ID-BASED STRUCTURED MULTISIGNATURE SCHEMES

Chih-Yin Lin, Tzong-Chen Wu* and Jing-Jang Hwang
Institute of Information Management, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.
**Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C.*

Abstract: The signing structure of a multisignature scheme specifies the signing order for all signers when signing messages, and any multisignature not obeying the specified signing order will be verified as invalid. In accordance with the different responsibilities of the participant signers, the signing structure of a multisignature scheme could be further classified as the following three types: serial, parallel and mixed, where the mixed structure is regarded as the mix of the serial and the parallel. Based on the well-known ID-based public key system, we will propose three ID-based structured multisignature schemes and each scheme respectively realizes the serial, parallel and mixed signing structures. In the proposed schemes, the length of a multisignature is fixed and the verification of a multisignature is efficient, regardless of the number of signers involved. Besides, any invalid partial multisignature can be effectively identified during the generation of the multisignature.

Keywords: Multisignature, structured multisignature, ID-based public key, signing structure.

1. INTRODUCTION

A multisignature scheme allows multiple signers to sign messages in which all signers have to sign and individual signer's identity can be identified from the multisignature [2-6, 8-10, 13-16, 19]. Furthermore, a structured multisignature scheme [4, 6] is a multisignature scheme that additionally requires all signers to obey a predefined signing structure when signing messages, and any multisignature generated without obeying the specified signing structure will be verified as invalid. The signing structure of a multi-

signature scheme indicates the signing order among all participant signers when signing messages, As a consequence, the multisignature of a message in a structured multisignature scheme is said to be valid when the following conditions are satisfied: (i) All signers had signed the message; (ii) All signers perform their signing operations in compliance with the specified signing structure; (iii) The multisignature and all partial multisignatures generated during the multisignature generation process have been successfully verified. Typical applications of the structured multisignature scheme are multisignatures used in a corporate organization or hierarchical environment. For example, a legitimate working report should be signed accordingly in the order of the operators, the section leader and the department manager. Signing structures can be classified into three basic types: serial, parallel, and mixed, where the mixed structure is the mix of the serial and the parallel. For the serial structure, all signers sign messages in a predetermined sequence, and hence the generated multisignatures are sensitive to the signing order. As to the parallel structure, all signers sign messages in a parallel manner and the generated multisignatures are independent of the signing order. In the mixed structure, the signing structure is composed by substructures that could be serial, parallel, or another mixed structure, and the generated multisignatures are sensitive to the signing order specified in the corresponding signing structure. Figure 1 depicts these three types of signing structures.

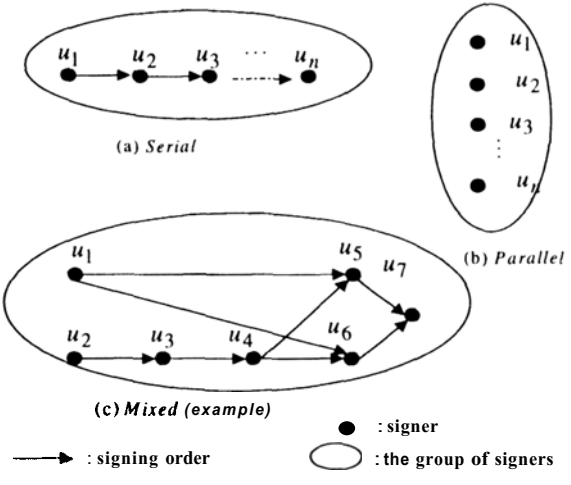

Figure 1 – Three types of the signing structure.

Most of the previously proposed multisignature schemes are irrelevant to signers' signing order, while some others are order-sensitive. The schemes presented in References 2, 3, 5, 8, 13, 14, and 19 are order-irrelevant, and the schemes presented in References 4, 6, 9, 10, 15, and 16 are order-sensitive. Among the order-sensitive schemes, the schemes proposed by Ham and Kielser [9], Itakula and Nakamura [10], and Okamoto [16] are RSA-like multisignature schemes in which the signers' signing order has to be properly arranged by different modules of their public keys; otherwise, messages to be signed might be modularly truncated. Besides, the length of the multisignature and the verification time required by these RSA-like schemes varies proportionally with the amount of the signers participated. In 1998, Doi *et al.* [6] firstly proposed a multisignature scheme considering the mixed signing structure. They used structured group identity and proposed two structured multisignature schemes for common modular RSA-type and ElGamal-type signature schemes. However, the length of the multisignature generated by their schemes varies with the number of the signers involved. Later, Burmester *et al.* [4] proposed an EIGamal-type multisignature scheme with a structured public key approach. In their scheme, the secret and the public keys for each signer could be generated either by a trusted centre or by cooperative signers using a distributed protocol. Moreover, Burmester *et al.* assumed that there exists at least one honest signer for their scheme to be secure. This assumption is somewhat less practical and incompatible especially when applying to a delegation scheme, i.e. proxy signature [11-12], in which the original signer has to consider the threat that all (proxy) signers in the signing structure may commit frauds or collusions.

Based on the well-known ID-based public key systems [7, 18], we propose three structured multisignature schemes whose security is based on the difficulty of solving discrete logarithm modulo a large composite (DLMC) [1] and factorising a large composite (FAC) [l, 17]. Since ID-based digital signature and multisignature schemes [7, 18-19] use the identity of the signer as the public key, our scheme has the advantage that the signature verification requires no extra interaction for public key verification. The proposed schemes have the following merits:

(1) The length of the multisignature is fixed to different messages.
(2) The length of the multisignature is fixed regardless of the number of signers.
(3) The computation cost required for the multisignature is efficiently fixed to the amount of signers participated.
(4) Any violation to the signing order will be detected and identified immediately.

This paper is sketched as follows. After the introduction, we specify the notations, parameters and signing structures in section 2. In section 3, we will propose the serial, the parallel and the mixed structure multisignature schemes. Security analysis is given in section 4. We conclude the paper in section 5.

2. NOTATIONS AND PARAMETERS

Let $G = \{u_1, u_2, ..., u_n\}$ be a group consists of n signers and $sk_i = (sk1_i, sk2_i)$ be u_i's private key. The verification key for the partial multisignatures generated by u_i is named "partial verification key" and denoted by PK_i. The verification key for verifying the multisignatures generated by G is named "public verification key" and denoted by VK_G.

For each signer $u_i \in G$, his identity ID_i is a message digest of his/her public identification information I_i using a one-way hash function, said F, such that $ID_i = F(I_i)$. As defined in other ID-based crypto schemes [7, 18-191, I_i can be a combination of u_i's name, age, gender, telephone number or home address, provided that this combination can uniquely identify u_i. Note that a system authority *SA* is assumed [7, 18-19] for setting up the ID-based cryptosystem.

2.1 Signing Structure

Two types of notations are used for describing the signing structures. SER[ ] denotes the serial structure; and PAR[ ] the parallel structure. For $G_1 = \{u_1, u_2, u_3\}$, if the legal signing sequence is $< u_1, u_2, u_3 >$, then the corresponding signing structure is $\mathrm{SER}[u_1, u_2, u_3]$. Another example is, for $G_2 = \{u_1, u_2, u_3, u_4\}$ with a mixed signing structure SER[u_1, PAR[u_2, u_3],u_4], there are exactly two legal signing sequences, which are $< u_1, u_2, u_3, u_4 >$ and $< u_1, u_3, u_2, u_4 >$. Furthermore, we can use a diagram to represent the corresponding signing structure as in Figure 1 and 2. In the diagram, each node indicates a signer and each arrow implies the signing order for the two signers it connects. If an arrow points from u_i to u_j, it means u_j should sign after u_i signs. In the above example group of signers G_2, it can be draw as in Figure 2(a). Notably, in order to facilitate the tasks performed in the structured multisignature schemes described later, we add two dummy nodes *s* and *t* to the diagram representation where *s* and *t* denote the start node and terminate node, as shown in Figure 2(b). The general diagrams for a group of serial signers, a group of parallel signers and an

example diagram for a group of mixed-structure signers are shown (i.e. $\mathrm{SER}[\mathrm{PAR}[u_1,\mathrm{SER}[u_2,u_3,u_4]],\mathrm{PAR}[u_5,\ u_6],u_7]$) respectively in Figure 1.

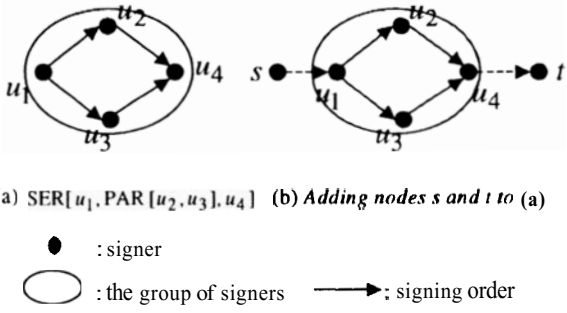

Figure 2 -The diagrams of an example signing structure.

2.2 System Parameters

SA initialises the ID-based public key system applicable for structured multisignatures by first preparing the following parameters.

p, q : two large prime integers, where $2p+1$ and $2q+1$ are also primes.
N : the product of $2p+1$ and $2q+1$ that $N=(2p+1).(2q+1)$.
w : the product of p and q that $w=p.q$.
a : a base of order w modulo N.
r : a random number, where $r\in Z_{\omega}^{*}$.
β : $\beta=\alpha^{r} \bmod N$.
f,h : two hash functions, where $f(x)\in \min(p,q)$ and $h(x)<\min(p,q)$.

SA keeps p, q, w and r secret, while publishing N, a, β f and h. Note that f is used to generate the public identities and verification keys and h is used to produce the message digest of the message to be signed. Throughout this paper, x^{-1} denotes the inverse of x modulo w.

2.3 Public Verification Keys

SA generates VK_G for G and PK_i for each $u_i \in G$ by the following rules. For serial structure, signers' public identities, i.e. ID_i's, are concatenated, and for parallel structure, signers' public identities are first sorted then

concatenated, to be the input of the function f. The output of f is the value of the public verification key.

Notice that the reason why we sort and concatenate the identities of signers in the parallel structure is to provide uniqueness of the verification key. Consequently, the possibility of the existence of two identical verification keys can be eliminated. To achieve this, we can use a function, said $\zeta(\ )$, that takes a variant numbers of values as input, sorts the input values, and finally outputs the value of the concatenation of the sorted input values. For example, the output of $\zeta(2983,9213,7615,1003,8714)$ will be 10032983761987149213. For $G=\{u_1,u_2,u_3,u_4\}$ and its signing structure $\mathrm{SER}[u_1,\mathrm{PAR}[u_2,u_3],u_4]$, $VK_i = f(ID_1 \| \zeta(ID_2,ID_3) \| ID_i)$, $PK_i = f(ID_1)$, $PK_i = f(ID_1 \| ID_i)$, $PK_i = f(ID_1 \| \mathrm{ID}_i)$, and $PK_i = VK_G = f(ID_1 \| \zeta(ID_2,ID_3) \| ID_i)$.

3. THE PROPOSED MULTISIGNATURE SCHEMES

The multisignature schemes for serial, parallel and mixed signing structures are presented respectively. Each proposed scheme consists of three phases: key generation, multisignature generation and multisignature verification. In key generation phase, the system authority generates the private key for each signer. In the multisignature generation phase, each signer follows the signing structure to sign messages after verifying the partial multisignatures generated by the preceding signers. Finally in the multisignature verification phase, the verifier verifies the validity of the multisignature. Details are given in the followings.

3.1 For serial signing structure

Without loss of generality, assume the group of signers $G=\{u_1,u_2,...,u_n\}$ is associated with the signing structure $\mathrm{SER}[u_1,u_2,...,u_n]$. That is, all $u_i \in G$ have to sign messages by following the serial order $u_1,u_2,...,u_n$ for generating a valid multisignature. The scheme is stated as follows.

Key generation phase:

SA prepares the partial verification keys and public verification key as $PK_i = f(ID1 \| ID_i \| ... \| ID_i)$, for $u_i \in G$, $i=1,2,...,n$, and $VKG = PKn$. Then, he performs the following operations:

Step 1. Compute k_i by the following equation, for $i=1,2,...,n$.

$$k_i = PK_i^{-1} \cdot r \bmod \omega . \tag{1}$$

Step 2. Select a random number k_0, where $k_0 \in Z_\omega^*$.
Step 3. Randomly select $sk1_i \in Z_\omega^*$, for $i = 1,2,...,n$.
Step 4. Calculate $sk2_i$ by the equation below, for $i = 1,2,...,n$.

$$sk2_i = k_i - k_{i-1} \cdot sk1_i \bmod \omega \tag{2}$$

Step 5. Securely distribute $sk_i = (sk1_i, sk2_i)$ to $u_i \in G$, for $i = 1,2,...,n$
Step 6. Compute and deliver $w = \alpha^{k_0} \bmod N$ to u_1.

Multisignature generation phase:

Suppose $G = \{u_1, u_2, ..., u_n\}$, with signing structure $SER[u_1, u_2, ..., u_n]$, want to generate *a* multisignature *MS* for a message *m*. Each u_i, for $i = 1,2,...,n$, performs the signing operations as below.

Step 1. Verify the partial multisignature S_{i-1} signed by u_{i-1}, (for $i \neq 1$) by testing if

$$(S_j)^{PK_j} = \beta^{h(m)} (\bmod N) . \tag{3}$$

(If the test fails, then the signing process is stopped and u_{i-1} is reported as a malicious signer.)

Step 2. Compute the partial multisignature S_i by

$$S_i = S_{i-1}^{sk1_i} \cdot \alpha^{h(m) \cdot sk2_i} \bmod N , \tag{4}$$

where S_{i-1} is generated by u_{i-1} and $S_0 = w^{h(m)} \bmod N$.

Step 3. Send S_i to u_{i+1}, for $i < n$.

The partial multisignature S_n generated by the last signer u_n is treated as the multisignature *MS* generated by *G* with $SER[u_1, u_2, ..., u_n]$ for message *m*.

Multisignature verification phase:

The multisignature *MS* of message *m* that signed by the signing group *G* with signing structure $SER[u_1, u_2, ..., u_n]$ can be publicly verified by VK_G as:

$$(MS)^{VK_G} = \beta^{h(m)} (\bmod N) \tag{5}$$

LEMMA 1. For any message m and its partial multisignature S_i generated by $u_i \in G$, $S_i = \alpha^{k_i \cdot h(m)} \pmod N$ in the serial approach.

Proof:

Multiplying $h(m)$ to Equation 2 and raising both sides of it to exponents with base a, it yields a recursive relation $\alpha^{k_i \cdot h(m)} = \alpha^{k_{i-1} \cdot h(m) \cdot sk1_i} \cdot \alpha^{h(m) \cdot sk2_i} \pmod N$, where the $k_0 \in Z_\omega^*$ is randomly chosen. By the above fact and $\alpha^{k_0 \cdot h(m)} = w^{h(m)} = S_0 \pmod N$, we can conclude $S_i = \alpha^{k_i \cdot h(m)} \pmod N$ by mathematical inductions. *Q.E.D.*

THEOREM 1. If all $u_i \in G$ honestly sign the message m by following $SER[u_1, u_2, ..., u_n]$, then the generated multisignature MS will be successfully verified by Equation 5.

Proof:

Recall that $MS = S_n$ and $VK_G = PK_n$. By Equation 1 and Lemma 1 we can obtain Equation 5

$$(MS)^{VK_G} = \alpha^{k_n \cdot h(m) \cdot VK_G} = \alpha^{r \cdot PK_n^{-1} \cdot h(m) \cdot VK_G} = \beta^{h(m)} \pmod N. \qquad \text{Q.E.D.}$$

THEOREM 2. Any disorder signing operation regarding $SER[u_1, u_2, ..., u_n]$ will be identified with the probability of $(\omega - 1)/\omega$.

Proof:

By following $SER[u_1, u_2, ..., u_n]$, u_i should sign the partial multisignature S_{i-1} generated by u_{i-1} for message m after verifying S_{i-1}'s validity. Assume a disorder operation takes place before u_i signs, whether by mistake or intentionally, that u_j, where $i < j \le n$, signs S_{i-1} instead of u_i. Then, the partial multisignature generated hereby will be

$$S_i' = S_{i-1}^{sk1_j} \cdot \alpha^{h(m) \cdot sk2_j} \pmod N.$$

For S_i' to be successfully verified by Equation 5, it has to satisfy that $S_i' = S_i \pmod N$, which implies

$$S_{i-1}^{sk1_j} \cdot \alpha^{h(m) \cdot sk2_j} = S_{i-1}^{sk1_i} \cdot \alpha^{h(m) \cdot sk2_i} \pmod N. \qquad (6)$$

By Lemma 1, the exponent part of Equation 6 indicates

$$k_{i-1} \cdot sk1_j + sk2_j = k_{i-1} \cdot sk1_i + sk2_i \pmod \omega. \qquad (7)$$

In order for S_i' to be valid, two distinct private keys (i.e. sk_i and sk_j) have to satisfy Equation 7. Since the values of $sk1_i$'s for all $u_i \in G$ are randomly selected and $sk2_i$'s are computed from Equation 2, it is to see that the probability for sk_i and sk_j to satisfy Equation 7 is $1/\omega$. Therefore, the probability for successfully identifying a disorder event is $(\omega-1)/\omega$. Q.E.D.

3.2 For parallel signing structure

Let $G=\{u_1,u_2,...,u_n\}$ be a group of signers with signing structure $PAR[u_1,u_2,...,u_n]$. The scheme is stated as follows.

Key Generation Phase:

SA prepares the partial verification keys and public verification key as $PK_i = f(ID_i)$, for $u_i \in G$, $i=1,2,...,n$, and $VK_G = f(\zeta(ID_1, ID_2,..., ID_n))$. Then, he performs the following operations:

Step 1. Compute k_i by the following equation, for $i = 1,2,\ldots,n$.

$$k_i = PK_i^{-1} \cdot r \bmod \omega . \tag{8}$$

Step 2. Select a random number k_0, where $k_0 \in Z_\omega^*$.

Step 3. Randomly select $sk1_i \in Z_\omega^*$ for $i=1,2,...,n$.

Step 4. Calculate the value of $sk2_i$ as follows, for $i=1,2,...,n$.

$$sk2_i = k_i - k_0 \cdot sk1_i \bmod \omega .$$

Step 5. Securely distribute sk, $=(sk1_i, sk2_i)$ to $u_i \in G$, for $i=1,2,...,n$.

Step 6. Calculate the value of v by

$$v = \alpha^{VK_G^{-1} \cdot r - \sum_{i=1}^{n} k_i} \bmod N . \tag{9}$$

Step 7. Compute $w = \alpha^{k_0} \bmod N$ and deliver w, v to all $u_i \in G$.

Multisignature generation phase:

Suppose the signing group G with signing structure $PAR[u_1,u_2,...,u_n]$ wants to generate the multisignature MS for message m. Each $u_i \in G$ performs the following tasks without concerning other signer's signing order.

Step 1.Compute the partial multisignature S_i as

$$S_i = (w^{h(m)})^{sk1_i} \cdot \alpha^{h(m) \cdot sk2_i} \bmod N .$$

Step 2. Send S_i to all $u_j \in G$, for $j \neq i$.
Step 3. Verify S_j sent from u_i, for $j \neq i$, by testing if

$$(S_j)^{PK_j} = \beta^{h(m)} (\text{mod } N). \tag{10}$$

(If the test fails, then the signing process is stopped and u_j is reported as a malicious signer.)

Step 4. Calculate the multisignature MS after receiving and verifying all S_j's, for $j \neq i$, $u_j \in G$, by the following equation.

$$MS = v^{h(m)} \cdot (\prod_{i=1}^{n} S_i) \bmod N. \tag{11}$$

Multisignature verification phase:

The multisignature MS, generated by the signing group G with signing structure $\text{PAR}[u_1, u_2, \ldots, u_t]$, for message m can be verified by testing if

$$(MS)^{VK_G} = \beta^{h(m)} (\text{mod } N). \tag{12}$$

THEOREM 3. If all $P_i \in G$, for $i = 1,2,\ldots, n$, honestly sign the message m by following $\text{PAR}[u_1, u_2, \ldots, u_t]$, then the multisignature MS generated by G will be successfully verified by Equation 12.

Proof:

Based on the fact that all valid partial multisignatures can be successfully verified by Equation 10, we can rewrite Equation 10 with Equation 8 as

$$S_i = \alpha^{k_i \cdot h(m)} (\text{mod } N). \tag{13}$$

Then, from Equation 9, 11 and 13, we can obtain that

$$
\begin{aligned}
MS &= v^{h(m)} \cdot (\prod_{i=1}^{n} S_i) (\mathrm{mod}\, N) \\
&= (\alpha^{VK_G^{-1} \cdot r - \sum_{i=1}^{n} k_i})^{h(m)} \cdot (\prod_{i=1}^{n} \alpha^{k_i})^{h(m)} (\mathrm{mod}\, N) \\
&= (\alpha^{VK_G^{-1} \cdot r - \sum_{i=1}^{n} k_i} \cdot \alpha^{\sum_{i=1}^{n} k_i})^{h(m)} (\mathrm{mod}\, N) \\
&= \alpha^{VK_G^{-1} \cdot r \cdot h(m)} (\mathrm{mod}\, N) \\
&= \beta^{VK_G^{-1} \cdot h(m)} (\mathrm{mod}\, N).
\end{aligned}
$$

This implies a valid *MS* will be successfully verified by Equation 12. Q.E.D.

3.3 For mixed signing structure

Assume $G = \{u_1, u_2, ..., u_n\}$ is a group consists of mixed-ordered signers. In any real case, the partial verification keys PK_i's and the public verification key VK_G can be easily computed by following the rules described in section 2.2. The diagram representation of the signing structure is employed here to facilitate the key generation and multisignature generation phases.

A new notation used here is $prev(x)$, where x is a node in the signing structure diagram and $prev(x)$ indicates the set of nodes that directly connect and point to node x in the diagram.

Key generation phase:

By observing the diagram of the signing structure of G, SA first prepares the partial verification keys and public verification key, and then generates the secret key sk; for each $u_i \in$ G as follows.

Step 1. Compute k_i by $k_i = PK_i^{-1} \cdot r \bmod \omega$, for $i = 1,2,...,n$.

Step 2. Select a random number k_s, where $k_s \in Z_\omega^*$.

Step 3. Randomly select $sk1_i$, such that $sk1_i \in Z_\omega^*$, for $i = 1,2,...,n$.

Step 4. Calculate the value of $sk2_i$ for each $u_i \in$ G as follows.

If $prev(u_i) = \{s\}$, then $sk2_i = k_i - k_s \cdot sk1_i \bmod w$;

Otherwise,

$$sk2_i = k_i - (\sum_{u_j \in prev(u_i)} k_j) \cdot sk1_i \bmod w$$

Step 5. Distribute $sk_i = (sk1_i, sk2_i)$ to each $u_i \in$ G via a secure channel.

Step 6. Calculate a value v as follows.
If $|prev(t)|=1$, then $v=1$;
Otherwise,

$$v=\alpha^{VK_G^{-1}\cdot r-\sum_{u_j\in prev(t)}k_j} \bmod N .$$

Step 7. Compute and deliver $w=\alpha^{k_0} \bmod N$ to all u_i, for $prev(u_i)=\{s\}$
Step 8. Send v to all $u_i \in G$.

Multisignature generation phase:

Supposing the signing group $G=\{u_1,u_2,...,u_n\}$ with a mixed signing structure wants to generate a multisignature MS for a message m. Then, with the help of the corresponding diagram, each $u_i \in \mathbf{G}$ performs the following operations to compute and distribute the partial multisignature.

Step 1. Compute the partial multisignature S_i as:
For u_i with $prev(P_i)=\{s\}$,

$$S_i=(w^{h(m)})^{sk1_i}\cdot\alpha^{h(m)\cdot sk2_i} \bmod N ; \text{ and,}$$

for u_i with $prev(u_i)\neq(\mathbf{s})$,

$$S_i=(\prod_{u_j\in prev(u_i)}S_j)^{sk1_i}\cdot\alpha^{h(m)\cdot sk2_i} \bmod N ,$$

where S_j is the partial multisignature generated by $u_j \in prev(u_i)$.

Step 2. Distribute S_i to all $u_j \in G$, for $prev(u_j)\supseteq\{u_i\}$, and to all for $\{u_i,u_k\}\subseteq prev(t)$.

Afterwards, the multisignature MS for message m can be calculated by any $u_k \in prev(t)$ with the following equation:

$$MS=v^{h(m)}.(\prod_{u_k\in prev(t)}S_k) \bmod N .$$

Note that before u_i signs, he should have verified the validity of each received partial multisignature S_j for $u_j \in prev(u_i)$ by testing if

$$(S_j)^{PK_j}=\beta^{h(m)} (\bmod N).$$

If the test fails, the signing process is stopped and the corresponding u_j is reported as a malicious signer.

Multisignature verijication phase:

The multisignature *MS,* generated by the signing group *G* for the message *m* can be publicly verified as below.

$$(MS)^{VK_G} = \beta^{h(m)} (\mathrm{mod}\, N) .$$

4. SECURITY ANALYSIS

Possible attacks to the proposed schemes include the attempts to disclose the signer's private key and to forge a structured multisignature. Although the proposed schemes solve three different signing structures, they adopted the same techniques for key generation, multisignature generation and multisignature verification. Hereby we will show that the proposed schemes are secure against these attacks by focusing our discussion on the serial approach. Note that the security of the proposed schemes relies on the difficulty of solving discrete logarithm modulo a large composite (DLMC) [1] and factorising a large composite (FAC) [1, 17].

ATTACK 1. *An attacker attempts to reveal the secret key* $sk_i = (sk1_i, sk2_i)$ *of a signer* $u_i \in$ G *from all available public information.*

Analysis :

From Equation 1 and 2, it is to see that the secret key sk_i of u_i would be disclosed by the attacker only when he knows either the values of ω , *r,* and all PK_i^{-1}'s; or the values of ω and all k_i's. However, given all public information α , β , N and all $PK_{G,i}$'s for $u_i \in$ G , computing ω from N is a problem of FAC intractability and deducing *r* from β is a difficulty of solving the problem of DLMC. In addition, the attacker may try to deduce the value of k_i from the result in Lemma 1, i.e. $S_i = \alpha^{k_i \cdot h(m)} (\mathrm{mod}\, N)$. However, he will obviously face the problem of the DLMC intractability. □

ATTACK 2. *An attacker attempts to reveal the private key ski =* $(sk1_i, sk2_i)$ *of a signer* $u_i \in G$ *from the partial multisignatures* S_i*'s (for all* $u_i \in G$*) of a message m.*

Analysis:

Given $h(m), a$, β , N and all S_i's for $u_i \in G$, directly computing $(sk1_i, sk2_i)$ from Equation 4 in the serial ordered multisignature scheme is an intractability of the DLMC problem. On the other hand, solving $(sk1_i, sk2_i)$ from Equation 2 is also infeasible since w and all k_i for $P_i \in$ G are secret parameters and known only to *SA*. □

ATTACK 3. *An attacker attempts to directly forge a valid multisignature for some message m for the signing group* $G = \{u_1, u_2, ..., u_n\}$.

Analysis

Since the private key of each $u_i \in$ G is securely kept, an attacker cannot create any partial multisignature or multisignature for some message m via Equation 4. Moreover, we know that a forged multisignature has to satisfy Equations 5 to be valid. However, with public information N, β, VK_G and $h(m)$, it's obviously that the attacker will face the FAC problem to directly solve *MS* from Equations 5. □

5. CONCLUSIONS

In this paper, we have addressed a new approach to multisignature schemes that applicable for various signing structures based on ID-based public keys. In addition to enforce the requirement that all signers in the signing group have to follow the predefined signing structure when generating a multisignature, our scheme has the merits that both the length of multisignature and the computation effort for multisignature verification are fixed and independent to the amount of signers. Due to the intractability of the DLMC problem and the FAC problem, the proposed scheme is secure against the deduction of the signer's secret key and forgery to the multisignature.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous referees for their useful comments on improving our paper. This work was partially supported by the Ministry of Education, Taiwan, Program of Excellence Research 90-E-FA04-1-1.

REFERENCES

[1] L.M. Adleman and K.S. McCurley, "Open problems in number-theoretic complexity, II", *Proc. First Algorithmic Number Theory Symposium,* Springer-Verlag, 1994, pp.291-322.

[2] C. Boyd, "Digital Multisignatures". *IMA Conference on Cryptography and Coding,* Oxford University Press, 1989, pp. 241-246.

[3] C. Boyd, "Multisignatures based on Zero Knowledge Schemes", *Electronics Letters.* 27 (22), October 1991, pp. 2002-2004.

[4] M. Burmester. Y. Desmedt, H. Doi, M. Mambo, E. Okamoto, M. Tada and Y. Yoshifuji, "A structured EIGamal-type multisignature scheme", *Proc. Workshop on Practice and Theory in Public Key Cryptosystems,* LCNS 1751, Springer-Verlag, 2000. pp. 466-483.

[5] Y.S. Chang, T.C. Wu and S.C. Huang, "EIGamal-like digital signature and multisignature schemes using self-certified public keys", *The Journal of Systems and Software,* 50 (2). 2000, pp. 99-105.

[6] H. Doi, E. Okamoto and M. Mambo, "Multisignature schemes for various group structures", *The 36-th Annual Allerton Conference on Communication, Control, and Computing,* 1999, pp. 713-722.

[7] A. Fiat and A. Shamir, "How to prove yourself: practical solution to identification and signature problems". *Advances in Cryptology - CRYPTO'86,* Springer-Verlag, 1987, pp. 186-194.

[8] T. Hardjono and Y. Zheng, "A practical digital multisignature scheme based on discrete logarithms". *Advance in Cryptology-A USCRYPT'92,* Springer-Verlag, 1993, pp. 122-132.

[9] L. Ham and T. Kielser, "New scheme for digital multisignatures", *Electronics Letters,* 25 (15), 1989, pp. 1002-1003.

[10] K. ltakura and K. Nakamura, "A public-key cryptosystem suitable for digital multisignature", *NEC Research and Development,* Vol. 71, October 1983, pp. 1-8.

[11] M. Mambo, K. Usuda and E. Okamoto, "Proxy signatures: Delegation of the power to sign messages", *IEICE Tran. Fundamentals,* E97-A (9), 1996, pp. 1338-1353.

[12] M. Mambo, K. Usuda and E. Okamoto, "Proxy signatures for delegating signing operation", *Proc. Conf on Computer and Comm. Security,* ACM press 1996, pp. 48-57.

[13] K. Ohta, S. Micali and L. Reyzin, "Accountable-subgroup Multisignatures", *Manuscript,* Massachusetts Institute of Technology, Cambridge, MA, USA, Nov. 2000.

[14] K. Ohta and T. Okamoto, "A digital multisignature scheme based on the Fiat-Shamir scheme", *Advance in Cryptology-ASIACRYPT'91,* Springer-Verlag, 1993, pp. 139-148.

[15] K. Ohta and T. Okamoto, "Multisignature schemes secure against active insider attacks", *IEICE Transactions on Fundamentals,* E82-A (1), 1999, pp. 21-31.

[16] T. Okamoto, "A digital multisignature scheme using bijective public-key cryptosystems", *ACM Tran. Computer Systems,* 6 (8), 1988, pp. 432-441.

[17] R.L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems", *Comm. of the ACM,* 21 (2), 1978, pp. 120-126.

[18] A. Shamir, "Identity-based cryptosystems and signature schemes", *Advances in Cryprology - CRYPTO'84,* Springer-Verlag, 1985, pp. 47-53.

[19] T.C. Wu, S.L. Chou and T.S. Wu, "Two ID-based multisignature protocols for sequential and broadcasting architectures", *Computer Comm.* 19 (9-10). 1996, pp. 851-856.