Scientia Iranica D (2012) 19 (3), 674–679

Sharif University of Technology

Scientia Iranica

Transactions D: Computer Science & Engineering and Electrical Engineering

www.sciencedirect.com

Research note

Analysis of frame attack on Hsu et al.'s non-repudiable threshold multi-proxy multi-signature scheme with shared verification

Samaneh Mashhadi

Department of Mathematics, Iran University of Science & Technology, Tehran, P.O. Box 1684613114, Iran

Received 23 February 2011; revised 2 August 2011; accepted 26 September 2011

KEYWORDS

Proxy signature;
Threshold proxy signature;
Threshold multi-proxy multi-signature scheme;
Non-repudiation.

Abstract Tzeng et al. proposed a threshold multi-proxy multi-signature scheme with threshold verification. Recently, Hsu et al. pointed out that Tzeng et al.'s scheme was vulnerable to insider attacks and proposed an improvement to eliminate the pointed out security leak. We will show that Hsu et al.'s improvement cannot resist the frame attack. That is, after intercepting a valid proxy signature, an adversary can change the original signers to himself, and forge a proxy signature. To remedy this weakness, we will propose a new method.

1. Introduction

In 1996, Mambo et al. [1] first introduced the concept of a proxy signature. A proxy signature scheme allows an original signer to delegate its signing power to a designated person, called the proxy signer, who can generate the proxy signature of a message on behalf of the original signer. The verifier can verify and distinguish between the original signature and the proxy signature at the verification stage.

In a (t, n) threshold proxy signature scheme, the proxy secret key, generated by an original signer, is shared among a proxy group of n proxy signers delegated by the original signer. In a proxy group, any t or more proxy signers can cooperatively recover the proxy secret to generate a valid proxy signature, but any $t - 1$ or less proxy signer cannot [2–6]. A secure (t, n) threshold proxy signature scheme should satisfy the following security requirements: secrecy, proxy protection, unforgeability, nonrepudiation, time constraint and known signers.

In all previous schemes, a single random verifier is adopted to verify the soundness of the ultimate signature. However, as Tzeng et al. [7] indicated, in many special applications, for instance a business contract that must be constituted between two companies, only specified verifiers have access to authenticate the proxy signature. In 2004, Tzeng et al. [7] proposed a non-repudiable threshold multi-proxy multi-signature scheme with shared verification. A threshold multi-proxy multi-signature scheme with shared verification allows a subset of original signers to delegate the signing power to a designated group of proxy signers. A valid proxy signature can be generated by a subset of the proxy signers delegated by the original signer group for a designated group of verifiers. The validity of the proxy signature can be verified by a subset of the designated verifiers [7]. As Tzeng et al. [7] indicated, the concepts and models proposed by them are very useful in many situations, especially when specific verifiers are needed. That is, only some specified verifiers can authenticate the authenticity of the proxy signature. However, their scheme was insecure.

In order to create a secure non-repudiable threshold multi-proxy multi-signature scheme with shared verification, the following security requirements should be satisfied for the proxy signature:

Secrecy. The original signers' private keys are very important. They must be kept secret. If they are discovered, the security of the system is ruined. Therefore, the system must ensure that the private keys never get derived from any information, such as the sharing of the proxy signing key or the original signers' public keys. Furthermore, no proxy signers should be able to cooperatively derive the original signers' private keys.

Proxy protection. Only the delegated proxy signer can generate valid partial proxy signatures. Even the original signers cannot create partial signatures.

E-mail address: smashhadi@iust.ac.ir.

Peer review under responsibility of Sharif University of Technology.

Production and hosting by Elsevier

doi:10.1016/j.scient.2011.09.019

(t_1, n_1) *threshold delegating.* A valid proxy share can only be cooperatively generated by t_1 or more original signers. This means that valid proxy shares cannot be created by $t_1 - 1$ or less original signers, or any third parties who are not designated as original signers.

Unforgeability. A valid proxy signature can only be cooperatively generated by t_2 or more proxy signers. This means that valid proxy signatures cannot be created by $t_2 - 1$ or less proxy signers, or any third parties who are not designated as proxy signers.

(t_3, n_3) *threshold verifying.* The validity of the generated proxy signature can only be verified by any t_3 or more verifiers out of n_3 designated verifiers. Hence, $t_3 - 1$ or fewer designated verifiers cannot check the validity of the generated proxy signature.

Non-repudiation. Any valid proxy signature must be generated by t_2 or more proxy signers. Therefore, proxy signers cannot deny that they have signed the message. In addition, the original signers cannot deny having delegated the power of signing messages to the proxy signers.

Time constraint. The proxy signing keys can be used during the delegated period only. Once they expire, the proxy signatures generated by using those keys become invalid.

Known signers. From a proxy signature, the identities of the actual original signers and the identities of actual signers can be determined.

In 2005, Bao et al. [8] pointed out that Tzeng et al.'s scheme [7] was vulnerable to frame attacks. In their attack, after intercepting a valid proxy signature generated by the subset of a proxy group, an adversary could change the warrant, m_W, and forge new proxy signatures. Therefore, the properties of proxy protection and unforgeability were not fulfilled in Tzeng et al.'s scheme. To remedy this weakness, Bao et al. also gave a new improvement. Two years later, Xie et al. [9] showed that Bao et al.'s scheme cannot resist a proxy relationship inversion attack. In their attack, the proxy group, in collusion with t_3 verifiers, could forge a valid proxy signature. This forged signature seems to be generated by the original group on behalf of the proxy group. Xie et al. also proposed a new improvement. Subsequently, Hsu et al. [10] also pointed out that Tzeng et al.'s scheme [7] was vulnerable to insider attacks, that is, any verifier could check the validity of the proxy signature by himself without the aid of other verifiers. Therefore, Tzeng et al.'s scheme could not satisfy the property of (t_3, n_3) threshold verifying. Hsu et al. [10] also proposed an improvement to eliminate the pointed out security leak. However, in 2009, Kang et al. [11] pointed out that both improved schemes proposed in [8,10] were insecure. Kang et al. [11] mentioned that [8] was also vulnerable to insider attacks, and [7,8,10] could not satisfy the property of (t_1, n_1) threshold delegating. To remedy these weaknesses, they made a new improvement, which was more secure than all the previous schemes mentioned above.

In this paper, we show that Hsu et al.'s scheme [10] still has some security weakness, which cannot resist the frame attack. As a result, it cannot satisfy the properties of proxy protection and unforgeability. Besides, we propose a new secure and practical non-repudiable threshold multi-proxy multi-signature scheme with shared verification. Our improved scheme has the following advantages over the scheme presented in [10].

- In our scheme, similar to Kang et al.'s scheme, each original signer randomly generates a $t_1 - 1$ degree polynomial and all original signers collectively generate an original signer group secret key. Thus, any t_1 or more original signer can cooperate to derive the original signer group secret key and delegate the signing capability to the proxy group, but any $t_1 - 1$ or less original signer cannot delegate the signing capability. This meets the security requirement of (t_1, n_1) threshold delegating.
- In our improved scheme, similar to [8,11], the warrant, m_W, is part of the partial signature. Hence, our scheme can resist the frame attack.

The rest of this paper is organized as follows: in Section 2, we briefly review Hsu et al.'s scheme [10]. In Section 3, our cryptanalysis on Hsu et al.'s scheme is given out. We introduce the new improvement in Section 4. The security properties of the proposed scheme are discussed in Section 5. Finally, we draw our conclusions in Section 6.

2. Brief review of Hsu et al.'s scheme

This scheme consists of five phases: initialization, secret share generation, proxy share generation, proxy signature generation, and proxy signature verification. The system parameters and corresponding notations are defined as follows:

- p: a large public prime such that $p - 1$ has a large prime factor.
- q: a large public prime factor of $p - 1$.
- g: a public integer of order q in $\mathbb{Z}_p^*$.
- $h(.)$: a public one-way hash function.
- $\|$: the concatenation of strings.
- $G_O = \{O_1, O_2, \ldots, O_{n_1}\}$: the original signer group of n_1 original signers.
- $G_P = \{P_1, P_2, \ldots, P_{n_2}\}$: the proxy signer group of n_2 proxy signers.
- $G_V = \{V_1, V_2, \ldots, V_{n_3}\}$: the verifier group of n_3 designated verifiers.
- m_W: a warrant which records the identities of the group members in G_O, G_P and G_V, the parameters (t_1, n_1), (t_2, n_2), (t_3, n_3) and the valid delegation time, etc.
- *AOSID*: the identities of the actual original signers.
- *APSID*: the identities of the actual proxy signers.
- $x_{O_i} \in \mathbb{Z}_q^*$: the secret key of O_i selected by him/herself.
- $y_{O_i} = g^{x_{O_i}} \bmod p$: the certified public key of O_i.
- $x_{P_i} \in \mathbb{Z}_q^*$: the secret key of P_i selected by him/herself.
- $y_{P_i} = g^{x_{P_i}} \bmod p$: the certified public key of P_i.
- $x_{V_i} \in \mathbb{Z}_q^*$: the secret key of V_i selected by him/herself.
- $y_{V_i} = g^{x_{V_i}} \bmod p$: the certified public key of V_i.
- $X_P \in \mathbb{Z}_q^*$: the proxy group private key.
- $X_V \in \mathbb{Z}_q^*$: the verifier group private key.
- $Y_P = g^{X_P} \bmod p$: the certified public key of G_P.
- $Y_V = g^{X_V} \bmod p$: the certified public key of G_V.
- CA: a certificate authority that certified all public keys.
- SDC: a share distribution center.

SDC selects p, q, g, and $h(.)$ and determines (X_P, Y_P) and (X_V, Y_V).

2.1. Secret share generation phase

SDC randomly chooses $f_p(x)$ and $f_v(x)$ for G_P and G_V, respectively, where:

$$f_p(x) = X_P + a_{p,1}x + \cdots + a_{p,t_2-1}x^{t_2-1} \bmod q,$$
$$f_v(x) = X_V + a_{v,1}x + \cdots + a_{v,t_3-1}x^{t_3-1} \bmod q.$$

Then, SDC computes P_i's secret share as $\alpha_{P_i} = f_p(y_{P_i})$ and its corresponding public key $\beta_{P_i} = g^{\alpha_{P_i}} \bmod p$. In the same way, SDC computes the secret share and its corresponding public key for $V_j \in G_V$, as $\alpha_{V_j} = f_p(y_{V_j})$ and $\beta_{V_j} = g^{\alpha_{V_j}} \bmod p$, respectively.

2.2. Proxy share generation phase

Without loss of generality, let $D_O = \{O_1, O_2, \ldots, O_{t_1}\}$. To delegate the signing capability, D_O acts as follows:

1. Each $O_i \in D_O$ chooses a random number, $k_i \in \mathbb{Z}_q^*$, and broadcasts $K_i = g^{k_i} \bmod p$ to other signers in D_O and a designated clerk (DC).
2. After receiving $K_j (j = 1, 2, \ldots, t_1; j \neq i)$, each $O_i \in D_O$ computes $K = \prod_{i=1}^{t_1} K_i \bmod p$, and $\sigma_{O_i} = k_i K + x_{O_i} h(K \| m_W \| AOSID) \bmod q$.
3. Each $O_i \in D_O$ sends partial proxy share σ_{O_i} to DC.

After receiving all partial proxy shares, DC performs the following steps to generate a valid proxy share σ:

1. Compute $\sigma_O = \sum_{i=1}^{t_1} \sigma_{O_i} \bmod q$.
2. Check $g^{\sigma_O} = K^K \prod_{i=1}^{t_1} y_{O_i}^{h(K \| m_W \| AOSID)} \bmod p$, compute the proxy share σ as $\sigma = t_2^{-1} \sigma_O \bmod q$, and broadcast $(\sigma, m_W, K, AOSID)$ to G_P.

2.3. Proxy signature generation phase

The validity of $(\sigma, m_W, K, AOSID)$ is verified by each $P_i \in G_P$ by checking that:

$$g^{\sigma} = \left(K^K \prod_{i=1}^{t_1} y_{O_i}^{h(K \| m_W \| AOSID)} \right)^{t_2^{-1}} \bmod p.$$

Without loss of generality, let $D_P = \{P_1, P_2, \ldots, P_{t_2}\}$. In order to create a threshold proxy signature on a message, m, D_P act as follows:

1. Each $P_i \in D_P$ randomly chooses a number, $\omega_i \in \mathbb{Z}_q^*$, and broadcasts $r_{P_i} = g^{\omega_i} \bmod p$ to other proxy signers in D_P and DC.
2. Each $P_i \in D_P$ computes and broadcast $r'_{P_i} = (Y_V)^{\omega_i + \alpha_{P_i} L_{P_i}} \bmod p$, where $L_{P_i} = \prod_{j=1, j \neq i}^{t_2} (-y_{P_j})(y_{P_i} - y_{P_j})^{-1} \bmod q$.
3. Each $P_i \in G_P$ computes $R = \prod_{i=1}^{t_2} r_{P_i} \bmod p$, $R' = \prod_{i=1}^{t_2} r'_{P_i} \bmod p$, and:

   $$s_i = R' \alpha_{P_i} L_{P_i} + \omega_i R + (\sigma + x_{P_i}) h(R \| R' \| APSID \| m) \bmod q.$$

   Here, s_i is the individual signature, which is sent to DC.

After receiving all valid s_i, DC computes $S = \sum_{i=1}^{t_2} s_i \bmod q$ and checks the validity of S and s_i by the following equalities:

$$g^S = Y_P^{R'} R^R \left(K^K \prod_{i=1}^{t_1} y_{O_i}^{h(K \| m_W \| AOSID)} \prod_{j=1}^{t_2} y_{P_j} \right)^{h(R \| R' \| APSID \| m)} \bmod p, \quad \text{and}$$

$$g^{s_i} = \beta_{P_i}^{R' L_{P_i}} r_{P_i}^{R} \times \left(\left(K^K \prod_{i=1}^{t_1} y_{O_i}^{h(K \| m_W \| AOSID)} \right)^{t_2^{-1}} y_{P_i} \right)^{h(R \| R' \| APSID \| m)} \bmod p.$$

Then, the proxy signature of m is $(m_W, K, AOSID, R, S, APSID)$.

2.4. Proxy signature verification phase

Without loss of generality, let $D_V = \{V_1, V_2, \ldots, V_{t_3}\}$. They can verify the validity of the threshold proxy signature by the following performance:

1. According to m_W, $AOSID$ and $APSID$, each verifier gets the public keys of the original signers and proxy signers from the CA, knows who the actual original signers and the actual proxy signers are, and the threshold parameters (t_1, n_1), (t_2, n_2), (t_3, n_3), etc.
2. Each $V_i \in D_V$ computes and sends $r'_{V_i} = (RY_P)^{\alpha_{V_i} L_{V_i}} \bmod p$, to other verifiers in D_V where $L_{V_i} = \prod_{j=1, j \neq i}^{t_3} (-y_{V_j})(y_{V_i} - y_{V_j})^{-1} \bmod q$.
3. Each $V_i \in D_V$ computes $\prod_{i=1}^{t_3} r'_{V_i} = R'$, and verifies the validity of the proxy signature $(m_W, K, AOSID, R, S, APSID)$ for the message, m, by the following equality:

   $$g^S = Y_P^{R'} R^R (V_O V_P)^{h(R \| R' \| APSID \| m)} \bmod p,$$

   where $V_O = K^K \prod_{i=1}^{t_1} y_{O_i}^{h(K \| m_W \| AOSID)}$ and $V_P = \prod_{j=1}^{t_2} y_{P_j}$.

3. Security leak of Hsu et al.'s scheme

Recently, Bao et al. [8] presented a frame attack on [7]. Subsequently, Xie et al. [9] showed that [8] cannot resist the proxy relationship inversion attack.

Here, by a combination of the two attacks mentioned above, we will propose a new frame attack on Hsu et al.'s scheme [10]. Assume that $(m_W, K, AOSID, R, S, APSID)$ is a valid proxy signature of message m generated by D_p on behalf of D_O. Let $B = \{B_1, B_2, \ldots, B_n\}$ be an adversary (B_i can be any participant) and B' be an arbitrary subset of B of order t, ($B' \subseteq B$, $\text{card}(B') = t$, and t, n are arbitrary). Let IDB and IDB' be the identities of B and B', respectively. B', in collusion with t_3 verifiers D_V, can generate a valid proxy signature $(m'_W, K', IDB', R, S', APSID)$ for message m, such that any arbitrary t_3 or more verifiers, D'_V, verify that $(m'_W, K', IDB', R, S', APSID)$ is generated by D_P on behalf of B'. In other words, we show that B' can change D_O, t_1, and G_O to himself, t, and B, respectively, and generate a valid proxy signature for message m. The details of this attack are as follows.

Firstly, D_V computes $R' = \prod_{i=1}^{t_3} r'_{V_i} = \prod_{i=1}^{t_3} (RY_P)^{\alpha_{V_i} L_{V_i}} \bmod p$ and sends R' to B'. Then, B' generates a warrant m'_W which records the identities of the group members in B, G_P, and G_V, the parameters (t, n), (t_2, n_2), (t_3, n_3) and the valid delegation time, etc. Then, any $B_i \in B'$ executes the following steps:

1. Choose a random $a'_i \in \mathbb{Z}_q^*$ and broadcast $k'_i = g^{a'_i} \bmod p$ to other players in B';
2. After receiving k'_j $(j = 1, 2, \ldots, t; j \neq i)$, each $B_i \in B'$ computes:

   $$K' = \prod_{i=1}^{t} k'_i \bmod p,$$

   and:

   $$\sigma_{B_i} = a'_i K' + x_{B_i} h(K' \| m'_W \| IDB') \bmod q,$$

   where x_{B_i} is the secret key of B_i.
3. Compute $\sigma_{B'} = \sum_{i=1}^{t} \sigma_{B_i} \bmod q$.
4. B' computes $S' = (\sigma_{B'} - t_2 \sigma) h(R \| R' \| APSID \| m) + S$.
5. Finally, B' forges an illegal proxy signature $(m'_W, K', IDB', R, S', APSID)$, and it seems to be generated by the proxy group D_P on behalf of B'.

Let D'_V be the arbitrary set of t_3 verifiers, who want to verify the validity of the proxy signature $(m'_W, K', IDB', R, S', APSID)$. They compute $\prod_{V_i \in D'_V} r'_{V_i} = R'$, and check whether the following equation holds or not.

$$g^{S'} = {Y_P}^{R'} R^R \left({K'}^{K'} \prod_{B_i \in B'} y_{B_i}^{h(K' \| m'_W \| IDB')} \prod_{j=1}^{t_2} y_{P_j} \right)^{h(R \| R' \| APSID \| m)} \bmod p.$$

The forged proxy signature $(m'_W, K', IDA, R, S', APSID)$ can pass the verification equation because:

$$\begin{aligned} g^{S'} &= g^S g^{(\sigma_{B'} - t_2\sigma)h(R\|R'\|APSID\|m)} \\ &= g^{\sum_{i=1}^{t_2} s_i} g^{(\sigma_{B'} - t_2\sigma)h(R\|R'\|APSID\|m)} \\ &= g^{(\sum_{i=1}^{t_2} R'\alpha_{P_i} L_{P_i} + \omega_i R + (\sigma + x_{P_i})h(R\|R'\|APSID\|m)) + (\sigma_{B'} - t_2\sigma)h(R\|R'\|APSID\|m)} \\ &= g^{(\sum_{i=1}^{t_2} R'\alpha_{P_i} L_{P_i} + \omega_i R + x_{P_i} h(R\|R'\|APSID\|m)) + \sigma_{B'} h(R\|R'\|APSID\|m)} \\ &= g^{x_P R'} g^{\sum_{i=1}^{t_2} \omega_i R} \left(g^{\sum_{i=1}^{t_2} x_{P_i}} g^{\sum_{i=1}^{t} a'_i K' + x_{B_i} h(K' \| m'_W \| IDB')} \right)^{h(R\|R'\|APSID\|m)} \\ &= Y_P^{R'} R^R \left({K'}^{K'} \prod_{i=1}^{t} y_{B_i}^{h(K' \| m'_W \| IDB')} \prod_{i=1}^{t_2} y_{P_i} \right)^{h(R\|R'\|APSID\|m)} \bmod p. \end{aligned}$$

Thus, the validity of the forged proxy signature $(m'_W, K', IDB', R, S', APSID)$ can be verified by any t_3 or more verifiers. The major problem of Hsu et al.'s scheme [10] is that the proxy signature generation is independent of the proxy certificate $(m_W, AOSID)$. Therefore, the adversary can easily modify the proxy signatures by our attack. To guard against this attack, it is better to integrate proxy signatures with the proxy certificate $(m_W, AOSID)$. Therefore, in our improved scheme, similar to [8,11], the proxy certificate $(m_W, AOSID)$ is part of $h(R\|R'\|AOSID\|APSID\|m\|m_W)$ in individual signature s_i. Thus, after intercepting a valid proxy signature, it is impossible for anyone to replace $(AOSID, m_W)$ by another $(AOSID', m'_W)$, and at the same time, the following equality holds:

$$h(R\|R'\|AOSID\|APSID\|m\|m_W) = h(R\|R'\|AOSID'\|APSID\|m\|m'_W).$$

This is because $h(.)$ is a collision resistant hash function.

4. The proposed scheme

Our scheme can be divided into five phases: initialization, secret share generation, proxy share generation, proxy signature generation and proxy signature verification. Since the system parameters are the same as those in Hsu et al.'s scheme [10], we only describe the remaining phases below.

4.1. Secret share generation phase

SDC performs the same steps as those in Section 2.1. Moreover, in our scheme, all $O_i \in G_O$ collectively run the following steps to generate their secret shadows and the original signer group public key:

1. Each $O_i \in G_O$ chooses a random number, $a_i \in \mathbb{Z}_q^*$, and broadcasts g^{a_i} to other original signers in G_O.
2. After receiving g^{a_j} $(j = 1, 2, \ldots, n_1; j \neq i)$, each $O_i \in G_O$ computes $A = \prod_{i=1}^{n_1} g^{a_i} \bmod p$ and randomly generates a polynomial of degree $t_1 - 1$:

   $f_i(x) = (a_i A + x_{O_i}) + a_{i,1}x + \cdots + a_{i,t_1-1}x^{t_1-1} \bmod q.$

   O_i publishes $A_{i,l} = g^{a_{i,l}} \bmod p (l = 1, 2, \ldots, t_1 - 1)$.
3. Each $O_i \in G_O$ computes $f_i(j)$ and sends it to $O_j \in G_O$ via a secure channel for $j \neq i$.
4. After receiving $f_j(i)$ from O_j, O_i can validate it by checking:

   $$g^{f_j(i)} = g^{a_j A} y_{O_j} \prod_{l=1}^{t_1-1} (A_{j,l})^{i^l} \bmod p.$$

Let $f(x) = \sum_{i=1}^{n_1} f_i(x) \bmod p$; then, the secret shadow of O_i is $f(i)$, and the corresponding public key is determined by $g^{f(i)}$. The original signer group public key is $Y_O = g^{f(0)} = A^A \prod_{i=1}^{n_1} y_{O_i} \bmod p$. Finally, G_O publishes A as public information.

4.2. Proxy share generation phase

Here, we replace σ_{O_i} with:

$$\sigma_{O_i} = k_i K + \left(x_{O_i} + f(i) \prod_{j=1,\, j\neq i}^{t_1} \frac{-j}{i-j} \right) \times h(K \| m_W \| AOSID) \bmod q.$$

Therefore:

$$g^{\sigma_{O_i}} = K_i^K \left(y_{o_i} g^{f(i) \prod_{j=1,\, j\neq i}^{t_1} \frac{-j}{i-j}} \right)^{h(K\|m_W\|AOSID)},$$

and:

$$g^{\sigma} = K^K \left(\prod_{i=1}^{t_1} y_{O_i} A \prod_{i=1}^{n_1} y_{O_i} \right)^{h(K\|m_W\|AOSID)} \bmod p.$$

Finally, DC broadcasts $(\sigma, m_W, K, A, AOSID)$ to G_P.

4.3. Proxy signature generation phase

In our scheme, s_i is computed by:

$$s_i = R'\alpha_{P_i} L_{P_i} + \omega_i R + (\sigma + x_{P_i}) \times h(R\|R'\|AOSID\|APSID\|m\|m_W) \bmod q.$$

After receiving all the valid s_i, the DC computes $S = \sum_{i=1}^{t_2} s_i \bmod q$ and checks the validity of S and s_i by the following equalities:

$$g^S = {Y_P}^{R'} R^R \left(K^{Kt_2} \left(\prod_{i=1}^{t_1} y_{O_i} A \prod_{i=1}^{n_1} y_{O_i} \right)^{t_2 h(K\|m_W\|AOSID)} \times \prod_{j=1}^{t_2} y_{P_j} \right)^{h(R\|R'\|AOSID\|APSID\|m\|m_W)},$$

and:

$$g^{s_i} = {\beta_{P_i}}^{R' L_{P_i}} {r_{P_i}}^R \left(K^K \left(\prod_{i=1}^{t_1} y_{O_i} A \times \prod_{i=1}^{n_1} y_{O_i} \right)^{h(K\|m_W\|AOSID)} y_{P_i} \right)^{h(R\|R'\|AOSID\|APSID\|m\|m_W)}.$$

Then, the proxy signature of m is $(m_W, K, AOSID, R, S, APSID)$.

4.4. Proxy signature verification phase

Here, the validity of the proxy signature $(m_W, K, AOSID, R, S, APSID)$ is checked by the following equality:

$$g^S = Y_P{}^{R'} R^R (V_O V_P)^{h(R\|R'\|AOSID\|APSID\|m\|m_W)} \bmod p,$$

where:

$$V_O = K^{Kt_2} \left(\prod_{i=1}^{t_1} y_{O_i} A^A \prod_{i=1}^{n_1} y_{O_i} \right)^{t_2 h(K\|m_W\|AOSID)},$$

and:

$$V_P = \prod_{j=1}^{t_2} y_{P_j}.$$

5. Security analysis of our proposed scheme

In this section, we will show that our improved scheme not only keeps the merits of the previous schemes proposed in [7,8,10,11], but also overcomes their weaknesses.

(i) First of all, an attacker cannot obtain any original signers' secret keys x_{O_i} from equation $y_{O_i} = g^{x_{O_i}} \bmod p$. This is because of the difficult problem of solving the discrete logarithm. Similarly, an attacker cannot obtain any proxy signers' secret keys, x_{P_i}, or any verifiers' secret keys, x_{V_i}, from the equation $y_{P_i} = g^{x_{P_i}} \bmod p$ or $y_{V_i} = g^{x_{V_i}} \bmod p$. Therefore, the property of proxy protection is fulfilled in our scheme.

(ii) In our scheme, similar to Kang et al.'s scheme, each $O_i \in G_O$ randomly generates a $t_1 - 1$ degree polynomial $f_i(x) = (a_i A + x_{O_i}) + a_{i,1}x + \cdots + a_{i,t_1-1}x^{t_1-1} \bmod q$; and all $O_i \in G_O$ collectively generate the original signer group secret key, $X_O = f(0)$. Thus, any t_1 or more original signers can cooperate to derive X_O, and delegate the signing capability to the proxy group, but any $t_1 - 1$ or less original signers cannot delegate the signing capability. This meets the security requirement of (t_1, n_1) threshold delegating.

(iii) Consider the scenario of an insider attack [10]. Suppose that a malicious verifier, $V_i \in G_V$, who participated in validating the valid proxy signature $(m_W^1, K^1, AOSID^1, R^1, S^1, APSID^1)$ for m^1 attempts to check the validity of the subsequent proxy signature $(m_W^2, K^2, AOSID^2, R^2, S^2, APSID^2)$ for m^2 by himself, without the assistance of other verifiers in G_V. After the malicious verifier participates in verifying the validity of the proxy signature for m^1, he can derive:

$$R' = \prod_{i=1}^{t_3} r'_{V_i} = \prod_{i=1}^{t_3} (RY_P)^{\alpha_{V_i} L_{V_i}} = R^{X_V} Y_P^{X_V} = g^{X_V \sum_{i=1}^{t_2} \omega_i} Y_P^{X_V}.$$

It can be seen that R' is randomized by random integers ω_i's, chosen by P_i's

Therefore, a malicious verifier, $V_i \in G_V$, who participated in validating the proxy signature $(m_W^1, K^1, AOSID^1, R^1, S^1, APSID^1)$ for m^1 can derive $R'^1 = g^{X_V \sum_{i=1}^{t_2} \omega'_i} Y_P^{X_V}$, but he cannot derive $R'^2 = g^{X_V \sum_{j=1}^{t_2} \omega''_i} Y_P^{X_V}$ by himself without the assistance of other verifiers in G_V. Hence, similar to [10,11], our scheme can resist the insider attack.

(iv) In Section 3, we presented frame attack on Hsu et al.'s scheme. As Bao and the author have analyzed, this security leak inherent in [7,10] is caused by the fact that the individual signature, s_i, is independent of the proxy certificate $(APSID, m_W)$. Thus, the adversary can easily substitute the proxy certificate and frame the innocent proxy signers. In our improved scheme, similar to [8,11], the proxy certificate $(AOSID, m_W)$ is a part of $h(R\|R'\|AOSID\|APSID\|m\|m_W)$ in individual signature $s_i = R'\alpha_{P_i} L_{P_i} + \omega_i R + (\sigma + x_{P_i}) h(R\|R'\|AOSID\|APSID\|m\|m_W)$. Thus, after intercepting a valid proxy signature, it is impossible for anyone to replace $(AOSID, m_W)$ by another $(AOSID', m'_W)$, and at the same time, the following equality holds:

$$h(R\|R'\|AOSID\|APSID\|m\|m_W) = h(R\|R'\|AOSID'\|APSID\|m\|m'_W).$$

This is because $h(.)$ is a collision resistant hash function.

Hence, our scheme can resist the frame attack.

(v) Consider the collusion attack made by proxy signers. Assume that any $t_2 - 1$ or fewer proxy signers in G_P want to conspire to sign a message, m. However, they cannot reconstruct the polynomial function $f_p(x)$ and further obtain other proxy signers' secret shares and the proxy group secret key. Similarly, any $t_3 - 1$ verifiers or less in the verifier group cannot obtain any other's secret shadows. Thus, our improved scheme can resist the conspiracy attacks made by proxy signers or verifiers. Moreover, from that which has been analyzed in (i), (iv), and (v), our scheme satisfies the properties of proxy protection and unforgeability.

(vi) Consider the collusion attack made by verifiers. Assume that any $t_3 - 1$ or fewer proxy signers in G_V want to conspire to verify the validity of a proxy signature. However, they cannot reconstruct the polynomial function $f_v(x)$ and further obtain other verifiers' secret shares and the verifier group secret key. Therefore, our new scheme can resist the collusion attack made by verifiers.

Furthermore, as discussed in (iii) and (vi), our scheme follows the property of (t_3, n_3) threshold verifying.

(vii) Consider the collusion attack made by original signers. There are two cases:

- Firstly, assume that any t_1 or more original signers in G_O want to conspire to sign any warrant, m_W. They all reveal their proxy private key $f(i)$ and cooperatively reconstruct the secret polynomial function, $f(x) = \sum_{i=1}^{n_1} f_i(x) \bmod p$. They compute the proxy group secret key, $X_O = f(0)$. Thus, they can easily derive any other original signer O_j's secret shadow $f(j)$. However, they cannot obtain each private key, x_{O_j}, from $y_{O_j} = g^{x_{O_j}} \bmod p$ and k_i from $K_i = g^{k_i}$ because of the difficult problem of solving the discrete logarithm. Therefore, they cannot derive each O_j's partial signature:

$$\sigma_{O_i} = k_i K + \left(x_{O_i} + f(i) \prod_{j=1,\, j\neq i}^{t_1} \frac{-j}{i-j} \right) \times h(K\|m_W\|AOSID) \bmod q.$$

Similarly, they can reveal their proxy private key, $f_j(i)$, and cooperatively reconstruct the secret polynomial function $f_j(x) = (a_j A + x_{O_j}) + a_{j,1}x + \cdots + a_{j,t_1-1}x^{t_1-1} \bmod q$. They compute $a_j A + x_{O_j} = f_j(0)$, but they cannot obtain each private key, x_{O_j}, from $a_j A + x_{O_j}$ without knowing random number a_j. Therefore, they cannot derive each partial signature; $\sigma_{O_i} = k_i K + (x_{O_i} + f(i) \prod_{j=1,\, j\neq i}^{t_1} \frac{-j}{i-j}) h(K\|m_W\|AOSID) \bmod q$. In other words, they cannot achieve a collusion attack successfully.

Table 1: Security comparison of existing scheme with proposed scheme.

Security features	Tzeng et al.	Hsu et al.	Bao et al.	Kang et al.	Our scheme
Scheme can resist frame attacks.	No	No	Yes	Yes	Yes
Scheme can resist insider attacks.	No	Yes	No	Yes	Yes
Scheme can resist collusion attacks.	Yes	Yes	No	Yes	Yes
Scheme satisfies the property of proxy protection.	No	No	Yes	Yes	Yes
Scheme satisfies the property of unforgeability.	No	No	Yes	Yes	Yes
Scheme satisfies the property of non-repudiation.	No	No	Yes	Yes	Yes
At least t_1 original signers can delegate the signing capability.	No	No	No	Yes	Yes
At least t_2 proxy signers can generate valid proxy signature.	Yes	Yes	Yes	Yes	Yes
At least t_3 original signers can verify the validity of proxy signature.	No	Yes	No	Yes	Yes
Scheme requires a secure channel.	Yes	Yes	Yes	Yes	Yes
Each user determines his private and public keys.	Yes	Yes	Yes	Yes	Yes
SDC determines group private and public keys.	Yes	Yes	No	No	No
Proxy group has private and public keys.	Yes	Yes	Yes	Yes	No
Verifier group has private and public keys.	Yes	Yes	Yes	Yes	No
The original signer's private key cannot be derived from any information.	Yes	Yes	Yes	Yes	Yes
The proxy signer's private key cannot be derived from any information.	Yes	Yes	Yes	Yes	Yes
The verifiers's private key cannot be derived from any information.	Yes	Yes	No	Yes	Yes
The partial proxy signature cannot be generated by others.	Yes	Yes	Yes	Yes	Yes

- Secondly, assume that $t_1 - 1$ or fewer original signers of group G_O conspire to derive the original group key and each original signer's secret key. They have to reconstruct the polynomial function, $f(x) = \sum_{i=1}^{n_1} f_i(x) \bmod p$, and compute the original group secret key, $X_O = f(0)$ and each original signer O_j's secret shadow $f(j)$. But, the secret polynomial function, $f(x)$, can only be reconstructed by at least t_1 original signer's secret shadows $f(j)$. Therefore, our new scheme can resist the collusion attack made by any $t_1 - 1$ or less original signers.

From what has been analyzed in (i) and (vii), we are fully certain that the requirements of secrecy and (t_1, n_1) threshold delegating are fulfilled in our scheme.

(viii) At last, in our new scheme, there are the designated warrant m_W, the identities of the actual original signer's *AOSID* and the identities of the actual proxy signer's *APSID*. Furthermore, all verifiable equalities consist of m_W, *AOSID* and *APSID*. As discussed in (iv), we know that our scheme can resist a frame and adaptively chosen warrant attacks. Thus, the verifier can be convinced that warrant, m_W, is published by the original signers and that m_W records the stipulated period of this proxy, which provides the time constraint. Therefore, our new scheme satisfies non-repudiation, unforgeability, verifiability, time constraint, known signers, etc.

We have compared the security of the new proposed scheme with the previous schemes, and summarized the result in Table 1.

6. Conclusions

In this paper, we have demonstrated a security leak inherent in Hsu et al.'s scheme [10] to show that their scheme violates the claimed security requirements of proxy protection and unforgeability. Further, we have proposed a new efficient and secure non-repudiable threshold multi-proxy multi-signature scheme with shared verification.

Acknowledgment

This research was partially supported by Tafresh University, under contract no. 65401113.

References

[1] Mambo, M., et al. "Proxy signature: delegation of the power to sign messages", *IEICE Transactions on Fundamentals of Electronics*, 79-A(9), pp. 1338–1353 (1996).
[2] Hsu, C., et al. "New nonrepudiable threshold proxy scheme with known signers", *The Journal of Systems and Software*, 58, pp. 119–124 (2001).
[3] Hu, J. and Zhang, J. "Cryptanalysis and improvement of a threshold proxy signature scheme", *Computer Standards & Interfaces*, 31, pp. 169–173 (2009).
[4] Mashhadi, S. "A novel non-repudiable threshold proxy signature scheme with known signers", *International Journal of Network Security*, preprint (2011).
[5] Shao, J., et al. "Improvement of Yang et al.'s threshold proxy signature scheme", *The Journal of Systems and Software*, 80, pp. 172–177 (2007).
[6] Yang, C.Y., et al. "On the efficiency of nonrepudiable threshold proxy signature scheme with known signers", *The Journal of Systems and Software*, 73, pp. 507–514 (2004).
[7] Tzeng, S.F., et al. "A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification", *Future Generation Computer Systems*, 20(50), pp. 887–893 (2004).
[8] Bao, H.Y., et al. "Improvement on Tzeng et al.'s nonrepudiable threshold multi-proxy multi-signature scheme with shared verification", *Applied Mathematics and Computation*, 169, pp. 1419–1430 (2005).
[9] Xie, Q., et al. "Improvement of nonrepudiable threshold multi-proxy threshold multi-signature scheme with shared verification", *Journal of Electronics (China)*, 24(6), pp. 806–811 (2007).
[10] Hsu, C.L., et al. "Cryptanalysis and improvement of nonrepudiable threshold multi-proxy multi-signature scheme with shared verification", *Information Sciences*, 177, pp. 543–549 (2007).
[11] Kang, B., et al. "A novel nonrepudiable threshold multi-proxy multi-signature scheme with shared verification", *Computers and Electrical Engineering*, 35, pp. 9–17 (2009).

Samaneh Mashhadi was born in Tafresh, Iran, on March 27, 1982. She received the B.S. and M.S. Degrees in Mathematics, with honors, from Iran University of Science and Technology (IUST), and Amirkabir University of Technology (AUT) in 2003 and 2005, respectively. She received her Ph.D. Degree, with honors, in Mathematics (Cryptography), from IUST, in 2008, where she is currently Assistant Professor in the Department of Mathematics. She is a member of IMS as well. Her research interests include the analysis, design and application of digital signatures, secret sharing schemes, security protocols, etc.