

Short Paper

Cryptanalysis and Improvement on a Threshold Proxy Signature Scheme^{*}

ZUO-WEN TAN^{1,2} AND ZHUO-JUN LIU³

¹*College of Information Management
Jiangxi University of Finance and Economics
Nanchang, Jiangxi Province 330013, P.R. China*

²*State Key Laboratory of Information Security
Institute of Software of CAS
Beijing 100190, P.R. China*

³*KLMM, AMSS of Chinese Academy of Sciences
Beijing 100190, P.R. China*

In a (t, n) threshold proxy signature scheme, an original signer can delegate the signature authority to a proxy group of n member such that t or more than t proxy signers can cooperatively sign messages on behalf of the original signer, but $t - 1$ or fewer proxy signers cannot generate a valid proxy signature. In this paper, we review the security of C. L. Hsu *et al.*'s threshold proxy signature schemes with known signers. We show that the threshold proxy signature scheme is insecure against forgery attack. C. L. Hsu *et al.*'s threshold proxy signature scheme is universally forgeable. A new improvement scheme is proposed. The new scheme remedies its weakness. Our proposed threshold proxy signature is secure against chosen message attacks and chosen warrant attacks in the random oracle model under DL assumption.

Keywords: proxy signature, threshold signature, forgery attack, universally forgeable, random oracle model, DL assumption

1. INTRODUCTION

The concept of proxy signature was first introduced by Mambo, Usuda and Okamoto [1]. A proxy signature scheme allows an entity, called the original signer, to delegate another entity, called a proxy signer to sign the message on behalf of the original signer. When the verifier validates a proxy signature, he or she is convinced of that the signature is signed by the proxy signer who is authorized by the original signer. For a secure proxy signature, only the proxy signer can create a valid proxy signature and anyone else, even the original signer, can not generate a valid proxy signature. Thus, for a valid proxy signature, the actual proxy signer cannot deny that he/she has signed the message and the

Received May 16, 2007; revised September 13, 2007; accepted June 12, 2008.

Communicated by Tzong-Chen Wu.

^{*} This paper was partially supported by the Science Research Fund of Jiangxi Province Education Department (No.273), the National Natural Science Foundation of China (No.10701040) and National Key Technology R&D Program (No. 2006BAJ05A01).

original signer cannot deny that he/she has delegated the signing authority to the actual proxy signer. Proxy signatures have found various practical applications, particularly in the distributed environment, such as e-cash systems [2], global distribution network [3], grid computing [4] and mobile agent applications [5].

In order to make the proxy signature scheme applicable in the group environment, Zhang *et al.* and Kim *et al.* proposed the first threshold proxy signature schemes [6, 7]. In a (t, n) threshold proxy signature scheme, the original signer can delegate the signing power to n proxy signers such that any t or more proxy signers can cooperatively sign messages on behalf of the original signer, but $t - 1$ or fewer proxy signers cannot generate a valid proxy signature. Since then, many threshold proxy signature schemes are proposed. A secure (t, n) threshold proxy signature scheme should hold the security properties: secrecy, proxy protection, unforgeability, nonrepudiation, time constraint and known signers [8].

In 1999, Sun first proposed a nonrepudiable threshold proxy signature scheme with known signers [9] based on Zhang's threshold proxy signature scheme [6]. Sun's scheme eliminates Kim *et al.*'s scheme's disadvantage that the verifier is unable to determine whether the proxy group key is generated by the legal proxy group. However, C. L. Hsu *et al.* showed that in Sun's signature scheme, the proxy signers might change the threshold value [10]. In order to defeat the weakness, C. L. Hsu *et al.* proposed an improved threshold proxy signature scheme. Unfortunately, Tan *et al.* pointed out that their modified scheme is also insecure [11].

Recently, Hwang *et al.* propose a nonrepudiable threshold proxy signature scheme with known signers [12]. C. L. Hsu and T. S. Wu showed Hwang *et al.*'s threshold proxy signature scheme is still vulnerable by the collusion attack [13]. Any t or fewer than t malicious proxy signers can still collusively forge valid proxy signatures. Furthermore, C. L. Hsu and T. S. Wu proposed an efficient nonrepudiable threshold proxy signature scheme against the collusion attack [13]. Though C. L. Hsu and T. S. Wu claimed that the proposed scheme improved the security of Hwang *et al.*'s threshold proxy signature scheme and achieved the nonrepudiation requirement, our analysis indicates that the scheme can not resist the universal forgery. For simplicity, we call their proxy signature scheme the HW scheme hereafter. Finally, we propose a new improvement on the HW scheme. The proposed threshold proxy signature is secure against chosen message attacks and chosen warrant attacks in the random oracle model.

The rest of this paper is organized as follows. In section 2, we briefly review the HW scheme. In section 3, we present our cryptanalysis on the HW scheme. The improvement on the HW scheme is given in section 4. In section 5, some discussion and analysis will be made. Finally, section 6 is dedicated to our conclusion.

2. BRIEF REVIEWS OF HW SCHEME

In this section, we briefly review the HW Scheme [13]. The scheme can be divided into four phases: the initialization phase, the proxy share phase, the proxy signature generation phase and the proxy signature verification phase.

2.1 Initialization Phase

Let p be a large prime, q a large prime divisor of $p - 1$, g an element with order q in GFp and $H()$ a secure hash function. These parameters p, q, g are public. Suppose that O be the original signer and G the proxy group which contains n proxy signers $\{P_1, P_2, \dots, P_n\}$. By ID_O and ID_i , we denote the identifiers of the original signer O and the proxy signer P_i ($i = 1, 2, \dots, n$) respectively, where $ID_O \in Z_q$ and $ID_i \in Z_q$ ($i = 1, 2, \dots, n$). The original signer O posses its public/private key pair (y_o, x_o) , here $y_o = g^{x_o} \text{ mod } p, x_o \in Z_q$. Each signer P_i owns its private key $x_i \in Z_q$ and public key $y_i = g^{x_i} \text{ mod } p$. All the keys are certified by the CA. Let m_ω be a warrant which specifies the identities ID_O of the original signer O and ID_i of the proxy signers P_i in the proxy group G , the threshold value t and the delegation time, etc.

2.2 Proxy Share Phase

The proxy share phase is subdivided into three stages.

• Proxy Sharing

All the proxy signers in the group G cooperately generate their secret shares by applying Pedersen’s verifiable secret sharing scheme [14]. First, the proxy signer P_i randomly chooses a $(t - 1)$ -degree polynomial

$$f_i(x) = \sum_{l=1}^{t-1} a_{il}x^l + (ID_i + a_{i,0}) \text{ mod } q, \tag{1}$$

where $a_{i,l} \in Z_q$ and $l = 0, 1, 2, \dots, t - 1$. P_i makes the value $A_{il} = g^{a_{il}} \text{ mod } p$ public and sends $f_i(ID_j)$ to all the other proxy signers P_j in the proxy group G via a secure channel. P_i determines the validity of the shared value $f_j(ID_i)$ by checking whether the following equality holds:

$$g^{f_j(ID_i)} = \left(\prod_{l=0}^{t-1} A_{jl}^{ID_i^l} \right) y_j \text{ mod } p. \tag{2}$$

When $n - 1$ shared values $f_j(ID_i)$ are valid, the proxy signer P_i computes the public value A_l for $l = 0, 1, 2, \dots, t - 1$ and the secret share γ_i :

$$A_l = \prod_{j=1}^n A_{jl} \text{ mod } p, \quad \gamma_i = \sum_{j=1}^n f_j(ID_i) \text{ mod } q. \tag{3}$$

• Proxy Signature Key Generation

The original signer O randomly chooses an integer $k \in Z_q$ and computes the public value K and the proxy signature key σ

$$K = g^k \text{ mod } p, \sigma = k + x_o H(m_\omega \| K) \text{ mod } q. \tag{4}$$

- Proxy Share Generation

The original signer O shares proxy signature key σ in the proxy signer group G by performing the following steps. O randomly chooses a $(t - 1)$ -degree polynomial as

$$f_o(v) = \sum_{j=1}^{t-1} b_j v^j + \sigma \pmod{q}, \quad (5)$$

where $b_j \in Z_q$ and $j = 1, 2, \dots, t - 1$. Next, the original signer O delivers $\sigma_i = f_o(v_j) \pmod{q}$ to each P_i in the proxy group G via a secure channel and broadcasts $m_o, K, B_j \in g^{b_j} \pmod{p}$ to the proxy group.

Each P_i in the proxy group validates the proxy share values σ_i by checking whether the following equality holds:

$$g^{\sigma_i} = y_o^{H(m_o \| K)} K \prod_{j=1}^{t-1} B_j^{ID_i^j} \pmod{p}. \quad (6)$$

When the proxy share value σ_i is valid, the proxy signer P_i computes the proxy signature key:

$$\sigma'_i = \sigma_i + \gamma_i H(m_o \| K) \pmod{q}. \quad (7)$$

2.3 Proxy Signature Generation Phase

For convenience, let $D = \{P_1, P_2, \dots, P_t\}$ be the actual proxy group to sign a message m , ASID the collection of identities of all proxy signers in the actual proxy group D . In order to generate the proxy signature on message m , the actual proxy group D performs the following steps.

- Proxy Sub-signature Generation

Each P_i in the group D randomly chooses $k_i \in Z_q$, computes and broadcasts $r_i \in g^{k_i} \pmod{p}$ in the proxy signer group D . After receiving $r_j \in g^{k_j}$ ($j = 1, 2, \dots, t - 1, j \neq i$), P_i computes

$$R = \prod_{j=1}^t r_j \pmod{p}, \quad L_i = \prod_{j=1, j \neq i}^t (-ID_j)(ID_i - ID_j)^{-1} \pmod{q}, \quad (8)$$

$$s_i = k_i R + (L_i \sigma'_i + x_i) H(A_0 \| R \| ASID \| m) \pmod{q}, \quad (9)$$

and then sends the proxy sub-signature s_i to a designated clerk.

- Proxy Signature Generation

The clerk first computes

$$Y = \prod_{j=1}^n y_j \pmod{p}, \quad A = \prod_{j=1}^{t-1} A_j^{ID_i^j}, \quad B = \prod_{j=1}^{t-1} B_j^{ID_i^j}. \quad (10)$$

Next, the clerk validates the proxy sub-signature s_i by checking whether the equality holds:

$$g^{s_i} = r_i^R (y_i ((y_o Y A_o A)^{H(m_\omega \| K)} KB)^{L_i})^{H(A_o \| R \| ASID \| m)} \pmod p. \tag{11}$$

If all the equations hold, the clerk calculates $S = \sum_{j=1}^t s_j \pmod q$.

Then, $(R, S, K, A_o, m_\omega, ASID)$ is the threshold proxy signature of message m .

2.4 Proxy Signature Verification Phase

On receiving the proxy signature $(R, S, K, A_o, m_\omega, ASID)$, the verifier can identify the original signer and the proxy group from the warrant m_ω and identify the actual proxy signers from ASID. The verifier computes $Y_D = \prod_{i=1}^t y_i$ and validates the proxy signature by checking if

$$g^S = R^R (KY_D (y_o Y A_o)^{H(m_\omega \| K)})^{H(A_o \| R \| ASID \| m)} \pmod p. \tag{12}$$

3. CRYPTANALYSIS OF HW SCHEME

The HW scheme is more secure than Hwang *et al.*'s nonrepudiable threshold proxy signature scheme with known signers [12]. Hsu *et al.* [13] claimed that their scheme is secure against some potential compromising, forgery, and collusion attacks based on the well known one-way hash function [15] and the discrete logarithm problem cryptographic assumptions [16]. In this section, we would like to show that the HW scheme cannot resist forgery attack which demonstrates that the HW scheme cannot satisfy the security requirements of unforgeability.

First, we review a detailed analysis of the scheme's security against the forgery attack in [13]. Hsu *et al.* analyze the security of the threshold proxy signature through the proxy signature verification Eq. (12). Let $V = K(y_o Y A_o)^{H(m_\omega \| K)}$. Under the *DL* and *OWHF* assumptions, although $(m', ASID', A_o', V')$ is given, the adversary cannot determine (R', S') which satisfy the proxy signature verification equation. In the same way, when $(m', ASID', A_o', V')$ is given, the adversary cannot determine (V', m_ω', K') which passes the signature verification equation. Nevertheless, we will show that the HW scheme can not resist universal forgery attack.

In the following, we show a kind of attack against the scheme. The attacker can forge a valid proxy signature of any message m' by performing the steps. The adversary randomly chooses a warrant m_ω' at will. The adversary can determine the content of the warrant, such as a new threshold value t' and the time constraint, etc. The adversary could frame any proxy subgroup with t' or more members as the actual proxy signature group. Without loss of generality, assume that the adversary wants to frame $D' = \{D_1, D_2, \dots, D_{t'}\}$ in the proxy group G . Let $ASID'$ be the collection of identities of all the actual proxy group D' .

In order to generate a (t', n) threshold proxy signature on message m' , the adversary computes $A_o' = (Y_{y_o})^{-1} \pmod p$, chooses two random integers $a, b \in Z_q$ and computes

$$R' = g^a \bmod p, \quad K' = g^b \left(\prod_{i=1}^{t'} y_i \right)^{-1} \bmod p, \quad (13)$$

$$S' = R' + (bH(m'_\omega \| K') + a)H(A'_0 \| R' \| ASID' \| m') \bmod q. \quad (14)$$

Therefore, $(R', S', K', A'_0, m'_\omega, ASID')$ is a valid threshold proxy signature of any message m' . This is because it can pass the signature verification equation.

$$\begin{aligned} g^{S'} &= g^{(R' + (bH(m'_\omega \| K') + a)H(A'_0 \| R' \| ASID' \| m'))} \\ &= R'^{R'} \left(g^{bH(m'_\omega \| K')} \cdot g^a \right)^{H(A'_0 \| R' \| ASID' \| m')} \\ &= R'^{R'} (K' Y_{D'} (y_o Y_{A'_0})^{H(m'_\omega \| K')})^{H(A'_0 \| R' \| ASID' \| m')} \bmod p \end{aligned} \quad (15)$$

Thus, any verifier will identify O as the original signer, $ASID'$ as the identity information of the actual proxy signers from the signature. In essence, the original signer O and the actual proxy group $ASID'$ have never participated in any proxy signature stage on message m' . In other words, any attacker can successfully forge a valid proxy signature with known signers on any message, any warrant and any actual proxy signer group. Therefore, the HW scheme does not hold the security properties of unforgeability.

4. NEW IMPROVEMENT

Based on the HW scheme, we propose our improved scheme. As we have discussed above, the reason our attack succeeds is the fact that the malicious attacker can choose two random shared integers R' and K' by avoiding the DL and $OWHF$ assumptions, then further determine the partial signature value S . In our proposed scheme, we will take some countermeasures so that the adversary has to be faced with the DL and $OWHF$ assumptions when the malicious attacker can choose the shared integers R' or K' . The new improvement can also be divided into four phases: the initialization phase, the proxy share phase, the proxy signature generation phase and the proxy signature verification phase.

4.1 Initialization Phase

The system parameters are almost the same as those in the HW scheme. The only difference is that in our improvement we use two hash functions $H_1(\cdot)$, $H_2(\cdot)$.

4.2 Proxy Share Phase

The proxy share phase is also subdivided into three stages.

- Proxy Sharing:

The proxy signer group G perform a (t, n) -VSS scheme to generate their secret shares. First, the proxy signer P_i randomly chooses a $(t-1)$ -degree polynomial over Z_q

$$f_i(x) = \sum_{l=1}^{t-1} a_{il}v^l + (ID_i + a_{i,0}A_o) \pmod q, \tag{16}$$

where $a_{i,l} \in Z_q$ and $l = 0, 1, 2, \dots, t - 1$. Then, each proxy signer P_i in the proxy group G obtains the secret share γ_i as in subsection 2.2.

• Proxy Signature Key Generation:

The original signer O randomly chooses an integer $k \in Z_q$ and computes the public value $K = g^k \pmod p$ and the proxy signature key

$$\sigma = kK + x_o H_1(m_\omega \parallel K) \pmod q. \tag{17}$$

• Proxy Share Generation:

The original signer O performs a (t, n) -VSS scheme to share its proxy signature key σ in the proxy signer group G as in section 2.2.

Each P_i in the proxy group validates the proxy share values σ_i by checking whether the following equality holds:

$$g^{\sigma_i} = y_o^{H_1(m_\omega \parallel K)} K^K \prod_{j=1}^{t-1} B_j^{ID_i^j} \pmod p. \tag{18}$$

4.3 Proxy Signature Generation Phase

Without loss of generality, let $D = \{P_1, P_2, \dots, P_t\}$ is the actual proxy group who want to cooperatively sign message m .

• Proxy Sub-signature Generation

Each P_i in the group D randomly chooses an integer $k_i \in Z_q$, computes and broadcasts $r_i = g^{k_i} \pmod p$ in the proxy signer group D .

After receiving $r_i = g^{k_i}$ ($j = 1, 2, \dots, t - 1, j \neq i$), P_i computes

$$R = \prod_{j=1}^t r_j \pmod p, L_i = \prod_{j=1, j \neq i}^t (-ID_j)(ID_i - ID_j)^{-1} \pmod q, \tag{19}$$

$$s_i = k_i R + (L_i \sigma_i' + x_i H_1(m_\omega \parallel K)) H_2(A_0 \parallel K \parallel R \parallel ASID \parallel m) \pmod q. \tag{20}$$

Then P_i sends the proxy sub-signature s_i to a designated clerk.

• Proxy Signature Generation

The clerk computes

$$Y = \prod_{j=1}^n y_j \pmod p, A = \prod_{j=1}^{t-1} A_j^{ID_i^j}, B = \prod_{j=1}^{t-1} B_j^{ID_i^j}. \tag{21}$$

Next, the clerk validates the proxy sub-signature s_i by checking whether the equality holds:

$$g^{s_i} = r_i^R (y_i ((y_o Y A_o^{A_o} A)^{H_1(m_o \| K)} K^K B)^{L_i})^{H_2(A_o \| K \| R \| ASID \| m)} \pmod{p}. \quad (22)$$

If all the proxy sub-signatures s_i 's are valid, the clerk calculates $S = \sum_{j=1}^t s_j \pmod{q}$. Therefore, $(R, S, K, A_o, m_o, ASID)$ is the threshold proxy signature of message m .

4.4 Proxy Signature Verification Phase

On receiving the proxy signature $(R, S, K, A_o, m_o, ASID)$, the verifier computes $Y_D = \prod_{i=1}^t y_i$. The verifier validates the proxy signature $(R, S, K, A_o, m_o, ASID)$ by checking

$$g^S = R^R (K^K Y_D (y_o Y A_o^{A_o})^{H_1(m_o \| K)})^{H_2(A_o \| K \| R \| ASID \| m)} \pmod{p}. \quad (23)$$

5. CRYPTANALYSIS OF OUR SCHEME

Now, we first explain that the improved proxy signature scheme removes the weakness of the HW scheme and is secure against the forgery attack mentioned in section 3. If an adversary makes use of the attack technique in section 3, it has to forge a new proxy signature $(R, S, K, A_o, m_o, ASID)$ to pass the signature verification Eq. (23). If the adversary first fixes (R, K, A_o) , then it is impossible to obtain S from Eq. (23) since the problem is equivalent to solving the discrete logarithm problem. If the adversary first fixes S and two of the three integers (R, K, A_o) , then it is impossible to obtain the other of the three integers (R, K, A_o) from Eq. (23). This is because the adversary will be faced with a problem to which no feasible solution is known [17].

In the following, we will first review the security model of signature schemes. Then we show that our improved scheme is secure against existential forgery under chosen message attacks (CMA) and chosen warrant attacks (CWA) in the random oracle model. The security of our scheme is based on the discrete logarithm assumption. For the security of the proxy signature, it is not enough to only consider the chosen message attack. We must still consider the security of the proxy signature under chosen warrant attacks [18].

Definition 1 *Existential CMA and CWA security of a threshold proxy signature:* A probabilistic algorithm $A(t, q_H, q_{sig}, \varepsilon)$ -breaks a (t, n) threshold proxy signature with non-negligible probability if A can fulfill one of the two targets: (1) After A corrupts the original signer and at most $t - 1$ proxy signers, makes queries to the hash function oracles and requests threshold proxy signatures on adaptively chosen messages, A outputs a new forged signature (m, σ) on some message m with non-negligible probability, where the probability is taken over the coins of A and the proxy signature algorithm and the hash function oracles. (2) After A corrupts proxy signers, makes queries to the hash function oracles requests delegations from the original signer on adaptively chosen warrants and proxy signatures on adaptively chosen messages, A outputs a new forged warrant with non-negligible probability, where the probability is taken over the coins of A and the proxy signature algorithm and the hash function oracles. A (t, n) threshold proxy signature is said to be $(t, q_H, q_{sig}, \varepsilon)$ -secure if no forger can $(t, q_H, q_{sig}, \varepsilon)$ -breaks it.

Definition 2 *DL assumption:* A probabilistic algorithm D is said to (t, ε) -break DL problems in a group $GF_{g,p}$, if D runs in at most t steps and computes the discrete logarithm $DL(g^a) = a$ a given input (g, p, g) and g^a with probability at least ε , where a is taken over the coins of D and a chosen uniformly from Z_q . The group $GF_{g,p}$ is said to be a (t, ε) - DL group, if no algorithm can (t, ε) -break DL problems in the group $GF_{g,p}$.

In fact, in the proposed scheme, both the proxy signature key and the proxy signature are produced by a variant of the Schnorr signature scheme [19]. Now, we use the technique of Pointcheval and Stern [20] to discuss the security of our scheme. First, let us review the forking lemma.

Lemma 1 (The Forking lemma) Let A be a probabilistic polynomial time Turing machine whose input only consists of public data. Assume that, within a time bound T , A produces a valid signature $(m, \sigma_1, h, \sigma_2)$ with non-negligible probability. If the triple (m, σ_1, h) can be simulated without knowing the secret key, without an indistinguishable distribution probability, then there is another machine which has control over the machine that can be obtained from A by replacing the interaction with the signer by a simulation and which produces two valid signature $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2)$ such that $h \neq h'$ (where $h = H(m \parallel \sigma_1)$, $h' = H(m \parallel \sigma_1')$).

In the threshold situation, assume that the adversary can corrupt at most $t - 1$ proxy signers. Now, we demonstrate the relation between the security of threshold proxy partial signature and the security of proxy sub-signature.

Lemma 2 In our threshold proxy signature scheme, the security of threshold proxy partial signature S is equivalent to the security of proxy sub-signature s_i .

Proof: Without loss of generality, assume that the actual proxy signers are $\{P_1, P_2, \dots, P_t\}$. The adversary A chooses P_i the target proxy signer of attack. A can corrupt the proxy signers $\{P_1, P_2, \dots, P_t\}$. A can obtain their proxy sub-signatures on message m straightly from $\{P_1, P_2, \dots, P_t\}$ or can generate those proxy sub-signatures s_j ($j = 1, 2, \dots, t$) since A even can have their private keys and their proxy signature keys. On one hand, if the proxy sub-signatures s_j can be forged, then the signature S can be forged. It is obviously right since $S = \sum_{j=1}^t s_j \pmod q$. On the other hand, given $t - 1$ proxy signatures $(r_j, s_j, K, A_o, m_o, ASID)$ ($j = 1, 2, \dots, t$), A can generate a valid threshold proxy signature $(R, S, K, A_o, m_o, ASID)$ on message m . Then A can compute a valid proxy sub-signatures s_j 's of the proxy P_j . A computes

$$r_i = \frac{R}{\prod_{j=1, j \neq i}^t r_j} \pmod p, \quad s_i = S - \sum_{j=1}^t s_j \pmod q.$$

Then $(r_i, s_i, K, A_o, m_o, ASID)$ is a valid sub-signature of proxy signer P_i . This is because

$$\begin{aligned}
g^{s_i} &= g^{\sum_{j=1, j \neq i}^{t-1} s_j} \pmod p \\
&= \frac{R^R (K^K \prod_{i=1}^t y_i (y_o Y A_o^{A_o})^{H_1(m_\omega \| K)})^{H_2(A_o \| K \| R \| ASID \| m)}}{\prod_{j=1, j \neq i}^t r_j^R (K^K y_i (y_o Y A_o^{A_o})^{H_1(m_\omega \| K)})^{H_2(A_o \| K \| R \| ASID \| m)}} \pmod p \\
&= r_i^R (y_i ((y_o Y A_o^{A_o})^{H_1(m_\omega \| K)} K^K B)^{L_i})^{H_2(A_o \| K \| R \| ASID \| m)} \pmod p. \tag{24}
\end{aligned}$$

Thus, we have proved the Lemma 2.

Theorem 1 Let GF_p be a (t', ε') -DL group. The proposed threshold proxy signature is $(t, q_H, q_{sig}, \varepsilon)$ -secure.

Proof: First, assume that an adversary A can break our proposed threshold proxy signature scheme by chosen message attack, then through the forking lemma, it can obtain valid proxy signatures $(R, S_1, K, A_o, m_\omega, ASID)$ and $(R, S_2, K, A_o, m_\omega, ASID)$.

$$\begin{aligned}
S_1 &= \sum_{i \in ASID} (k_i R + (L_i \sigma_i' + x_i H_1(m_\omega \| K))) H_2(A_o \| K \| R \| ASID \| m) \pmod q. \\
&= \sum_{i \in ASID} k_i R + (\sigma + \sum_{i \in ASID} x_i H_1(m_\omega \| K)) h_{21} \pmod q, \tag{25}
\end{aligned}$$

$$\begin{aligned}
S_2 &= \sum_{i \in ASID} (k_i R + (L_i \sigma_i' + x_i H_1(m_\omega \| K))) H_2'(A_o \| K \| R \| ASID \| m) \pmod q. \\
&= \sum_{i \in ASID} k_i R + (\sigma + \sum_{i \in ASID} x_i H_1(m_\omega \| K)) h_{22} \pmod q. \tag{26}
\end{aligned}$$

Thus, we get

$$S_2 - S_1 = (\sigma + \sum_{i \in ASID} x_i H_1(m_\omega \| K)) (h_{22} - h_{21}) \pmod q. \tag{27}$$

Then it is easy to compute

$$\sum_{i \in ASID} x_i = \frac{S_2 - S_1 - \sigma (h_{22} - h_{21})}{H_1(m_\omega \| K) (h_{22} - h_{21})} \pmod q. \tag{28}$$

It is feasible since under the chosen message attack, the adversary A can corrupt the original signer and at most $t - 1$ proxy signers. So σ can be obtained. Let P_i be the only uncorrupted proxy signers in D . Now by the adversary A , the secret key of P_i can be computed as follows,

$$x_i = \frac{S_2 - S_1 - \sigma(h_{22} - h_{21})}{H_1(m_\omega \| K)(h_{22} - h_{21})} - \sum_{j \in ASID, j \neq i} x_j \pmod{q}. \quad (29)$$

Note that $y_i = g^{x_i} \pmod{p}$. Therefore, we can solve the *DL* problem.

Secondly, assume that an adversary A can break our proposed threshold proxy signature scheme by chosen warrant attacks, then through the forking lemma, it can get two valid proxy signatures $(R, S_1, K, A_o, m_\omega, ASID)$ and $(R, S_2, K, A_o, m_\omega, ASID)$.

$$\begin{aligned} S_1 &= \sum_{i \in ASID} (k_i R + (L_i \sigma'_i + x_i H_1(m_\omega \| K))) H_2(A_o \| K \| R \| ASID \| m) \pmod{q} \\ &= \sum_{i \in ASID} k_i R + (\sigma + \sum_{i \in ASID} x_i h_{11}) H_2(A_o \| K \| R \| ASID \| m) \pmod{q}. \\ S_2 &= \sum_{i \in ASID} (k_i R + (L_i \sigma'_i + x_i H_1'(m_\omega \| K))) H_2(A_o \| K \| R \| ASID \| m) \pmod{q} \end{aligned} \quad (30)$$

$$= \sum_{i \in ASID} k_i R + (\sigma + \sum_{i \in ASID} x_i h_{12}) H_2(A_o \| K \| R \| ASID \| m) \pmod{q}. \quad (31)$$

Then, we have

$$S_2 - S_1 = \sum_{i \in ASID} x_i H_2(A_o \| K \| R \| ASID \| m)(h_{12} - h_{11}) \pmod{q}, \quad (32)$$

$$\sum_{i \in ASID} x_{i1} = \frac{S_2 - S_1}{H_2(A_o \| K \| R \| ASID \| m)(h_{12} - h_{11})} \pmod{q}. \quad (33)$$

In a similar way, we can compute the secret key σ of the uncorrupted proxy signer P_i and solve the *DL* problem.

6. CONCLUSION

In this paper, we show that the HW scheme is not secure against the universal forgery attack. In the HW scheme, one adversary can frame the original signer and any *actual* proxy group to forge (t, n) threshold proxy signature on any message m . To thwart the attack, we propose an improvement on the HW scheme only with minimal extra computation. Our proposed threshold proxy signature is secure against chosen message attack and chosen warrant attack in the random oracle model.

REFERENCES

1. M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: Delegation of the power to sign messages," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E79-A, 1996, pp. 1338-1354.

2. T. Okamoto, M. Tada, and E. Okamoto, "Extended proxy signatures for smart cards," in *Proceedings of the 2nd International Workshop on Information Security*, LNCS 1729, 1999, pp. 247-258.
3. A. Bakker, M. Steen, and A. S. Tanenbaum, "A law-abiding peer-to-peer network for free-software distribution," in *Proceedings of the IEEE International Symposium on Network Computing and Application*, 2001, pp. 60-67.
4. I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A security architecture for computational grids," in *Proceedings of the 5th Conference on Computer and Communications Security*, 1998, pp. 83-92.
5. S. S. M. Chow, R. W. C. Liu, L. C. K. Hui, and S. M. Yiu, "Identity-based delegation network," in *Proceedings of International Conference on Cryptology in Malaysia*, LNCS 3715, 2005, pp. 99-115.
6. K. Zhang, "Threshold proxy signature schemes," in *Proceedings of the 1st International Workshop on Information Security*, 1997, pp. 282-290.
7. S. J. Kim, S. J. Park, and D. H. Won, "Proxy signatures, revisited," Springer Berlin, Heidelberg, LNCS 1334, 1997, pp. 223-232.
8. C. H. Yang, S. F. Tzeng, and M. S. Hwang, "On the efficiency of nonrepudiable nonrepudiable threshold proxy signatures with known signers," *Journal of Systems and Software*, Vol. 73, 2004, pp. 507-514.
9. H. M. Sun, "An efficient nonrepudiable threshold proxy signatures with known signers," *Computer Communications*, Vol. 22, 1999, pp. 717-722.
10. C. L. Hsu, T. S. Wu, and T. C. Wu, "Improvement of threshold proxy signature scheme," *Applied Mathematics and Computation*, Vol. 136, 2003, pp. 315-321.
11. Z. W. Tan, Z. J. Liu, and M. S. Wang, "On the security of some nonrepudiable threshold proxy signature schemes," in *Proceedings of Information Security Practice and Experience*, LNCS 3439, 2005, pp. 374-385.
12. M. S. Hwang, I. C. Lin, and E. J. L. Lu, "A secure nonrepudiable threshold proxy signature scheme with known signers," *Informatica*, Vol. 11, 2000, pp. 137-144.
13. C. L. Hsu and T. S. Wu, "Efficient nonrepudiable threshold proxy signature scheme with known signers against the collusion attack," *Applied Mathematics and Computation*, Vol. 168, 2005, pp. 305-319.
14. T. P. Pedersen, "Distributed provers with applications to undeniable signatures," in *Proceedings of Advance in Cryptology – EUROCRYPTO*, LNCS 547, 1991, pp. 221-242.
15. W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. 22, 1976, pp. 644-654.
16. T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, Vol. 31, 1985, pp. 469-472.
17. D. R. Stinson, *Cryptography: Theory and Practice*, CRC Press, New York, 2002.
18. Z. W. Tan and Z. J. Liu, "Provably secure delegation-by-certification proxy signature Schemes," in *Proceedings of ACM 3rd International Conference on Information Security*, 2004, pp. 38-43.
19. C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, Vol. 4, 1991, pp. 161-174.
20. D. Pointcheval and J. Stern, "Security of proofs for signatures," in *Proceedings of Advance in Cryptology – EUROCRYPTO*, LNCS 1070, 1996, pp. 387-398.

Zuo-Wen Tan (譚作文) received the M.S. degree in Mathematics from Xiangtan University, Hunan, in 2002 and the Ph.D. degree in Applied Mathematics from the Institute of Systems Science (ISS), Academy of Mathematics and Systems Science (AMSS), Chinese Academy of Sciences (CAS) in 2005. He has been an Assistant Processor of Computer Science Department, College of Information Management, Jiangxi University of Finance & Economics, since 2006. His research interests include information security and cryptology.

Zhuo-Jun Liu (劉卓軍) received the Ph.D. degree in Computer Science and Mathematics from the Institute of Systems Science (ISS), the Chinese Academy of Sciences (CAS) in 1988. He became a professor of Computer Science at ISS of CAS in 1995. From May of 1992 to December of 1994, he was doing Symbolic and Algebraic Computation (SAC) research at Kent State University as a visiting professor. His areas of research include symbolic computation, error-correcting codes and applications and cryptography.