# On Security of Two Nonrepudiable Threshold Multi-proxy Multi-signature Schemes with Shared Verification

Rongxing Lu, Zhenfu Cao, and Jun Shao (Corresponding author: Rongxing Lu)

Department of Computer Science and Engineering, Shanghai Jiao Tong University No. 1954, Huashan Road, Shanghai 200030, P. R. China (Email: {rxlu, cao-zf, shao-jun}@cs.sjtu.edu.cn)

(Received Oct. 25, 2005; revised and accepted Nov. 23, 2005 & Jan. 27, 2006)

## Abstract

In 2004, Tzeng, Yang and Hwang proposed a nonrepudiable threshold multi-proxy multi-signature scheme with shared verification. However, Bao, Cao and Wang pointed out that Tzeng et al.'s scheme cannot resist frame attacks and therefore presented an improved version. In this paper, we will show that there is another security flaw existing in both Tzeng et al.'s scheme and Bao et al.'s scheme.

Keywords: Cryptanalysis, shared verification, threshold multi-proxy multi-signature scheme

## 1 Introduction

In 2004, Tzeng, Yang and Hwang [3] proposed a novel variation of proxy signature [2] called threshold multiproxy multi-signature scheme with shared verification. In their scheme, there are four participants, i.e. the share distribution center (SDC), the original signer group, the proxy signer group and the designated verifier group. A subset of original signer group can authorize their signing rights to the proxy signer group. Then, a subset of proxy signer group can sign a valid proxy signature on behalf of the original signer group. Later, only a subset of designated verifier group can verify the validity of a purported proxy signature. Just as these features, Tzeng, Yang and Hwang claimed that their scheme can adapt to some practical applications. However, in 2005, Bao, Cao and Wang [1] analyzed Tzeng et al.'s scheme and showed that their scheme cannot resist the frame attacks. That is, a malicious subset of original signer group, after intercepting some accessible information, can frame some subsets of proxy group and forge a valid proxy signature. Therefore, they also presented an improved version, which can overcome the weakness above.

Although Bao et al.'s scheme [1] gets rid of the security

weakness encountered in Tzeng et al.'s scheme [3], yet it is still an insecure scheme. In this paper, we will point out there is another inherent security flaw existing in both Tzeng et al.'s scheme and Bao et al.'s scheme. That is, both these two schemes do not satisfy the requirement of threshold shared verification.

The rest of this paper is organized as follows. We will first review Tzeng et al.'s scheme and Bao et al.'s scheme in Section 2 and Section 3, respectively. Then, we point out the inherent security flaw in Section 4. Finally, we draw our conclusion in Section 5.

## 2 Review of Tzeng et al.'s Scheme

There are four phases constituting in Tzeng et al.'s scheme [3], i.e. the secret share generation phase, the proxy signature generation phase and the proxy signature verification phase. At the beginning, some parameters are selected and published as follows.

- p, q: Two large secure primes such that q|p-1;
- g: A generator in  $\mathbb{Z}_p^*$  of order q;
- $h(\cdot)$ : a one-way hash function;
- $m_w$ : A warrant which records the identities of the original signers in the original signer group, the proxy signers in the proxy signer group and the verifiers in the designated verifier group, the parameters  $(t_1, n_1)$ ,  $(t_2, n_2)$  and  $(t_3, n_3)$  and the valid delegation time, etc.;
- AOSID : The identities of the actual original signers;
- APSID : The identities of the actual proxy signers;

Each original signer  $U_{O_i}$  selects a secret key  $x_{O_i} \in \mathbb{Z}_q^*$ and obtains a public key  $y_{O_i} = g^{x_{O_i}} \mod p$  certificated by a certificate authority (CA). Similarly, each proxy signer  $U_{P_i}$  or verifier  $U_{V_i}$  also owns a secret key  $x_{P_i} \in \mathbb{Z}_q^*$  or  $x_{V_i} \in \mathbb{Z}_q^*$  and a public key  $y_{P_i} = g^{x_{P_i}} \mod p$  or  $y_{V_i} = g^{x_{V_i}} \mod p$  which is also certified by CA.

Let  $G_O = \{U_{O_1}, U_{O_2}, \dots, U_{O_{n_1}}\}, G_P = \{U_{P_1}, U_{P_2}, \dots, U_{P_{n_2}}\}$  and  $G_V = \{U_{V_1}, U_{V_2}, \dots, U_{V_{n_3}}\}$  be  $n_1$  original signers,  $n_2$  proxy signers and  $n_3$  designated verifiers, respectively. According to the definition of a threshold multi-proxy multi-signature scheme [3], any  $t_1$  out of  $n_1$  original signers  $(1 \leq t_1 \leq n_1)$  can represent the original signer group to delegate signing capability. Any  $t_2$  out of  $n_2$  proxy signers  $(1 \leq t_2 \leq n_2)$  can represent the original signer group. Similarly, any  $t_3$  out of  $n_3$  verifiers  $(1 \leq t_3 \leq n_3)$  can represent the designated verifier group to verify the proxy signature.

#### 2.1 Secret Share Generation Phase

The SDC first chooses a random number  $X_P \in Z_q^*$  as the proxy signer group's secret key and computes the proxy signer group's public key  $Y_P = g^{X_P} \mod p$ . Similarly,  $X_V \in Z_q^*$  and  $Y_V = g^{X_V} \mod p$  are the designated verifier group's secret key and public key. For the proxy signer group  $G_P$  and the designated verifier group  $G_V$ , the SDC randomly generates two secret polynomial functions:

$$f_P(x) = X_P + P_1 x + \dots + P_{t_2-1} x^{t_2-1} \mod q$$

and

$$f_V(x) = X_V + V_1 x + \dots + V_{t_3 - 1} x^{t_3 - 1} \mod q \qquad (1)$$

and the degrees are  $t_2 - 1$  and  $t_3 - 1$ , respectively.

Then, SDC generates secret shadows  $f_P(y_{P_i})$  and corresponding public keys  $y_{f_{P_i}} = g^{f_P(y_{P_i})} \mod p$ , where  $i = 1, 2, \cdots, n_2$  for each proxy signers. Similarly,  $f_V(y_{V_i})$  and  $y_{f_{V_i}} = g^{f_V(y_{V_i})} \mod p$ , for  $i = 1, 2, \cdots, n_3$ , are each verifier's secret shadows and corresponding public keys.

#### 2.2 Proxy Share Generation Phase

Without loss of generality, assume that  $t_1$  original signers to delegate the signing rights are  $D_O = \{U_{O_1}, U_{O_2}, \dots, U_{O_{t_1}}\}$ . Then,  $D_O$  as a group executes the following steps to delegate the signing rights to  $G_P$ .

• Each  $U_{O_i} \in D_O$  selects a random number  $a_i \in \mathbb{Z}_q^*$ , computes and broadcasts  $k_i$ , where

$$k_i = g^{a_i} \bmod p.$$

• After receiving  $k_j$ ,  $(j = 1, 2, \dots, t_1; j \neq i)$ , each  $U_{O_i} \in D_O$  computes K and  $\sigma_{O_i}$ , where

$$K = \prod_{i=1}^{t_1} k_i \mod p,$$
  
$$\sigma_{O_i} = a_i K + x_{O_i} h(K || m_w || \text{AOSID}) \mod q.$$

- And then, each  $U_{O_i} \in D_O$  sends  $\sigma_{O_i}$  to the designated clerk via a public channel.
- After receiving  $\sigma_{O_i}$ , the designated clerk first computes:

$$\sigma_O = \sum_{i=1}^{t_1} \sigma_{O_i} \mod q.$$

Then, he checks whether the following equation holds:

$$g^{\sigma_O} = K^K \prod_{i=1}^{t_1} y_{O_i}^{h(K \parallel m_w \parallel \text{AOSID})} \mod p.$$

If it does hold, he computes

$$\sigma = t_2^{-1} \cdot \sigma_O \mod q.$$

Otherwise, he checks whether the following equation holds:

$$g^{\sigma_{O_i}} = k_i^K \cdot y_{O_i}^{h(K \parallel m_w \parallel \text{AOSID})} \mod q.$$

In such a way, the designated clerk can detect an incorrect signature and then request the actual original signer to deliver a valid one.

• The designated clerk broadcasts  $(\sigma, m_w, K, AOSID)$  to  $G_P$ .

After receiving  $(\sigma, m_w, K, \text{AOSID})$ , each  $P_i \in G_P$  checks whether or not the following equation holds:

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

If it does,  $P_i$  uses  $\sigma$  as his proxy share.

#### 2.3 Proxy Signature Generation Phase

Without loss of generality, let  $D_P = \{U_{P_1}, U_{P_2}, \cdots, U_{P_{t_2}}\}$  be the actual proxy signers. Then  $D_P$  as a group executes the following steps to generate the proxy signature.

• Each  $U_{P_i} \in D_P$  selects a random number  $b_i \in \mathbb{Z}_q^*$ , computes and broadcasts  $r_{P_i}$ , where

$$r_{P_i} = g^{b_i} \bmod p.$$

• Then, each  $U_{P_i} \in D_P$  uses a secret shadow  $f_P(y_{P_i})$ and a group public key  $Y_V$  of the verifiers to compute and broadcast  $r'_{P_i}$ , where

$$r'_{P_i} = (Y_V)^{f_P(y_{P_i}) \prod_{j=1, j \neq i}^{t_2} \frac{0 - y_{P_j}}{y_{P_i} - y_{P_j}}} \mod p.$$

• After receiving all  $r_{P_j}$  and  $r'_{P_j}$ ,  $(j = 1, 2, \dots, t_2; j \neq i)$ , each  $U_{P_i} \in D_P$  computes R, R' and  $s_i$ , where

$$\begin{split} R &= \prod_{i=1}^{t_2} r_{P_i} \bmod p, \\ R' &= \prod_{i=1}^{t_2} r'_{P_i} \bmod p, \\ s_i &= R' f_P(y_{P_i}) \prod_{j=1, j \neq i}^{t_2} \frac{0 - y_{P_j}}{y_{P_i} - y_{P_j}} + b_i R + \\ & (\sigma + x_{P_i}) h(R \|\text{APSID}\|m) \bmod q, \end{split}$$

here,  $s_i$  is the individual proxy signature which is sent to the designated clerk.

• When the designated clerk receives  $s_i$ , he first computes

$$S = \sum_{i=1}^{t_2} s_i \bmod q.$$

Then, he checks whether or not the following equation holds:

$$g^{S} = Y_{P}^{R'} R^{R} \times \left( K^{K} \prod_{i=1}^{t_{1}} y_{O_{i}}^{h(K \parallel m_{w} \parallel \text{AOSID})} \prod_{j=1}^{t_{2}} y_{P_{j}} \right)^{h(R \parallel \text{APSID} \parallel m)} \mod p$$

If it does not hold, he checks whether the following equation holds,

$$\begin{array}{lcl} g^{s_{i}} & = & y_{f_{P_{i}}}^{R'\prod_{j=1, j\neq i}^{t_{2}} \frac{0-y_{P_{j}}}{y_{P_{i}}-y_{P_{j}}}} \times \\ & & \left( \left( K^{K}\prod_{i=1}^{t_{1}} y_{O_{i}}^{h(K\parallel m_{w}\parallel \text{AOSID})} \right)^{t_{2}^{-1}} y_{P_{i}} \right) \bmod p. \end{array}$$

Then, he can find a wrong signature and ask the actual proxy signer to transfer a valid one. In the end, the proxy signature of message m is  $(m_w, K, \text{AOSID}, R, S, \text{APSID}).$ 

#### 2.4 Proxy Signature Verification Phase

Without loss of generality, any  $t_3$  out of  $n_3$  verifiers in the group  $G_V$  can cooperate to verify the validity of the proxy signature. Let  $D_V = \{U_{V_1}, U_{V_2}, \dots, U_{V_{t_3}}\}$  be the actual verifiers. They can execute the following steps to verify the signature.

• According to  $m_w$ , AOSID and APSID, each verifier gets the public keys of the original signers and proxy signers from the CA and knows who the actual original signers and the actual proxy signers are.

• Each  $U_{V_i} \in D_V$  uses his secret shadow  $f_V(y_{V_i})$  and the group public key  $Y_P$  of the proxy signers to compute and broadcast  $r'_{V_i}$ ,

$$r'_{V_i} = (Y_P)^{f_V(y_{V_i}) \prod_{j=1, j \neq i}^{t_3} \frac{0 - y_{V_j}}{y_{V_i} - y_{V_j}}} \mod p.$$
(2)

• After receiving  $r'_{V_j}$ ,  $(j = 1, 2, \cdots, t_3; j \neq i)$ , each  $U_{V_i} \in D_V$  computes

$$R' = \prod_{i=1}^{t_3} r'_{V_i} \mod p.$$
(3)

Then, each verifier  $U_{V_i} \in D_V$  can check the validity of the proxy signature by the following equation,

$$g^{S} = Y_{P}^{R'} R^{R} \times \begin{pmatrix} K^{K} \prod_{i=1}^{t_{1}} y_{O_{i}}^{h(K \parallel m_{w} \parallel \text{AOSID})} \end{pmatrix}^{h(R \parallel \text{APSID} \parallel m)} \mod p.$$
(4)

If it does hold,  $(m_w, K, \text{AOSID}, R, S, \text{APSID})$  is a valid proxy signature of message m.

## 3 Review of Bao et al.'s Scheme

Bao et al.'s scheme [1] is an improved version of Tzeng et al.'s scheme [3], which not only resists the frame attacks but also eliminates the share distribution center (SDC). Thus, the secret share generation phase is not required. The systems parameters in Bao et al.'s scheme are the same as those in Tzeng et al.'s scheme [3]. To generate the verifier group public key and their secret shadows, all  $U_{V_i} \in G_V$  collectively run the following steps:

• Each  $U_{V_i} \in G_V$  randomly selects a  $t_3 - 1$  degree polynomial:

$$f'_i(x) = x_{V_i} + b_{i,1}x + \cdots + b_{i,t_3-1}x^{t_3-1} \mod q,$$

where  $x_{V_i}$  is  $U_{V_i}$ 's private key.  $U_{V_i}$  publishes  $d_{i,l} = g^{b_{i,j}} \mod p \ (l = 1, 2, \cdots, t_3 - 1).$ 

- Each  $U_{V_i} \in G_V$  computes and sends  $f'_i(j)$  to  $U_{V_j} \in G_V$  via a secure channel for  $j \neq i$ .
- After receiving  $f'_j(i)$  from  $U_{V_j}$ ,  $U_{V_i}$  can validate it by checking the following equation,

$$g^{f'_j(i)} = y_{V_j} \prod_{i=1}^{t_3-1} (d_{j,l})^{i^l} \mod p.$$

Let  $f'(x) = \sum_{i=1}^{n_3} f'_i(x) \mod q$ , then  $U_{V_i}$ 's secret shadow is f'(i), and the verifier group public key is  $Y_V = \prod_{i=1}^{n_3} g^{x_{V_i}} = \prod_{i=1}^{n_3} y_{V_i} \mod p$ .

### 3.1 Proxy Share Generation Phase

Assume  $D_O = \{U_{O_1}, U_{O_1}, \cdots, U_{O_{t_1}}\}$  are the  $t_1$  original signers to delegate the signing capability. Similar to Tzeng-Yang-Hwang scheme,  $D_O$  preforms the same steps as those in Section 2.2. Finally,  $(\sigma, m_w, K, \text{AOSID})$  are broadcasted to  $G_P$  and each  $U_{P_i}$  obtains  $\sigma$  as his proxy share. Here, a slight difference is that  $\sigma = n_2^{-1} \sigma_O \mod q$ .

### 3.2 Proxy Signature Generation Phase

First, all  $U_{P_i} \in G_P$  collectively run the following steps to generate the proxy group public key and their secret shadows:

• Each  $U_{P_i} \in G_P$  randomly generates a secret  $t_2 - 1$  degree polynomial:

$$f_i(x) = a_{i,0} + a_{i,1}x + \dots + a_{i,t_2-1}x^{t_2-1}$$
  
=  $(\sigma + x_{P_i} + k_i) + a_{i,1}x$   
 $+ \dots + a_{i,t_2-1}x^{t_2-1} \mod q,$ 

where  $k_i$  is a random number chosen by  $U_{P_i}$ .  $U_{P_i}$  then broadcasts  $g^{k_i}$  and publishes  $c_{i,l} = g^{a_{i,l}} \mod p$ ,  $(l = 1, 2, \dots, t_2 - 1)$ . Here  $c_{i,0} = g^{a_{i,0}} \mod p$  doesn't need to be published, since

$$c_{i,0} = g^{a_{i,0}} = g^{\sigma + x_{P_i} + k_i}$$
  
=  $\left( K^K \prod_{i=1}^{t_1} y_{O_i}^{h(K \parallel m_w \parallel \text{AOSID})} \right)^{n_2^{-1}} y_{P_i} g^{k_i}.$ 

- Each  $U_{P_i} \in G_P$  computes and sends  $f_i(j)$  to  $U_{P_j} \in G_P$  via a secure channel for  $j \neq i$ .
- After receiving  $f_j(i)$  from  $U_{P_j}$ ,  $U_{P_i}$  can validate it by checking

$$g^{f_j(i)} = \prod_{l=0}^{t_2-1} (c_{j,l})^{i^l} \mod p.$$

Finally,  $G_P$  publishes  $K'' = \prod_{i=1}^{n_2} g^{k_i} \mod p$  as public information. Let  $f(x) = \prod_{i=1}^{n_2} f_i(x) \mod p$ , then  $U_{P_i}$ 's secret shadow is f(i), and the proxy group public key is

$$Y_{P} = g^{f(0)}$$
  
=  $\left(K^{K} \prod_{i=1}^{t_{1}} y_{O_{i}}^{h(K \parallel m_{w} \parallel \text{AOSID})}\right) \prod_{i=1}^{n_{2}} y_{P_{i}} K'' \mod p.$ 

Without loss of generality, assume  $D_P = \{U_{P_1}, U_{P_2}, \cdots, U_{P_{t_2}}\}$  are  $t_2$  actual proxy signers who represent the proxy group to sign a message m. Then  $D_P$  performs the following steps:

- Each  $U_{P_i} \in D_P$  chooses a random  $b_i \in Z_q^*$ , computes  $r_{P_i} = g^{b_i} \mod p$  and broadcasts  $r_{P_i}$ .
- Each  $U_{P_i} \in D_P$  computes and broadcasts  $r'_{P_i}$ ,

$$r'_{P_i} = (Y_V)^{f(i) \prod_{j=1, j \neq i}^{\tau_2} \frac{0-j}{i-j}} \mod p.$$

• After receiving all  $r_{P_j}$  and  $r'_{P_j}$   $(j = 1, 2, \cdots, t_2; j \neq i)$ , each  $U_{P_i} \in D_P$  computes  $R = \prod_{i=1}^{t_2} r_{P_i} \mod p$ ,  $R' = \prod_{i=1}^{t_2} r'_{P_i} \mod p$ , and

$$s_i = b_i R + \left( x_{P_i} + R'f(i) \prod_{j=1, j \neq i}^{t_2} \frac{0-j}{i-j} \right) \times h(R \|\text{APSID}\|m\|m_w) \mod q.$$

Here  $s_i$  is the individual signature, which is sent to the designated clerk.

• The designated clerk checks whether or not the following equation holds:

$$g^{s_{i}} = r_{P_{i}}^{R} (y_{P_{i}} ((\prod_{i=1}^{n_{2}} y_{P_{i}} K'' \prod_{j=1}^{t_{2}-1} \prod_{k=1}^{n_{2}} c_{k,j}^{ij}) \times (K^{K} \prod_{i=1}^{t_{1}} y_{O_{i}}^{h(K \parallel m_{w} \parallel \text{AOSID})}))^{R' \prod_{j=1, j \neq i}^{t_{2}} \frac{0-j}{i-j}})^{h(r \parallel \text{APSID} \parallel m \parallel m_{w})} \mod n.$$

If it does hold,  $s_i$  will be accepted; otherwise, rejected.

• After receiving all valid  $s_i$ , the designated clerk computes  $S = \sum_{i=1}^{t_2} s_i \mod q$  satisfying

$$g^{S} = R^{R} (\prod_{i=1}^{t_{2}} y_{P_{i}}((\prod_{i=1}^{n_{2}} y_{P_{i}}K'') \times (K^{K} \prod_{i=1}^{t_{1}} y_{O_{i}}^{h(K \parallel m_{w} \parallel \text{AOSID})}))^{R'})^{h(r \parallel \text{APSID} \parallel m \parallel m_{w})} \mod p.$$

In this way, the proxy signature of m is  $(m_w, K, K'', \text{AOSID}, R, S, \text{APSID}).$ 

#### 3.3 Proxy Signature Verification Phase

Without loss of generality, assume  $D_V = \{U_{V_1}, U_{V_2}, \cdots, U_{V_{t_3}}\}$  are  $t_3$  actual verifiers. Then  $D_V$  as a group performs the following steps:

- According to  $m_w$ , AOSID and APSID, each verifier gets the public keys of the original signers and proxy signers from the CA and knows who the actual original signers and the actual proxy signers are.
- According to the actual proxy signers' identities,  $D_V$  computes

$$Y_{P} = \left( K^{K} \prod_{i=1}^{t_{1}} y_{O_{i}}^{h(K \parallel m_{w} \parallel \text{AOSID})} \right) \times \prod_{i=1}^{n_{2}} y_{P_{i}} K'' = g^{f(0)} \mod p. \quad (5)$$

• Each  $U_{V_i} \in D_V$  uses his secret shadow f'(i) and the group public key  $Y_P$  of the proxy signers to compute and broadcast  $r'_{V_i}$ ,

$$r'_{V_i} = (Y_P)^{f'(i) \prod_{j=1, j \neq i}^{t_3} \frac{0-j}{i-j}} \mod p.$$
(6)

• After receiving  $r'_{V_j}$ ,  $(j = 1, 2, \cdots, t_3; j \neq i)$ , each  $U_{V_i} \in D_V$  computes

$$R' = \prod_{i=1}^{t_3} r'_{V_i} \mod p.$$
(7)

Then, the validity of the signature can be checked by the following equation,

$$g^{S} = R^{R} \times (\prod_{i=1}^{t_{2}} y_{P_{i}}((\prod_{i=1}^{n_{2}} y_{P_{i}}K'') \times (K^{K} \prod_{i=1}^{t_{1}} y_{O_{i}}^{h(K \parallel m_{w} \parallel \text{AOSID})}))^{R'})^{h(R \parallel \text{APSID} \parallel m \parallel m_{w})} \mod p.$$
(8)

# 4 Security on Tzeng et al.'s and Bao et al.'s Schemes

In this section, we shall show that there is an inherent security flaw existing in Tzeng et al.'s scheme [3], as well as in Bao et al.'s scheme [1]. According to the definition of threshold multi-proxy multi-signature scheme with shared verification [3], only  $t_3$  out of  $n_3$  designated verifiers can cooperatively verify the validity of a proxy signature. However, by a close look at the proxy signature verification phase, we will find that the above feature doesn't follow.

In Tzeng et al.'s scheme [3], after each  $U_{V_i} \in D_V$  computes and broadcasts  $r'_{V_i}$  in the proxy signature verification phase, *anyone* can obtain these values  $r'_{V_1}, r'_{V_2}, \cdots, r'_{V_{t_3}}$ . Then, one can, with these values, compute R' as in Equation (3). That is,

$$R' = \prod_{i=1}^{t_3} r'_{V_i} \bmod p.$$

Then, from the Equations (1), (2), and Lagrange interpolation formula, we know

$$R' = \prod_{i=1}^{t_3} r'_{V_i}$$
  
= 
$$\prod_{i=1}^{t_3} (Y_P)^{f_V(y_{V_i}) \prod_{j=1, j \neq i}^{t_3} \frac{0 - y_{V_j}}{y_{V_i} - y_{V_j}}}$$
  
= 
$$Y_P^{X_V}$$
  
= 
$$q^{X_P X_V} \mod p.$$

Finally, when one holds  $R' = g^{X_P X_V} \mod p$ , he can verify the validity of any proxy signature  $(m_w, K, \text{AOSID}, R, S, \text{APSID})$  of message m by Equation (4),

$$\boldsymbol{g}^{S} = \boldsymbol{Y}_{\boldsymbol{P}}^{\boldsymbol{R}'} \boldsymbol{R}^{\boldsymbol{R}} \left( \boldsymbol{K}^{\boldsymbol{K}} \prod_{i=1}^{t_{1}} \boldsymbol{y}_{O_{i}}^{h(\boldsymbol{K} \parallel \boldsymbol{m}_{w} \parallel \textbf{AOSID})} \right)^{h(\boldsymbol{R} \parallel \textbf{APSID} \parallel \boldsymbol{m})} \text{ mod } \boldsymbol{p}.$$

Therefore, it is obviously that *anyone* can verify a proxy signature, and the property of shared verification doesn't follow in Tzeng et al.'s scheme [3].

Similarly, in Bao et al.'s scheme [1], the value R' is also always the same, no matter what the signature on message m is.

From the Equations (5), (6), (7), and Lagrange interpolation formula, we know

$$\begin{aligned} R' &= \prod_{i=1}^{t_3} r'_{V_i} \\ &= \prod_{i=1}^{t_3} (Y_P)^{f'(i) \prod_{j=1, j \neq i}^{t_3} \frac{0-j}{i-j}} \\ &= Y_P^{f'(0)} = g^{f(0)f'(0)} \bmod p \end{aligned}$$

Thus, when one holds  $R' = g^{f(0)f'(0)} \mod p$ , he can always verify the validity of any proxy signature  $(m_w, K, K'', \text{AOSID}, R, S, \text{APSID})$  of message m by Equation (8).

$$\begin{split} g^{S} &= R^{R} (\prod_{i=1}^{t_{2}} y_{P_{i}} ((\prod_{i=1}^{n_{2}} y_{P_{i}} K'') \times \\ (K^{K} \prod_{i=1}^{t_{1}} y_{O_{i}}^{h(K \parallel m_{w} \parallel \text{AOSID})}))^{R'})^{h(R \parallel \text{APSID} \parallel m \parallel m_{w})} \text{ mod } p. \end{split}$$

## 5 Conclusion

Threshold multi-proxy multi-signature scheme with shared verification is a novel variation of proxy signature. In this paper, we have shown that there is an inherent security flaw existing in both Tzeng et al.'s scheme [3] and Bao et al.'s scheme [1]. That is, *anyone* can verify the validity of the proxy signature, and the property of shared verification doesn't follow. Therefore, how to achieve the indeed threshold shared verification is our future work.

## Acknowledgment

The authors would like to thank Haiyong Bao for his valuable comments. This work is supported in part by the National Natural Science Foundation of China for Distinguished Young Scholars under Grant No. 60225007 and 60572155, and the Science and Technology Research Project of Shanghai under Grant Nos. 04JC14055 and 04DZ07067.

## References

 H. Y. Bao, Z. F. Cao, and S. B. Wang, "Improvement on Tzeng et al.'s nonrepudiable threshold multiproxy multi-signature scheme with shared verification", *Applied Mathematics and Computation*, vol. 169, pp. 1419-1430, 2005.

- [2] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature: delegation of the power to sign message", *IEICE Transactions on Fundamentals*, vol. E79-A, pp. 1338-1353, 1996.
- [3] S. F. Tzeng, C. Y. Yang, and M. S. Hwang, "A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification", *Future Generation Computer Systems*, vol. 20, pp. 887-893, 2004.



**Zhenfu Cao** received his B.S. degree in computer science and technology from Harbin Institute of Technology, China, in 1983, and his Ph.D. degree in mathematics from the same university. Currently, he is a professor and a doctoral supervisor at the Department of Computer Science and Engineering

of Shanghai Jiao Tong University. His main research areas are number theory, modern cryptography, theory and technology of information security etc..



Jun Shao received his B.S.in computer science from Northwestern Polytechnical University in 2003. Currently, he is a PhD candidate in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests lie in threshold cryptography and computer

security.



**Rongxing Lu** received his B.S. and M.S. degrees in computer science from Tongji University in 2000 and 2003 respectively. Currently, he is a doctoral candidate at the Department of Computer Science and Engineering of Shanghai Jiao Tong University. His research interests lie in cryptography

and network security.