes of Tspi_TPM2_ CreatePrimaryKey test interface is as follows:

The data packet transmitted is in hexadecimal form. Two digits represent one byte. Important data identified in the figure are described as follows.

- The value of the representative parameter TPMI_ST_COMMAND_TAG of the first two bytes is TPM_ST_SESSIONS=0x8002. It indicates that the command has one or more connection conversations. The authorization value is within the current range;
- The value of the representative parameter TPM_CC of the 7-10 bytes is TPM_CC_CreatePrimary=0x00000131, which means that a primary cryptogrammic key has been generated;
- The value of the representative parameter TPM_AUTH_HANDLE of the 11-14 bytes is TPM_RH_PLATFORM= 0x4000000C, which means that it is a cryptogrammic key for platform identification;
- Set Primarykey password to 0x 00 ff. Data is received from TPM and the generated handle is 0x 80 00 00 00 which is expressed as an ordinary temporary object,and transferred to the next function Tspi_TPM2_ CreateKey.

The test results indicate that the data returned from TPM is received and the primary cryptogrammic key of the platform is established successfully.

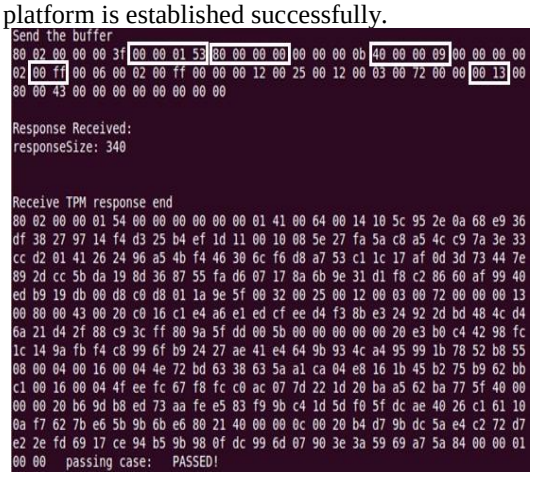

Figure7. Test interface of Tspi_TPM2_ CreateKey function

Parameter description for related bytes of Tspi_TPM2_ CreateKey test interface is as follows:

- Representative parameter TPM_CC_Create=0x0000 0153 of the 7-10 bytes indicates the operation of cryptogrammic key generation;
- The 11-14 bytes represent the TPMI_DH_OBJECT parent handle of @parentHandle=0x80 00 00 00. It is a handle generated by Tspi_TPM2_ CreatePrimaryKey function;
- The value of TPM_AUTH_HANDLE is TPM_RS_PW= 0x40000009, which indicates that the password selected is cryptography verification;
- Authorization code used is 00 ff; the 2-3 bytes count backwards of line 2 represent that the value of TPM_ALG_SM4 is 0x0013, which means that SM4 cryptography algorithm is selected.

The test results show that the handle of the primary cryptogrammic key has been received successfully, the cryptogrammic key is generated successfully and SM4 cryptography algorithm is supported.

5 Conclusions

This paper proposed a TCSS design method by referring to the ideal of CC, extracted security feature set based on analysis of security functional requirements of TCSS, determined corresponding subfunctions and core functions as well, and produced a TCSS prototype combining with practical application requirements. Finally, the completed TCSS related functional functions are tested. The experiment indicates that TCSS prototype produced in the paper can provide cryptographic support services, trusted authentication, integrity protection and meet other security targets, as well as support call of national cryptographic algorithm SM4. TCSS design specified by TPM2.0 will be further improved and supported in the future to support remote applications.

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China (61272452), the Educational Commission of Hubei Province of China(B2016257),the Public Security Department Of Hubei Province(hbst2014yycx04),the Hubei University Of Police (2015ZD006).

References

1. SHENG C X.Building Network Security with Trusted Computing[J].Qiushi journal,2015 (20): 33-34.
2. Shen C X, Zhang H G, Wang H M, et al. Research and development of the Trusted Computing[J]. Science China, 2010, 40(2):139–166.
3. Proudler G, Chen L, Dalton C. Trusted Computing Platforms: TPM2. 0 in Context[M]. Springer, 2015.

4. International Business Machine. TrouSerS[EB/OL]. [2008-10-03] [2017-05-28]. http://trousers.sourceforge.net/ .
5. IAIK Graz University of Technology.Trusted Computing API for Java[EB/OL].[2011-12-05] [2017-5-28]. https://jcp.org/en/jsr/detail?id=321 .
6. Lenovo.Client Security Solution 8.3 for ThinkVantage[EB/OL].[2014-02-28] [2017-05-26]. http://think.lenovo.com.cn/support/driver/detail.aspx ?docID=DR1256885488596.
7. Trusted Computing Group. TCG Software Stack (TSS) Specifiction,Version 1.2, Errata A [EB/OL]. [2009-03-01] [2017-5-21].http://www.trustedcomputinggroup.org/resources/tcg_software_stack_tss_specification.
8. Common Criteria for Information Technology Security Evaluation, Part1 Introduction and general model,Version 3.1, revision 5[S]. Common Criteria Recognition Arrangement Management Committee,2017:11-12.
9. CCRA Management Committee.CCRA:Arrangement on the recognition of Common Criteria certificates in the field of information technology security[EB/OL].[2014-07-02] [2017-5-28].http://www.commoncriteriaportal.org/ccra/.
10. Standardization Administration of the People's Republic of China.GB/T 18336.1-2008, Information technology-Security techniques-Evaluation criteria for IT security-Part1: Introduction and general model[S]. Beijing:Standards Press of China,2008:11-12
11. SHI H S, GAO J P, JIA W, et al. Analyse of the security architecture and policy model in the Common Criteria[J]. Journal of Tsinghua University (Science and Technology), 2016, 65(5): 493-498.
12. Common Criteria for Information Technology Security Evaluation, Part2 Security functional components,Version 3.1, revision 5[S]. Common Criteria Recognition Arrangement Management Committee,2017:48-51.