A TCM-Based Remote Anonymous Attestation Protocol for Power Information System

Ruizhong CHEN[1, a], Lihao WEI[1, b], Hong ZOU[1, c], Meijie ZHAI*[2,d]

[1]Information Center, Guangdong Power Grid Corporation, Guangzhou, 510000, China

[2]SKLOIS, Institute of Information Engineering, CAS, Beijing, 100093, China

[a]email: crzlhs@163.com, [b]email: wlh_wind@126.com, [c]email:zouhong@gdxx.csg.cn,

*Corresponding author: [d]email: zhaimeijie@iie.ac.cn

Keywords: Remote Anonymous Attestation; Attribute Credential; Ring Signature; ECC; TCM

Abstract. Project development in a power enterprise always needs to authorize external devices access to the enterprise intranet for testing. In order to avoid an external device with a virus and pose a security risk to the power information system, external devices should have strict security assessment before access the enterprise intranet. But after the security assessment, the device user still be possible to change the platform configuration. Remote attestation is one of important measures when two sides need to communicate. It is concernful to attest the remote platform is trusty but not revealing the any private information of the platform. For this reason, we designed a novel remote anonymous attestation protocol based on TCM. The proposed protocol does not need extra zero knowledge proof and the involvement of the third trusted party and the composite signature scheme is proved secure against existential forgery on adaptively chosen message. So this protocol has better security and execution property.

Introduction

Terminals security is one of the biggest problems in the power intranet. Project development in a power enterprise always needs to authorize external devices access to the enterprise intranet for testing. In order to avoid an external device with a virus and pose a security risk to the power information system, external devices should have strict security assessment before access the enterprise intranet. But after the security assessment, the device user still be possible to change the platform configuration. In addition, it is very likely reveal platform private information in remote attestation process.

Remote attestation is one of important measures when two sides need to communicate. It is concernful to attest the remote platform is trusty but not revealing the any private information of the platform. In 2004, E. Brickell at el. introduced a new remote attestation scheme named Direct Anonymous Attestation (DAA) based on Trusted Platform Module (TPM) proposed by Trusted Computing Group (TCG) [1]. DAA employs Zero Knowledge Proof advanced by S.Goldwasser, S. Micali, and C. Racko [2] and Group Signature put forward by D.Chaum [3] to realize the anonymousness and omit the trusted third party (Private CA). DAA had become a part of TCG specifications (version 1.2) in 2006 [4]. However, the several times of Zero knowledge Proof make it not efficient enough.

A. R. Sadeghi proposed an efficient protocol called Property-based Attestation (PBA) in 2004 [5]. A property indicates various platform configurations in this scheme to protect the platform from revealing configuration information. Therefore, verifier knows nothing about specific platform configuration but the information that the platform has a certain property. Based on this protocol, L. Chen and R. Landfermann introduced a new property-based attestation [6]. A property and its corresponding configuration information are contained in a Property-Configuration Certificate issued and managed by a trusted third party. This method also uses Zero Knowledge proof to conceal the real configuration information, so that it is hard to realize. Moreover, the trusted third party must store the information of all platform configurations and sign them.

J. Q. Liu proposed a remote attestation protocol based on TPM Remote Anonymous Attestation (RAA) [7]. This protocol protects private information without additional zero knowledge proof so that it can be realized effectively.

In December 2007, the specification named Functionality and Interface Specification of Cryptographic Support Platform for Trusted Computing is issued by State Cryptography Administration [8]. In this specification, the theory and demand of trusted cryptographic support platform are described and application interface specifications of cryptographic algorithm, key management, certification management, cryptographic protocol and cryptographic service for trusted cryptographic support platform are defined. A new trusted module called Trusted Cryptographic Module (TCM) which has an ECC engine is used as a substitute for TPM in this specification. According this specification, we proposed a new TCM-based remote anonymous attestation protocol. Our protocol not only satisfies the requirement of ECC algorithm, but also has better security characters, such as private information protection and resistance of impersonation attack. And we prove our TCM-based RAA protocol is secure against existential forgery on adaptively chosen message under the elliptic curve discrete logarithm problem (ECDLP).

ECC encryption algorithm

In this paper, we amend the Elliptic Curve Integrated Encryption Scheme（ECIES）[9]. Let f , g be the encryption and decryption function. $H_2()$ is a one-way hash function. $Enc(,)$ and $Dec(,)$ are the symmetric encryption and decryption functions with l_2 bit secret key Key. For plaintext m and opposite ciphertext c, the functions must satisfy the equation: $m = Dec(Enc(m, Key), Key)$. Assume A and B are two sides of communication. Elliptic curve is $E(F_p)$: $y^2 \equiv x^3 + ax + b (\text{mod } p)$,where $a, b \in F_p$. The base point is $G = (x_G, y_G)$ on $E(F_p)$. The prime number n is the order of base point G and integer $h = \#E(F_p)/n$. Therefore, the parameters of elliptic curve on finite field F_p are $T = (p, a, b, G, n, h)$, and they must satisfied some restrictions[9].

Algorithm 1. is our ECC encryption algorithm. After receiving the ciphertext $c = (R_1, EM)$, decryption implements with the private key d_B by the **Algorithm 2**.

Algorithm 1. ECC encryption

```
01: Function ECC_ENC (){
02:   while (A got plaintext m and public key Q_B){
03:     j = H_1(m) ; /* H_1 : {0,1}^* → Z_n */
04:     R_1 = (x_{R_1}, y_{R_1}) = jG  and  R_2 = (x_{R_2}, y_{R_2}) = jQ_B ; /* Q_B is the public key of B.*/
05:     if ( R_1 != O & R_2 != O ){
06:       z = x_{R_2} ∈ F_p ;
07:       Key = H_2(z) ;  /* H_2 : {0,1}^{l_1} → {0,1}^{l_2} , l_2 is the length of Key, the secret key of
            symmetric encryption/decryption function.*/
08:       EM = Enc(m, Key) ;
09:       Output c = R_1 || EM ;}
10:     else break;}}
```

Algorithm 2. ECC decryption

```
01: Function ECC_DEC (){
02:   while (ciphertext c = R_1 || EM received){
03:     Separate R_1 and EM from c;
04:     if ( R_1 != O ){
```

05: $R_2 = (x_{R_2}, y_{R_2}) = d_B R_1$; /* d_B is the private key of B.*/

06: $z' = x_{R_2} \in F_p$;

06: $Key' = H_2(z')$;

07: Decrypt *EM* with Key' to get $m' = Dec(EM, Key')$;

08: Compute $j' = H_1(m')$, $R_1' = j'G$ and $R_2' = j'Q_B$ in turn;

09: **if** ($R_1 == R_1' \& R_2 == R_2'$){

10: Output correct plaintext m';}

11: **else break**; **}**

12: **else break**; }

Ring Signature

Ring signature was proposed by R. L. Rivest, A. Shamir and Y. Tauman in 2001[10]. A set of possible signers is called a ring. There is only one actual signer who produces the signature and each of the other ring members is called non-signer. A ring signature scheme is set-up free: the signer does not need the assistance of the other non-signer except the regular public keys of them in the ring. The signer can choose the size of the ring and the member of the ring by himself. And signer can choose different public key signature algorithm freely.

We suppose that each possible signer is associated with a public key Q_s and a corresponding secret key d_s to generate regular signatures. There are two stages in a ring signature scheme: ring signature generation and ring signature verification.

Given the message m and the public keys $Q_1, Q_2, \cdots, Q_t$ of the t ring members, the actual signer will generate a ring signature $\sigma = (Q_1, Q_2, \cdots Q_t, v, x_1, x_2, \cdots, x_t)$ as **Algorithm 3**. When receiving the signature credential, a verifier can verify an alleged signature $\sigma = (Q_1, Q_2, \cdots Q_t, v, x_1, x_2, \cdots, x_t)$ on the message m with **Algorithm 4**.

Algorithm 3. Ring signature generation

01: **Function** RING_SIG_GEN () {

02: **while** (need to generate ring signature){

03: $k = H_0(M, Q_1, Q_2, \cdots, Q_t)$; /**k* is a symmetric key. */

04: Pick a value v uniformly at random; /* $v \in \{0,1\}^b$, 2^b is larger than module *n*.*/

05: **for** (*i* =1; *i* <=*t*; *i* ++) {

06: **if** (*i* ==*s*) **continue**;

07: $y_i = f(x_i)$; }/*use **Algorithm 1**.*/

08: Solve y_s from $C_{k,v}(y_1, y_2, \cdots, y_t) = v$;

09: $x_s = g(y_s)$; /*use **Algorithm 2**.*/

10: Output $(2t+1)$-tuple credential $(Q_1, Q_2, \cdots Q_t, v, x_1, x_2, \cdots, x_t)$;}}

Algorithm 4. Ring signature verification

01: **Function** RING_SIG_VERI () {

02: **if** (receiving ring signature credential) {

03: $k = H_0(M, Q_1, Q_2, \cdots, Q_t)$; /**k* is the symmetric key. */

04: **for** (*i* =1; *i* <=*t*; *i* ++) {

05: $y_i = f(x_i)$;} /* use **Algorithm 1.***/

06: **if** (y_i's satisfy the equation: $C_{k,v}(y_1, y_2, \cdots, y_t) = v$){

06: Accept the ring signature;}

07: **else break**; }}

The Remote Anonymous Attestation Protocol Based on TCM

Generally, for attestation of the platform, challenge requests the configuration information of platform in attestation of the platform and verifies the configuration information and the signature.

Without loss generality, assume the platforms A and B are two sides which need remote anonymous attestation, where platform A is the requester and platform B is the resource provider. Each platform of the two sides must have TCM with standard ECC engine. And each TCM owns a pair of ECC secret key (d,Q), which d is the private key and Q is the public key. Attribute P corresponds with several platform configurations $C_1,C_2,\cdots,C_t$ in attribute credential.

Initialization

When platform A needs to prove that it has some certain attributes, it selects other $t-1$ platforms with same kind of security chips to compose a ring. The public keys of these platforms can be obtained from their public key certificates. To facilitate, we let $Q_1,Q_2,\cdots,Q_t$ be the t public keys. In addition, platform A needs to define the parameters of the ECC algorithm. The parameters of elliptic curve on finite field F_p are $T=(p,a,b,G,n,h)$. Finally, platform A chooses its private key $d_A\in[1,n-1]$, and computes point $Q_A=(x_{Q_A},y_{Q_A})$ on the elliptic curve which $Q_A=d_AG$ as its public key.

TCM-Based RAA Protocol

Because TCM has limited computing and storage capacity, the generation and computing of unimportant data are completed outside it. From the above assumption, the ring is composed of *t* platforms with TCM, and their public keys are $Q_1,Q_2,\cdots,Q_t$ respectively. Let the sequence number of the real signer (platform A) is *s*.

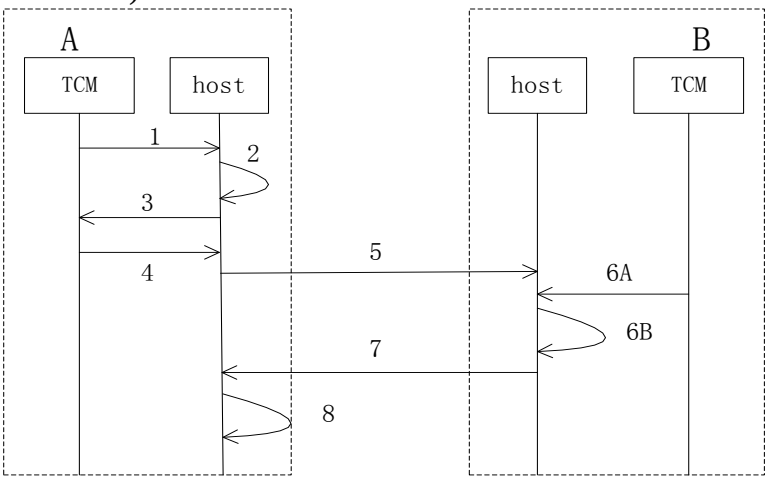

Figure 1. Process of TCM-based remote anonymous attestation

The process of TCM-based remote anonymous attestation is shown in Figure 1:

1) Platform A randomly selects a private information $x_A(1\le x_A\le n-1)$, then takes out the configuration digest values from the Platform Configuration Registers(PCRs) in TCM, and send them to the host.
2) Host A calculates $x_AG=(x',y')$ and $r=x' \bmod n(r\neq 0)$, platform A computes the attribute hiding value $y_A=x_A^{-1}(H_3(P,C_1,C_2,\cdots,C_t)+d_Ar) \bmod n\ (y_A\neq 0)$. So the information need to be signed by platform A is $M=(p,a,b,G,n,h,Q_A,y_A,r,P)$. Platform A computes hash value $k=H_0(M,Q_1,Q_2,\cdots,Q_t)$, and picks a random integer as original value v and a random sequence $x_1,\cdots,x_{s-1},x_{s+1},\cdots,x_t$. Then it respectively computes $y_i=f(x_i),1\le i\le t,i\neq s$ using Algorithm 1. Platform A chooses the following ring equation:

$$C_{k,v}(y_1,y_2,\cdots y_t)=E_k(y_t\oplus E_k(y_{t-1}\oplus E_k(\cdots\oplus E_k(y_1\oplus v)\cdots)))=v \quad (1)$$

where $E_k()$ denotes symmetric encryption algorithm and $k = H_0(M, Q_1, Q_2, \cdots, Q_t)$ is the key, $\oplus$ indicates bit XOR operation. Platform A figures out y_s from ring equation (1):

$$y_s = E_k(y_{s-1} \oplus E_k(y_{s-2} \oplus E_k(\cdots \oplus \cdots E_k(y_1 \oplus v)\cdots))) \oplus D_k(y_{s+1} \oplus D_k(y_{s+2} \oplus D_k(\cdots \oplus D_k(y_t \oplus D_k(v))\cdots)))$$

where $D_k()$ is the decryption algorithm corresponding to $E_k()$.

3) After that, host A sends y_s into TCM and works out $x_s = g(y_s)$ using Algorithm 2 with private key d_A inside TCM.
4) TCM sends x_s back. Then platform A produces a $(2t+1)$ -tuples credential $(Q_1, Q_2, \cdots, Q_t; v; x_1, x_2, \cdots, x_t)$.
5) Platform A sends the information $M = (p, a, b, G, n, h, Q_A, y_A, r, P)$ and credential $(Q_1, Q_2, \cdots, Q_t; v; x_1, x_2, \cdots, x_t)$ to platform B.
6) Platform B receives the information and attributes credential signed inside TCM from platform A. B computes $b_1 = H_4(d_B G)$ inside TCM and sends it to the host. Host B validates the ring signature using Algorithm 4. If the signature is illegal, platform B rejects the information; Otherwise, platform B picks a random private information $x_B (1 \le x_B \le n-1)$ and computes $u_1 = H_3(P, C_1, C_2, \cdots, C_t) y_A^{-1} \bmod n$, $u_2 = r y_A^{-1} \bmod n$, then calculates $k_1 = H_5(b_1 x_B (u_1 G + u_2 Q_A))$.
7) Platform B sends $x_B G$ and resource R encrypted with secret key k_1 to platform A.
8) Platform A calculates $b_2 = H_4(Q_B)$ and $k_2 = H_5(b_2 x_A (x_B G))$, If $k_1 = k_2$, platform A can get the resource R by decrypting the message with key k_2, namely, platform A has attribute P; Otherwise, platform A can't obtain the resource from platform B.

Security Analysis

For the anonymity, we use ring signature to conceal the identity of platform. Any adversary has probability at most $1/t$ to determine the identity of the actual signer in a ring of size *t*.

In this protocol, property P denotes various platform configurations $C_1, C_2, \cdots, C_t$ so as to hiding the true attribute C_r. The verifier can't infer the specific configuration from property P. Moreover, platform A computes attribute hidden value $y_A = x_A^{-1}(H_3(P, C_1, C_2, \cdots, C_t) + d_A r) \bmod n$ with private value x_A picked by itself and create the certificate $(Q_1, Q_2, \cdots, Q_t; v; x_1, x_2, \cdots, x_t)$ inside TCM. Consequently the configuration values are non-forgeable. Private values x_A and x_B make communication fresh between A and B. Adversary can't impersonate B to provider malicious resource, because of the involvement of d_B in computation when verification.

For security, we have following theorem:

Theorem 1 *The TCM-based RAA protocol is secure against existential forgery on adaptively chosen message under the elliptic curve discrete logarithm problem (ECDLP).*

Conclusion

In this paper, we design a remote anonymous attestation scheme based on TCM for power information system. This protocol does not need zero knowledge proof and the involvement of the third trusted party. And from analysis of security, it can prevent the platform configuration information from leaking, resist impersonation attack. Furthermore, it is secure against existential forgery on adaptively chosen message under the ECDLP.

Acknowledgement

The research was sponsored by the Information Center of Guangdong Power Grid Corporation's project of Study on Data Security in Big Data Environments (No.K-GD2014-1019) and Xinjiang Uygur Autonomous Region science and technology plan (No.201230121), the Strategic Priority

Research Program of Chinese Academy of Sciences (No. XDA06040602).

References

[1] E. Brickell, J. Camenisch, and L. Chen, Direct Anonymous Attestation, In Proceedings of the 11th ACM Conference on Computer and Communications Security, Washington, DC, USA, Oct. 2004.

[2] S. Goldwasser, S. Micali, and C. Racko, The Knowledge Complexity of Interactive Proofs, SIAM J. Comput., 18(1), 186-208, 1989.

[3] D. Chaum and E. van Heyst. Group signatures. In: Advances in Cryptology-EUROCRYPT'91, LNCS 950, 257-265. Springer-Verlag, 1992.

[4] Trusted Computing Group , TPM Main Specification, main Specification Version 1.2 rev.94, 29 March 2006.

[5] A. R. Sadeghi and C. St¨uble, Property-based attestation for computing platforms: Caring about properties, not mechanisms, In The 2004 New Security Paradigms Workshop, Virginia Beach, VA, USA, Sept. 2004.

[6] L. Chen, R. Landfermann, H. L¨ohr, M. Rohe, A. Sadeghi, and C. St¨uble, A Protocol for Property-Based Attestation, STC'06, Alexandria, Virginia, USA, November 3, 2006.

[7] J. Q. Liu, J. Zhao and Z. Han. A remote anonymous attestation protocol in trusted computing.

[8] State Cryptography Administration. Functionality and Interface Specification of Cryptographic Support Platform for Trusted Computing, Dec. 2007.

[9] Certicom Research. SEC 1: Elliptic Curve Cryptography, Version 1.7, November 13, 2006.

[10] R. L. Rivest, A. Shamir and Y. Tauman. How to Leak a Secret: Theory and Applications. In Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, volume 2248 of Lecture Notes In Computer Science, 552 – 565, Springer, Jan. 2001.