2013 International Conference on Cyber Science and Engineering (CyberSE 2013)
ISBN: 978-1-60595-134-8

A Public-Verifiable Secure Logging Scheme on Chinese Trusted Platform

PENG SHUANGHE, FAN MENG, QIN YINGJIE
and CHEN ZHIGE

ABSTRACT

System logs are an important part of any secure IT system. In this paper, a secure logging scheme using Merkle hash tree is proposed to protect the integrity of audit log files on Chinese Trusted Computing Platform. Root node of the tree is signed by Trusted Cryptography Module, and the signature is verified at the server side. The proposed scheme is efficient and suitable for the environment with meager resources.

Key words: Trusted Cryptography Module; Combined Public Key; Secure Logging; Merkle Hash Tree.

I. INTRODUCTION

Audit logs are a fundamental digital forensic mechanism for providing security in computer systems. They are used to keep track of important events about the system activities such as program executions/crashes, data modifications, and user activities. The forensic value of audit logs makes them an attractive target for attackers, who aim to erase the traces of their malicious activities recorded by logs. Providing security for audit logs on a trusted machine is a challenging task, especially in the presence of active adversaries.

A public-verifiable scheme allows auditors outside the system to make sure no tampering takes place within the system. Therefore, it can be used for systems which need to be publicly audited, such as financial records for public companies and voting systems in democratic countries.

Unfortunately, existing secure audit logging schemes have significant limitations that make them impractical to be used on devices with meager resources.

In 2009, Attila A. Yavuz and Peng Ning proposed BAF[1], a public verifiable secure logging scheme under appropriate computational assumptions, which does not require any online Trusted Third Party(TTP) support. FI-BAF extends the BAF by offering a fine-grained log verification without compromising its security[2]. Both BAF and FI-BAF require logging client to support elliptic curve cryptosystem, which can’t always be met in resources limited environment.

In this paper, a public-verifiable secure logging scheme is proposed by using trusted cryptography module(TCM)[3] based on combined public key (CPK) cryptography[4]. The TCM supports efficient implementations of common cryptographic algorithms such as SM3 cryptographic hash algorithm[8], SMS4 symmetric encryption/decryption algorithm[8] and SM2 public key encryption/decryption and signature algorithm[6]. All these algorithms are designed by Chinese. Besides, Merkle hash tree is used in the proposed scheme to reduce the number of signatures so that to improve the efficiency of the scheme.

II. BACKGROUND

Elliptic Curve Digital Signature Algorithm(ECDSA) is the public key cryptography methodology. In this paper, we use ECDSA as the base to implement CPK. CPK and TCM are summarized in this section. For a comprehensive description, the reader is referred to the CPK overview and to specification of TCM. Following notations and definitions are employed in this paper.

TABLE I. NOTATIONS AND DEFINITIONS

Notations	Definitions
d_A	The longterm private key of client A
P_A	The longterm public key of client A
[a, b]	The set of integers x such that a ≤ x ≤ b
Fq	The finite field containing q elements
E(Fq)	The set of all points on an elliptic curve E defined over Fq and including the point at infinity$\odot$
G	A distinguished point on an elliptic curve called the base point or generating point
[k]P	Scalar multiplication of a point P, [k]P = $\underbrace{P+P+\cdots+P}_{ktimes}$
ψ	Function of public key index
ID_X	Identification of X
PSK	Public key matrix in combined public cryptography system
SSK	Private key matrix in combined public cryptography system
k_X	A key pair with public key K_X and private key K_x^{-1}

2.1 Ideal of Combined Public Key

The advantage of using combined public key cryptosystem is to reduce the task of key management. The main idea of CPK is that the huge number of public and private key pairs of user are generated from matrixes with small scale. Given the parameters of the elliptic curve cryptosystem, the private matrix SSK and public matrix PSK can be created as follow.

$$\text{SSK} = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,t} \\ d_{2,1} & d_{2,2} & \cdots & d_{2.t} \\ \cdots & \cdots & \cdots & \cdots \\ d_{s,1} & d_{s,2} & \cdots & d_{s,t} \end{bmatrix} \tag{1}$$

$$\text{PSK} = \begin{bmatrix} d_{1,1}\cdot G & d_{1.2}\cdot G & \cdots & d_{1,t}\cdot G \\ d_{2,1}\cdot G & d_{2,2}\cdot G & \cdots & d_{2,t}\cdot G \\ \cdots & \cdots & \cdots & \cdots \\ d_{s,1}\cdot G & d_{s,2}\cdot G & \cdots & d_{s,t}\cdot G \end{bmatrix} = \begin{bmatrix} (x_{1,1}, y_{1,1}) & (x_{1,2}, y_{1,2}) & \cdots & (x_{1,t}, y_{1,t}) \\ (x_{2,1}, y_{2,1}) & (x_{2,2}, y_{2,2}) & \cdots & (x_{2.t}, y_{2,t}) \\ \cdots & \cdots & \cdots & \cdots \\ (x_{s,1}, y_{s,1}) & (x_{s,2}, y_{s,2}) & \cdots & (x_{s,t}, y_{s,t}) \end{bmatrix} \tag{2}$$

Where d_{ij} is the scale of point (x_{ij}, y_{ij}) with base point G, so SSK is the scale matrix of PSK relative to base point G. PSK is the public matrix and SSK is the private matrix of CPK. If SSK and PSK are both $s \times t$, at most s^t private and public key pairs can be generated. The public key of user is computed from public matrix and mapping function concerned. The mapping function ψ is chosen to be sure that different users have different mapping values. If the mapping serial values are $(a_1, a_2,..., a_t)$, then key pair of user X can be computed from PSK and SSK respectively.

$$\begin{aligned} K_x &= (x_{a_1,1}, y_{a_1,1}) + (x_{a_2,2}, y_{a_2,2}) + \cdots + (x_{a_t,t}, y_{a_t,t}) \\ K_x^{-1} &= d_{a_1,1} + d_{a_2,2} + \cdots + d_{a_{t,}t} \end{aligned} \tag{3}$$

2.2 TCM and its Commands Concerned

When we talk about trusted computing, we can’t miss TPM, the chip designed by TCG. TPM specification[5] is made by TCG, its current version is 1.2, using the following cryptographic algorithms: RSA, SHA1, and HMAC.

In China, the initiative research of trusted computing is not late, and the achievements are plentiful and substantial. In 2006, hosted by the State Cryptography Administration Committee of China, two specifications, cryptographic technology specification for trusted computing, and functionality and interface specification of cryptographic support platform for trusted computing, were established. In 2007, ZTE IC developed the first TCM chip according to trusted computing specification made in China. TCM can't be viewed as the correspondent part of TPM in China though TCM is similar to TPM in structure. They have different cryptography system support. Though symmetric encryption/decryption algorithm is required in TPM specification, it has little application. Symmetric encryption/decryption algorithm SMS4 is used in TCM. Besides, they have different sign algorithms, one is RSA and the other is SM2 which is based on Elliptic Curves. TCM commands required in the proposed scheme are briefly introduced here. For a comprehensive description of the commands, the reader is referred to the TCM main specifications[3].

Tspi_Key_CreateKey() generates an asymmetric key and returns the public key in plain text and the private key encrypted with the key pair's parent key.

Tspi_Key_LoadKey() loads an asymmetric key onto the TCM and returns the key handle that points to the loaded key, thus making the key available for use in subsequent TCM commands.

Tspi_Key_UnLoadKey() unloads an asymmetric key that has been loaded into TCM thus making the space available for use to other key.

Tspi_Key_WrapKey() wraps the private key hKey using the public key addressed by hWrappingKey. On successful return from this call, hKey can be loaded into a TCM.

Tspi_Key_GetPubKey() gets the public portion of a given key object.

Tspi_Hash_Sign() signs the hash data of an object with a given signing key.

Tspi_Hash_VerifySignature() verifies the hash value of a given hash object with a given signature.

III. SCHEME DESCRIPTION

Definition 1. Audit Logging Tree. In this paper, Merkle hash tree[7] is used as audit logging tree. Leaf nodes of Merkle hash tree are the pieces of audit logging file. Each leaf node contains the hash value of its element. Internal nodes which correspond to the hash value of the concatenation of its children (maintaining their order). Figure 1 shows the signature and verification on audit log client and server based on audit logging tree.

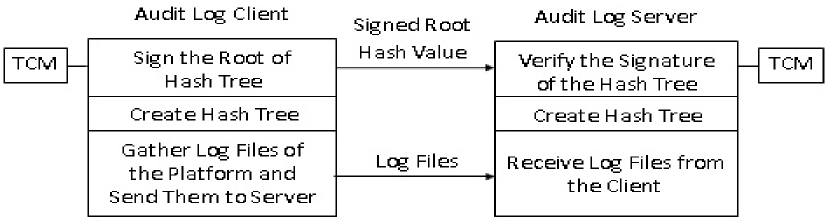

Audit Log Client
Signed Root Hash Value
Audit Log Server
TCM
Sign the Root of Hash Tree
Verify the Signature of the Hash Tree
TCM
Create Hash Tree
Create Hash Tree
Gather Log Files of the Platform and Send Them to Server
Log Files
Receive Log Files from the Client

Figure 1. Signature and Verification based on Audit Logging Tree

Firstly, audit logging client gathers log files of the platform and sends them to the audit log server. Secondly, audit logging client and server build the hash tree according to the log files concerned. Thirdly, as figure 1 shows audit logging client *X* uses TCM to compute signature of the root node using its private key after the audit logging tree is built, and sends the signed root hash value to the audit logging server. After receiving the signed root hash value, audit logging server computes the public key of the client *X* according to the client's ID_X from the public key matrix PSK and verifies the signature of the root node using TCM.

3.1 Work Flow

Figure 2 shows the work flow of the proposed scheme.

It is composed of five main steps.

(1) ECC Key pair generation. The ECDSA algorithm needs the domain parameters $T = (a, b, G, n, q)$ for perform signature generation and signature verification. Besides, $s \times t$ public and private key pairs are needed, where $s \times t$ is the size of PSK and SSK matrix.

(2) Generation of Key Matrix

Key management center is responsible for the generation of key matrix. The public key matrix PSK is stored plainly on hard disk, while the private key matrix SSK is encrypted using the public key of key management center.

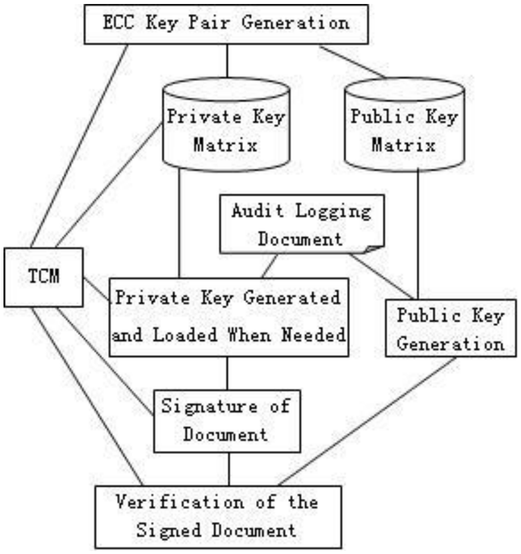


Figure 2. Work Flow of the Proposed Scheme

(3) User Key Pair Generation

User register management center is responsible for the generation of user key pair. The private part of user key is loaded into TCM when needed to sign the root value of Merkle hash tree.

(4) Signature of Root Node of Audit Log Tree on Client Side

When the audit log client wants to send audit log to the server, it builds an audit log tree based on Merkle hash tree, generates the signature of the root node of the tree using TCM and sends the signature along with the log to the server.

(5) Verification of Signature on Server Side

After receiving the signature of the audit log, the server retrieves the public key of client and verifies the signature using the public key.

3.2 Generation of Public and Private Matrix of CPK

Based on the parameter $T = (a, b, G, n, q)$ of elliptic curve cryptography, we can construct both public key matrix PSK and private key matrix SSK.

In Elliptic curve cryptography system, for a integer $k \in F_q$, $[k]G$ is the public part of the key pair. A certain number (the number is decided by the row and column requirement of the matrix) of private and public key pair can be built by the matrixes. The private matrix SSK is composed of integers *(r_{ij})*, and the public matrix PSK is composed of points *$(r_{ij}G) = (x_{ij}, y_{ij})$*.

In the scheme random $d_A \in [1, n-1]$ is generated by TCM as long-term private key. The long-term public key P_A is computed as $P_A = [d_A]G.$. If the size of key matrix is $s \times t$, then $s \times t$ random should be generated and the public key be computed accordingly.

3.3 User Key Pair Generation and Use

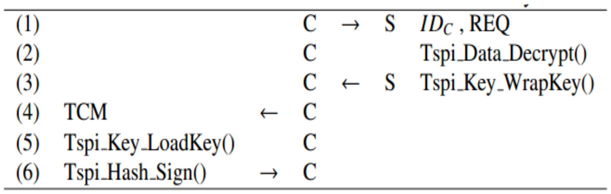
Key management center chooses a one way trap function ψ as public and private key index to select different elements of matrix to compute public key and private key of user according to its identity.

User key pair is generated as Figure 3 shows.

(1) When client *C* wants to get key pair from key management center *S*, it sends a request along with its ID_C to *S*.

(2) After receiving request from *C*, *S* decrypts the private matrix stored on hard disk and compute *C*'s public and private key according to his or her ID_C using interface *T spi_Data_Decrypt()*.

(3) *S* wraps the private part of user key pair using interface *T spi_Key_WrapKey()*, sends it to *C*.



(1)			C	→	S	ID_C , REQ
(2)			C			Tspi_Data_Decrypt()
(3)			C	←	S	Tspi_Key_WrapKey()
(4)	TCM	←	C			
(5)	Tspi_Key_LoadKey()		C			
(6)	Tspi_Hash_Sign()	→	C			

Figure 3. Generation of User Private Key and its Using

(4) When the private key is needed, *C* requests TCM to load the key.

(5) TCM uses interface *T spi Key LoadKey()* to load the key. And then all the operations concerned with the private key is processed in the TCM.

(6) TCM signs the Merkle hash tree root node by using interface *T spi_Hash_Sign()* and sends it to *C*.

IV. PERFORMANCW AND SECURITY ANALYSIS

4.1 Performance Analysis

If the number of piece of the log file is *n*, node number of the Merkle tree is about *2n a*ccording to the characteristic of the complete binary tree. So *n* × hashtime and *n* × signtime are needed in the traditional way, while in the proposed scheme, only *2n* × hashtime and one signature should be computed. As for the cost of computing, hash function is much lower than signature function. It is more efficient using the proposed scheme. Besides, only one signature value is needed to be transmitted to the server, the communication cost is also lower compared with the traditional way.

4.2 Security Analysis

Since the private key used to sign the root of Merkle hash tree is stored in the TCM and all the operations concerned with the private key are performed by the TCM, the security is improved compared with the software only method.

V. CONCLUSION AND FUTURE WORK

According to the characteristic of audit logging, Merkle hash tree is used in this paper to build audit logging tree on Chinese trusted computing platform. The root node of Merkle hash tree is signed by TCM using private key of CPK cryptography at client side, and then be verified at the server side. Key management center can run offline in isolated environment after the distribution of user private key and public key matrix. The process of sign and verification can work without the participation of the third trusted part.

The proposed scheme is efficient by verifying only the root hash of Merkle tree. It can also create audit logging tree according to different granularity, which improves the flexibility of the audit logging verification.

Only static Merkle tree is used in the proposed scheme, it can't support insert, delete and modify operations to the tree. We will use dynamic Merkle tree to implement audit file insert, modify and delete in the future.

VI. ACKNOWLEDGMENT

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant No.2011JBM221, the Program for New Century Excellent Talents in University (NCET-11-0565), the Program for Innovative Research Team in University of Ministry of Education of China (IRT201206).

REFERENCES

[1] Attila A. Yavuz and Peng Ning. BAF: An Efficient Publicly Verifiable Secure Audit Logging Scheme for Distributed Systems, Proceedings of the 25th Annual Computer Security Applications Conference, 2009:219-228.

[2] Attila Altay Yavuz, Peng Ning and Michael K. Reiter.BAF and FI-BAF: Efficient and Publicly Verifiable Cryptographic Schemes for Secure Logging in Resource-Constrained Systems, ACM Transactions on Information and System Security,15(9),2012.

[3] State Cipher Administration. Cipher Support Platform Function and Interface Specification of Trusted Computing in Chinese, 2007.

[4] Nan Xianghao. CPK Cryptosystem and Cyber Security, National Defense Industry Press,2008,12.

[5] Trusted Computing Group, Incorporated. TPM Main, Part 1 Design Principles, Specification Version 1.2, March, 2011.

[6] State Cipher Administration. Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves, Part 1: General. 2010,12.

[7] Ralph C. Merkle. A Certified Digital Signature, CRYPTO 1989:218-238.

[8] State Cipher Administration. State Cipher Administration proclamation (No.7) [EB/OL].http://www.chinabwips.org/gzdt-51.htm