*Submit to the **Workshop on Cryptology and Information Security**, ICS2002*

Repairing ElGamal-like multi-signature schemes using self-certified public keys

Hwang, Shin-Jia and Lee, Yun-Hwa*

Department of Computer Science and Information Engineering,

TamKang University, Tamsui, Taipei Hsien, 251, Taiwan, R.O.C.

E-mail: sjhwang@mail.tku.edu.tw

*E-mail: 690190235@s90.tku.edu.tw

Abstract

Recently, Chang et al. proposed an ElGamal-like multi-signature scheme using self-certified public keys. Being inspired of the insider attack, an insider attack is proposed on their scheme to show that a malicious member in the signing group can forge a valid multi-signature without the other members' secrete keys. To remove this attack, an improved multi-signature is also proposed.

Keywords: multi-signatures, insider attacks, digital signatures

Correspondence address:

Hwang, Shin-Jia
Department of Computer Science and Information Engineering,
TamKang University, Tamsui, Taipei Hsien, 251, Taiwan, R.O.C.
Tel: +886-2-26215656 ext 2727
Fax: +886-2-26209749
E-mail: sjhwang@mail.tku.edu.tw

Repairing ElGamal-like multi-signature scheme using self-certified public keys

Abstract

Recently, Chang et al. proposed an ElGamal-like multi-signature scheme using self-certified public keys. Being inspired of the insider attack, an insider attack is proposed on their scheme to show that a malicious member in the signing group can forge a valid multi-signature without the other members' secrete keys. To remove this attack, an improved multi-signature is also proposed.

Keywords: multi-signatures, insider attacks, digital signatures

1.INTRODUCTION

Digital signature and multi-signature are the major research topics in modern cryptography and computer security. Digital signatures can be used to sign the digital messages because hand-written signatures cannot. To expand the applications of the digital signature schemes, Itakura and Nakamura [5] proposed the first multi-signature scheme in which a group of members have to sign a message by all members in the group cooperatively.

Some famous digital signature schemes are proposed base on the well-known public key system [2, 3, 7]. In these signature schemes, a signer has to keep a secret key and to publish his public key. By using his public key, anyone can validate his digital signatures. So his public keys must be first authenticated. To authenticate public keys, there are three possible approaches: the certificate-based, the identity-based and the self-certified approaches. The self-certified approach provides more secrecy against the active and impersonation attacks than that provided in the other two approaches [1]. So Chang et al proposed their ElGamal-like

signature and multi-signature schemes using self-certified public keys [1].

However, being inspired of the proposed insider attack [6, 11], a new attack is proposed on Chang et al's ElGamal-like multi-signature scheme. Through this new attack, a malicious member in the signing group is able to forge a valid multi-signature for a message by himself. To guard against this new attack, an improved ElGamal-like multi-signature scheme using self-certified public keys is also proposed.

In the following section, the brief review of Chang et al.'s signature and multi-signature schemes is given. On Change et al.'s multi-signature scheme, an insider attack is proposed in Section 3 while the improved scheme is present in Section 4. Section 5 is our security and performance analysis. The final section is our conclusion.

2. BRIEF REVIEW OF CHANG ET AL'S ELGAMAL-LIKE SCHEMES

Girault's self-certified key system based on RSA [4] is described first since Chang et al's schemes are proposed based on Girault's system. Then Chang et al's ElGamal-like schemes are reviewed. [1]

Girault's self-certified public key system based on RSA:

Girault's scheme contains two phases: System initial phase and user register phase. Two phases are described in the following, respectively.

System initialization phase

To set up the system, a system authority (SA for short) performs the following steps to generate the following system parameters.

Step 1: Generate two random secret prime integers p and q. Then compute the public product N= pq.

Step 2: Select a generator g with the maximal order in the multiplicative group Z_N^*.

Step 3: Construct the key pair (e, d) satisfying $e \times d \equiv 1 \pmod{\phi(N)}$. Then e is the public key of SA and d is the secret key of SA.

User registration phase

Assume each user U_i has a unique identity ID_i. Suppose that the user U_i wants to construct his public key. He first randomly choose an integer $x_i \in Z_N^*$ as his secret key. He computes $v_i = g^{-x_i} \bmod N$ and sends $\{ID_i, v_i\}$ to the SA. Then the SA computes the public key $y_i = (v_i - ID_i)^d \bmod N$ for the user U_i. Finally, the user U_i has a self-certified public key y_i.

Based on Girault's system, Chang et al. proposed their digital signature and multi-signature schemes. Their signature scheme is described as below.

Chang et al's digital signature scheme

Chang et al's digital signature scheme contains four phases: system initialization, user registration, signature generation, and signature verification phases. The system initialization and user registration phases are the same as the corresponding phases in Girault's system. The signature generation and signature verification phases are described below.

Signature generation phase

Suppose that M is the signing message. The signer U_i choices an integer $w_i \in Z_N$ and computes the signature (r_i, s_i) for M, where $r_i = g^{w_i} \bmod N$, and $s_i = w_i + x_i \times h(M, r_i)$. Then U_i sends M and signature (r_i, s_i) to the verifier.

Signature verification phase

After receiving M and its signature (r_i, s_i), the verifier checks the correctness of the signature by the equation $g^{s_i} \times (y_i^e + ID_i)^{h(M, r_i)} \equiv r_i \pmod N$. If the equation holds, the verifier accepts the validity of the signature of M.

Chang et al's multi-signature scheme

Chang et al's multi-signature scheme contains four phases: system initialization, user registration, multi-signature generation and multi-signature verification phases. The system initialization phase is the same as the corresponding phase described in Girault's system. The other three phases are described as below.

User registration phase

Let $G= \{U_1, U_2, \ldots, U_n\}$ be the registered group and GID be the identity of the group G. When all individuals U_i's in the group G have been registered, the SA computes the group public key $Y= ((\Pi_{i=1}^{n} v_i)- GID)^{d} \pmod N$ for the group G and then the group secret key of G is $X= \Sigma_{i=1}^{n} x_i \pmod{\phi(N)}$. All individuals in the group G can cooperatively verify the validity of Y by presenting $v_i= g^{-x_i} \pmod N$ to the others and check $Y^{e}+ GID \equiv \Pi_{i=1}^{n} v_i \equiv g^{-X} \pmod N$.

Multi-signature generation phase

Suppose that the message M is the message signed by all individuals in the group G. Each member U_i choices his secret integer w_i in Z_N, computes $r_i= g^{w_i} \bmod N$, and broadcasts r_i to the clerk and the other members in the group. After receiving all r_j's from all members in the group G, the clerk and all members in the group G compute the product $R= \Pi_{i=1}^{n} r_i \bmod N$. Then each member U_i in the group G computes $s_i= w_i+ x_i h(M,R)$ and sends s_i to the clerk. The clerk verifies every individual signature (r_i, s_i) by the equation $g^{si} \times (y_i^{e}+ ID_i)^{h(M,R)} \equiv r_i \pmod N$ for i=1, 2, …, n. If all (r_i, s_i)'s have been verified, the clerk computes $S= \Sigma_{i=1}^{n} s_i$ and publishes (R, S) as the multi-signature of the message M generated by the group G.

Multi-signature verification phase

A verifier checks (R, S) by using the equation $g^{S} \times (Y^{e}+ GID)^{h(M,R)} \equiv R \pmod N$. If the equation holds, the verifier accepts the validity of the multi-signature (R, S) of

M.

3. AN ATTACK ON CHANG ET AL'S ELGAMAL-LIKE MULTI-SIGNATURE SCHEME

An insider attack is proposed on Chang et al's ElGamal-like multi-signature scheme. Without losing generality, suppose a malicious member U_1 purposed to forge a multi-signature on the message M. In the user registration phase, after knowing the other n-1 members' public keys, the attacker U_1 randomly selects his secret key x_1' and computes $v_1'= g^{-x_1'}\times((y_2^e+ID_2)\times(y_3^e+ID_3)\times \dots \times(y_n^e+ID_n))^{-1} \bmod N= g^{-x_1'}\times(v_2\times v_3\times \dots \times v_n)^{-1} \bmod N$ and send v_1' to the SA. The SA computes U_1's public key $y_1'= (v_1'- ID_1)^d \bmod N$. After receiving y_1', U_1 verifies $(y_1')^e +ID_1= v_1' \bmod N$. Then the group public key of G is $Y= (v_1'\times\prod_{i=2}^{n} v_i - GID)^d \bmod N= ((v_1' \times v_2 \times \dots \times v_n)- GID)^d \bmod N = (g^{-x_1'}- GID)^d \bmod N$. Then $Y^e+ GID\equiv g^{-x_1'}\equiv g^{-X} \pmod N$. This also means that the group secret key of G is $X= x_1'$.

Suppose that the attacker U_1 wants to forge the multi-signature on the message M. He performs the following these steps:

Step 1: U_1 randomly selects $w_1, w_2 ,\dots ,w_n$ and then computes $r_i= g^{w_i} \bmod N$ for all i = 1, 2,…,n.

Step 2: U_1 generates the multi-signature (R, S) on M by computing $R= \prod_{i=1}^{n} r_i \bmod N$ and $S = \Sigma_{i=1}^{n} w_i + x_1'\times h(M, R)$.

The following gives the reason why the illegal multi-signature (R, S) of the message M can pass the verification.

$$g^S\times(Y^e+ GID)^{h(M,R)}\equiv (g^{\sum_{i=1}^{n} w_i}\times g^{x_1'\times h(M,R)})\times(g^{-x_1'})^{h(M,R)}\equiv R \pmod N.$$

4. OUR IMPROVEMENT

Our improved scheme is proposed in this section. Our scheme also contains four phases: system initialization, user registration, multi-signature generation and multi-signature verification phases. The system initialization phase is still the same as the corresponding phase described in Girault's system.

User registration phase

The user U_i randomly selects x_i in Z_N as his secret key, computes $v_i = g^{-x_i} \bmod N$, and sends $\{ID_i, v_i\}$ to SA. The SA computes $y_i = (v_i^{h(v_i)} - ID_i)^d \bmod N$ as the public key for U_i. When all individuals U_i's in G have been registered, the SA computes the group public key $Y = ((\Pi_{i=1}^{n} v_i^{h(v_i)}) - GID)^d \bmod N$. Then the group secret key of G is $X = \Sigma_{i=1}^{n} x_i h(v_i) \pmod{\phi(N)}$. All members in the group G can cooperatively verify the validity of Y by presenting $v_i = g^{-x_i} \bmod N$ to the others and checking $Y^e + GID \equiv \Pi_{i=1}^{n} v_i^{h(v_i)} \equiv g^{-X} \pmod N$.

Multi-signature generation phase

Suppose that M is the signing message. Each signer U_i choices his secret integer w_i in Z_N, computes $r_i = g^{w_i} \bmod N$, and sends it to the clerk and the other members. After receiving all r_j's from all members in the group G, the clerk and all members compute the product $R = \Pi_{i=1}^{n} r_i \bmod N$. Then each member U_i in G computes $s_i = w_i + x_i \times h(v_i) \times h(M,R)$ and sends s_i to the clerk. The clerk verifies every individual signature (r_i, s_i) by the equation $g^{s_i} \times (y_i^e + ID_i)^{h(M,R)} \equiv r_i \pmod N$ for i= 1, 2, …, n. If all (r_i, s_i)'s are valid, the clerk computes $S = \Sigma_{i=1}^{n} s_i$ and publishes (R, S) as the multi-signature of M signed by G.

Multi-signature verification phase

A verifier validates (R, S) and M by using the equation $g^S \times (Y^e + GID)^{h(M,R)} \equiv R$

(mod N). If the equation holds, the verifier accepts the validity of the signature (R, S) of M.

In the following, a theorem is given to show why the verification equation can be used to verify multi-signatures.

Theorem If the equation $g^S \times (Y^e + GID)^{h(M,R)} \equiv R \pmod N$ is holds, then the multi-signature of M for G is verified and meanwhile the group public key of G is authenticated.

Proof. We have $S = \Sigma_{i=1}^{n} w_i + \Sigma_{i=1}^{n} x_i \times h(v_i) \times h(M, R)$ and $X = \Sigma_{i=1}^{n} x_i \times h(v_i) \bmod \phi(N)$. Raising both sides of above equation to exponents with base g yields

$$g^S \equiv g^{\sum_{i=1}^{n} w_i + \sum_{i=1}^{n} x_i \times h(v_i) \times h(M, R)}$$

$$\equiv g^{\sum_{i=1}^{n} w_i} \times g^{\sum_{i=1}^{n} x_i \times h(v_i) \times h(M, R)}$$

$$\equiv g^{\sum_{i=1}^{n} w_i} \times g^{X \times h(M, R)} \pmod N$$

Since $g^{\sum_{i=1}^{n} w_i} \equiv \Pi_{i=1}^{n} r_i \equiv R \pmod N$, we have

$$g^S \equiv R \times (g^X)^{h(M, R)}$$

$$\equiv R \times (g^{-X})^{-h(M, R)} \pmod N$$

$$\equiv R \times (Y^e + GID)^{-h(M,R)} \pmod N.$$

Therefore, $g^S \times (Y^e + GID)^{h(M,R)} \equiv R \pmod N$. The equation $g^S \times (Y^e + GID)^{h(M,R)} \equiv R \pmod N$ can be derived from the equation $Y^e + GID \equiv g^{-X} \pmod N$ and $g^S \equiv g^{\sum_{i=1}^{n} w_i} \times g^{X \times h(M, R)} \pmod N$. Thus, (R, S) are verified if Y is authenticated.

Based on the equation $g^S \equiv g^{\sum_{i=1}^{n} w_i} \times g^{X \times h(M,R)} \pmod N$, we have $(Y^e + GID)^{h(M,R)} \equiv (g^{-X})^{h(M,R)} \pmod N$. The equation $Y^e + GID \equiv g^{-X} \pmod N$ can be derived from the equation $(Y^e + GID)^{h(M,R)} \equiv (g^{-X})^{h(M,R)} \pmod N$. The verifier can ensure that Y is indeed associated to the signature of X and GID. That is, Y is authenticated if (R, S)

are verified.

5. SECURITY AND PERFORMANCE ANALYSIS

5.1 Security analysis

The security of the improved scheme is based on the secure one-way hash function h and the two well-known cryptographic assumptions [9, 11]: factorization (FAC) and discrete logarithm (DL) assumptions. In the following, some possible attacks (Attacks 1-9) against the proposed scheme are discussed under the above assumptions.

Attack 1: An adversary may try to reveal the secret key x_i of the user U_i from the corresponding public key y_i

Security analysis of Attack 1: The adversary obtains $v_i = (y_i^e + ID_i) \bmod N$ according to the equation $y_i = (v_i - ID_i)^d \bmod N$. He might try to compute x_i from the equation $v_i \equiv y_i^e + ID_i \equiv g^{-x_i} \pmod N$. However, he will face the FAC and DL assumptions to compute x_i from $v_i \equiv y_i^e + ID_i \equiv g^{-x_i} \pmod N$, as discussed in [4]. So this possible attack fails.

Attack 2: An adversary may try to reveal the secret key x_i of the user U_i from the individual signature (r_i, s_i) and the multi-signature (R, S) of the message M.

Security analysis of Attack 2: The adversary cannot obtain x_i from the equation $s_i = w_i + x_i \times h(v_i) \times h(M, R)$ unless he knows w_i in advance. The adversary should solve w_i by the equation $r_i = g^{w_i} \bmod N$ but the equation is protected by the FAC and DL assumptions as analyzed in Attack 1. In another way, the adversary might directly solve x_i (and w_i) from the equation system formed by the equations $s_i = w_i + x_i \times h(v_i) \times h(M, R)$. Since the number of unknown variables x_i's and w_i's is always greater than the number of equations in the system, the adversary cannot solve x_i or w_i from the equation system.

Attack 3: An adversary tries to impersonate U_i and forges the individual signature (r_i, s_i) for a randomly chosen M without knowing x_i.

Security analysis of Attack 3: There are two possible approaches are adopted to forge a valid individual signature for the message M to pass the signature verification equation $g^{s_i} \times (y_i^e + ID_i)^{h(M, R)} = r_i \pmod N$. In the first approach, the adversary first determines the value of r_i and consequently fixes the value of R. Now he wants to find the value of s_i. However, he faces the FAC and DL assumptions for computing s_i. In the next approach, he may first determine the value of s_i. Since R is the product of the r_i's, it is harder than the first approach to compute r_i from $g^{s_i} \times (y_i^e + ID_i)^{h(M,R)} = r_i \pmod N$ based on the one-way hash function and FAC and DL assumptions.

Attack 4: An adversary tries to reveal the group secret key X from either the group public key Y or the multi-signature (R, S) of M for G with knowing $v_i = g^{-x_i} \bmod N$ of all signers U_i's in the group G.

Security analysis of Attack 4: The adversary might directly solve X by the equation $Y^e + GID \equiv g^{-X} \pmod N$. However, he also faces the FAC and DL assumptions to compute X from $Y^e + GID \equiv g^{-X} \pmod N$, as analyzed in Attack 1. In another way, the adversary might solve X from the equation $S = \Sigma_{i=1}^{n} s_i = \Sigma_{i=1}^{n} w_i + X \times h(M, R)$ derived from the equations $s_i = w_i + x_i \times h(v_i) \times h(M, R)$, $X = \Sigma_{i=1}^{n} x_i h(v_i)$ and $S = \Sigma_{i=1}^{n} s_i$. The equation $S = \Sigma_{i=1}^{n} s_i = \Sigma_{i=1}^{n} w_i + X \times h(M, R)$ implies that the adversary can easily obtain X from the above equation if he knows all w_i's in advance. However, the adversary will face the FAC and DL assumptions to find all w_i's according to the equation $r_i = g^{w_i} \bmod N$, as analyzed in Attack 2.

Attack 5: An adversary tires to forge the multi-signature (R, S) of a message M for the group G without the approval of all signers in the group.

Security analysis of Attack 5: The adversary might plot such attacks by adopting the following two approaches. In the first approach, the adversary first finds the group secret key X from all available public parameters and then universally forges the multi-signature of any message M for G. However, X is protected under the FAC and DL assumptions, as analyzed in Attack 4. So this approach cannot success. The other approach is that the adversary directly computes the multi-signature (R, S) for the message M passing the verification $g^{S\times} (y^{e+} GID)^{h(M,\ R)} \equiv R \pmod N$. According to the same reasons discussed in Attack 3, the adversary cannot forge such R and S without knowing the group secret key X.

Attack 6: Some malicious impostors in the group G try to reveal certain co-signer's secrete key from the individual signatures (r_i, s_i)'s contributed to the multi-signature (R, S) for a given message M.

Security analysis of Attack 6: Without losing generality, let U_1, U_2, …, U_t, U_t ($t< |G|$) be the impostors want to reveal U_{t+1}'s secrete key x_{t+1}. They might plot such attack via directly solving x_{t+1} from the equation $X= \Sigma_{i=1}^{n} x_i h(v_i) \bmod \phi(N)$ or $s_i= w_i+ x_i\times h(v_i)\times h(M, R)$. These impostors cannot obtain x_i from $X= \Sigma_{i=1}^{n} x_i\ h(v_i) \bmod \phi(N)$ unless they know X. Fortunately, X is protected under the FAC and the DL assumptions, as analyzed in Attack 4. So this attack fails. Note that the individual signatures (r_i, s_i)'s are generated by $s_i= w_i+ x_i\times h(v_i)\times h(M, R)$ without any relationship. The impostors cannot solve x_{t+1} from the equation system formed by the equations $s_i= w_i+ x_i\times h(v_i)\times h(M, R)$ even if they present their own secret parameters (i.e., w_i's or x_i's) to each other.

Attack 7: Some malicious impostors in the group G try to forge an individual signature of a given message M for certain co-signer without knowing that

co-signer's secrete key. Then they can forge the multi-signature of M for G.

Security analysis of Attack 7: With the same assumption of Attack 6, the impostors should first give the value of r_{t+1} and then determine s_{t+1} from $g^{s_{t+1}} \times (y_{t+1}^{e} + ID_{t+1})^{h(M, R)} = r_{t+1} \pmod N$ to pass the verification of (r_{t+1}, s_{t+1}) by the clerk. However, it is hard to directly compute s_{t+1} for a given r_{t+1} based on the FAC and the DL assumptions as analyzed in Attack 3.

Attack 8: Some malicious impostors in the group G try to universally forge multi-signature for a given message M, that is rejected to be signed by the other co-signers.

Security analysis of Attack 8: The impostors could succeed such attack by adopting the following two possible ways: (1) first computing X and then forging (R, S) for M, and (2) directly computing (R, S) for M satisfying $g^{S} \times (y^{e} + GID)^{h(M, R)} = R \pmod N$. It is easy to see that the first way does not work, since X is protected by the FAC and the DL assumptions, as analyzed in Attack 4. As to second way, the impostors might first fixing R and the computing S, or first fixing S and then computing R for the verification equation $g^{S} \times (y^{e} + GID)^{h(M, R)} = R \pmod N$. However, these two approaches do not work as analyzed in Attack 5.

Attack 9: Some malicious insider forger in the group G, try to forge multi-signature for a given message M, without knowing x_i.

Security analysis of Attack 9: Without loss of generality, we assume that A_1 wants to perform the insider attack to forgery a multi-signature on M. So he must compute v_1' and x_1' satisfying $v_1'^{h(v_1')} = g^{-x_1'} \times (\Pi_{i=2}^{n} v_i^{h(vi)})^{-1} \pmod N$. The product $\Pi_{i=2}^{n} v_i^{h(vi)}$ is known in advance. Now he has two choices. He may first fixing v_1' and then computing x_1'. Then he face the FAC and DL assumptions for computing x_1' from

the equation $v_1'^{h(v_1')} = g^{-x_1'} \times (\Pi_{i=2}^{n} v_i^{h(v_i)})^{-1} \bmod N$. He may first fixing x_1' and then computing v_1'. He will face the one-way hashing function and DL assumptions to compute v_1' from equation $v_1'^{h(v_1')} = g^{-x_1'} \times (\Pi_{i=2}^{n} v_i^{h(v_i)})^{-1} \bmod N$. Therefore, this attack fails.

According to the above security analysis, the improved scheme is secure even if the adversary adopts our insider attack in Section 3.

5.2 Performance Analysis

The performance evaluation of the proposed multi-digital signature concerns the bit-size of multi-signatures and the computation cost. The following notations are used for analyzing the performance of the proposed schemes:

|N|: the bit-size of modular N.

|G|: the bit-size of group members.

|h|: the output bit-size of a one-way hash function h

T_h: the time for calculating the one-way hash function h once.

T_m: the time for a multiplication without modulo N.

T_{mm}: the time for a multiplication with modulo N.

T_{me}: the time for an exponentiation with modulo N.

Note that the time for addition with or without modulo N is relatively smaller than those of T_m, T_{mm} and T_{me}. The cost of these operations is ignored in the analyses of the time complexities of the proposed schemes.

(A) The bit-size of multi-signatures

(1) The bit-size of an individual signature (r_i, s_i) is bounded by 2|N|

Girault (1991) suggested that any user may choose his secret key x_i, with160 bits, while SA's secret parameters p and q should be with more than 350 bits against the exhaustive search attack from $s_i = w_i + x_i \times h(v_i) \times h(M, r_i)$. According to Secure

Hashing Algorithm [8], the bit-size of one-way hashing function |h| is 160 bits. It is easy to see that if $|w_i|$ is bounded to 480 bits, then the bit-size of s_i will be bounded to |N|. Thus, the size of an individual signature $|r_i|+|s_i|$ is bounded to 2|N| bits.

(2) The bit-size of a multi-signature (R, S) is bounded by |G|+2|N| bits

Since $R= \Pi_{i=1}^{n} r_i \pmod N$, |R| is bounded to |N| bits. Since $S= \Sigma_{i=1}^{n} s_i$ and $|s_i|$ is bounded by |N|, so |S| is bounded to |G|+ |N| bits. Therefore, the size |R|+ |S| of the multi-signature is bounded to |G| +2|N| bits.

(B) Computation cost

In the user registration phase, the SA needs $|G|(2T_{me})$ to compute all individual public keys y_i's according to the equation $y_i= (v_i^{h(v_i)} -ID_i)^d \bmod N$. To compute the group public key $Y= ((\Pi_{i=1}^{n} v_i^{h(v_i)})-GID)^d \pmod N$, the SA needs $(|G|-1)\ T_{mm}+T_{me}$ by storing all of the values $v_i^{h(v_i)}$'s in the equations $y_i= (v_i^{h(v_i)} -ID_i)^d \bmod N$.

Every signer in G requires $T_{me}+ (|G|-1)\times T_{mm}+ 2T_m+ 2T_h$ to generate his individual signature to construction of the multi-signature of M for G. From the equations $r_i= g^{w_i} \bmod N$ and $s_i= w_i+ x_i \times h(v_i)\times h(M,R)$, he needs one T_{me} to compute r_i, $(|G|-1)\times T_{mm}$ to compute R, two T_h, for the calculations of $h(v_i)$ and h(M, R), and two T_m to compute s_i.

The verification cost of an individual signature is $3T_{me}+ T_{mm}+ T_h$ according to the equation $g^{s_i}\times(y_i^e+ ID_i)^{h(M,R)} \equiv r_i \pmod N$. The clerk also needs $(|G|-1)\times T_{mm}$ to compute R. Totally, the computation cost of the clerk is $|G|(3T_{me}+ T_{mm}+ T_h)+ (|G|-1)\times T_{mm}$.

According to the verification equation $g^S\times(Y^e+ GID)^{h(M,R)} \equiv R \pmod N$, the verifier needs $3T_{me}+ T_{mm}+ T_h$ to finish the verification.

6. CONCLUSIONS

An insider attack is first proposed to attack Chang et al's ElGamal-like multi-signature scheme using self-certified public key. By our attack, any malicious singer in the signing group can forge a valid multi-signature for any messages without any secrete keys of the other signers. So Chang et al's scheme is insecure. To guard against the insider attack, an efficient improved multi-signature scheme is also proposed.

REFERENCES

[1] Yuh-Shihng Chang, Tzong-Chen Wu and Shih-Chan Huang (2000): "ElGamal-like digital signature and multisignature schemes using self-certified public key," *The Journal of Systems and Software*, 50, pp. 99-105.

[2] W. Diffie and M. Hellman (1976): "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. IT-22, 1976, pp. 644- 654.

[3] T. ElGamal (1985): "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, Vol. IT-31, No. 4, July 1985, pp. 469- 472.

[4] M. Girault (1991): "Self-certified public key," *In: Advances in Cryptology-EUROCRYPT'91*, Springer-Verlag, Berlin, pp. 491-497.

[5] K. Itakura and K. Nakamura (1983): "A public-key cryptosystem suitable for digital mulitsignatures," *NEC Res. Development* 71, pp.1-8.

[6] Z.C. Li, L.C.K. Hui, K.P. Chow, C.F. Chong, W.W. Tsang and H.W. Chan (2000): "Cryptanalysis of Harn digital multi-signature scheme with distinguished signing authorities," *Electronics Letters*, 17th February 2ooo Vol.36 No.4.

[7] R.L. Rivest, A. Shamir and L. Adleman (1978): "A method for obtaining digital signatures and public-key cryptosystems," *Communications of ACM*, Vol. 21,

No. 2, 1978, pp. 120-126.

[8] William Stallings (1999): *Cryptography and Network Security: Principles and Practice*, 2nd ed., Prentice-Hall, New Jersey, pp. 281-286.

[9] B. Schneier (1996): *Applied Cryptography*, 2nd ed. Wiley, New York.

[10] D.R. Stinson (1995): *Cryptography: Theory and Practice*, CRC Press, Boca Raton, FL.

[11] Hung-Min Sun, Biing-Jang Chen and Tzonelih Hwang (1999): "Cryptanalysis of group signature scheme using self-certified public keys," *Electronics Letters*, 28th October 1999 Vol.35 No.22.