2017 9th IEEE International Conference on Communication Software and Networks (ICCSN 2017)

Volume 3

May 6-8, 2017
GuangZhou, China

IEEE

Proceedings of

2017 9th IEEE International Conference on Communication Software and Networks (ICCSN 2017)

May 6-8, 2017

Guangzhou, China

ISBN: 978-1-5090-3820-6

IEEE Catalog Number: CFP1769F-CDR

2017 9th IEEE International Conference on Communication Software and Networks

ICCSN 2017

Table of Contents

Volume 3

Modern Speech Processing Technology

Computer Theory and Application

A Multi-proxy Multi-signature Scheme with Different Proxy Groups

Danni Liu, Lijuan Wang
Department of Information Science
Dalian institute of science and technology
Dalian, PR China
e-mail: dnliu2010@163.com

Dongdong Wu, Honglei Wei
School of Mechanical Engineering and Automation
Dalian Polytechnic University
Dalian, PR China
e-mail: weihl2005@163.com

***Abstract*—In most of the existing multi-proxy multi-signature (MPMS) schemes, the same proxy group is delegated the proxy right to sign by all the original members. Nevertheless, in many practical applications, original signer often demands to designate the proxy group in his own organization which is different from others'. It is seldom considered in the MPMS schemes. In this paper, we propose a MPMS scheme with different proxy groups. Furthermore, in our scheme, when authenticating the final proxy signature, a group of specified verifiers have access to do it. To prove the safety of the new scheme, we improve a security model to testify that the new one is secure based on the computational Diffie - Hellman assumption. Compared with the previous scheme, the new one offers tighter safety and better computational efficiency.**

Keywords-cryptography; revocation; addition; multi-proxy multi-signature (MPMS)

I. INTRODUCTION

In 2002, multi-proxy multi-signature(MPMS) scheme was proposed by Tzeng et al. [1], allows the group of original signers to delegate the signing capacity to the designated group of proxy signers. In our real life, there exist many applications of MPMS schemes [2-7]. Many schemes suffer a common drawback that the same proxy group is delegated the proxy capacity by all the original members. Nevertheless, in many practical applications, original signer often demands to designate his own proxy group which is different from others'.

In many schemes, a single random verifier is adopted to verify the validity of the final signature. However, as Tzeng et al. [2] indicated, in many special applications, only some specified verifiers have access to authenticate the final proxy signature together [2-4]. The problem of shared verification has not been solved in majority of multi-proxy multi-signature schemes.

In our scheme, every original signer can designate his own proxy group which is different from others', and a group of verifiers are required to authenticate the final proxy signature generated by all the proxy groups together. For convenience, multi-proxy multi-signature is called MPMS for short, and multi-proxy multi-signature scheme is abbreviated to MPMS scheme in the following content.

II. THE PROPOSED MPMS SCHEME

Inspired by the work of Sun et al. [5], Liu et al. [6] and Kang et al. [7], we proposed an unforgeable MPMS scheme with the properties of shared verification. The algorithm of the scheme is described in details as follows. p is a big prime. Both G and G_T are the groups of order p. g is a generator of G where $|G|=|G_T|=p$.

Para Gen. $G_o=(O_1,O_2,....,O_l)$ is the set of original members including l individuals; $U_j=\{P_{j,1},P_{j,2},....P_{j,n_j}\}$ is the set of proxy members including n_j individuals which are delegated signing capacity by O_j ($j=1,2,....l$). $G_v=(V_1,V_2,...,V_m)$ is the verifier group of m verifiers. ID_i is a string composed of n characters, which notes the identity of every member in G_o, G_v and U_j ($j=1,2,....l$). Let $M\in\{0,1\}^n$ be the message which is a bit string of length n. Pick random parameters $u',u_1,u_2,...,u_n,v\in G$, then set $u=(u_1,....,u_n)$. Publish the parameters (G,G_T,p,g,e,u',u,v). $H:\{0,1\}^*\in G$ is a hash function.

Key Gen. Each $O_j(1\le j\le l)\in G_o$ randomly selects $ox_j,oy_j\in Z_p^*$ and sets his secret key as $Osk_j=(ox_j,oy_j)$. Then he computes his public key $Opk_j=(OX_j,OY_j)=(g^{ox_j},g^{oy_j})$. Likewise, the secret and public keys of each $P_{j,i}(1\le i\le n_j)\in U_j$ where $j=1,2,....l$ are $Ppk_{j,i}=(PX_{j,i},PY_{j,i})=(g^{px_{j,i}},g^{py_{j,i}})$ and $Psk_{j,i}=(px_{j,i},py_{j,i})$. Then, each verifier $V_z(1\le z\le m)\in G_v$ determines his secret and public keys as $Vsk_z=(vx_z,vy_z)$ and $Vpk_z=(VX_z,VY_z)=(g^{vx_z},g^{vy_z})$, respectively.

Delegation Gen. Original signers in G_o cooperate with each other to sign a message. Each signer O_j in G_o delegates the signing capacity to $U_j=\{P_{j,1},P_{j,2},....P_{j,n_j}\}$ on behalf of O_j. Every signer O_j could delegate the proxy capability to his own group, which is different from other original ones'. Each O_j randomly picks a number $r_j\in Z_p^*$ and generates his delegation $\sigma_{W_j}=(\sigma_{W_{j1}},\sigma_{W_{j2}})$ where $\sigma_{W_{j1}}=g^{ox_j r_j}(u'\prod_{i=1}^{n}u_i^{m_i})^{r_j}$, $\sigma_{W_{j2}}=g^{r_j}$. O_j sends the delegation (W,σ_{W_j},U_j) to his proxy group U_j and SEM.

Delegation Ver. Having received the delegation, $P_{j,l}$ firstly confirms the validity. If Eq.(1) holds, $P_{j,l}$ accepts it. Otherwise, $P_{j,l}$ rejects it and requests a new delegation from O_j.

$$e(\sigma_{w_{j1}}, g) = e(OX_j, OY_j)e(u' \prod_{i=1}^{n} u_i^{m_i}, \sigma_{w_{j2}}) \quad (1)$$

Similarly, SEM validates the delegation from O_j. If Eq. (1) is valid, SEM puts (W, σ_{W_j}, U_j) into the PL. When W is expired, it is removed to the RL.

PSG of Proxy Signer. Firstly, each proxy signer $P_{j,l}$ randomly picks $k_l \in Z_p^*$ and computes $r = F(M \| W, \sigma_{W_{j2}}, g^{k_l})$. Then, $P_{j,l}$ generates his proxy signature $\sigma_{j,l} = (\sigma_{j,l1}, \sigma_{j,l2}, \sigma_{j,l3})$ for O_j where $\sigma_{j,l1} = \sigma_{W_{j1}} g^{px_{j,l} py_{j,l}} (Mv)^{k_l k_r}$, $\sigma_{j,l2} = \sigma_{W_{j2}}$, $\sigma_{j,l3} = g^{k_l k_r}$.

$P_{j,l}$ sends $\sigma_{j,l}$ and (W, σ_{W_j}, U_j) to a designated receiver c_j. The duty of c_j is to gather the proxy signatures of all the members in U_j. The receiver will check (W, σ_W, U_j) in the PL and $P_{j,l}$ in U_j. When (W, σ_{W_j}, U_j) exists in the PL and $P_{j,l}$ belongs to U_j, the receiver validates if Eq. (2) holds.

$$\begin{aligned} e(\sigma_{j,l1}, g) &= e(OX_j, OY_j)e(PX_{j,l}, PY_{j,l}) \\ &\cdot e(u' \prod_{i=1}^{n} u_i^{m_i}, \sigma_{j,l2})e((Mv)^{k_l k_r}, \sigma_{j,l3}) \end{aligned} \quad (2)$$

If it does, the receiver accepts it; otherwise, the receiver requests a valid one from $P_{j,l}$.

PSG of Proxy Group. Each proxy signer $P_{j,l}$ in U_j cooperates with each other to obtain the signature of the proxy group U_j. The designated receiver c_j combines these valid proxy signatures of members in U_j to generate a proxy signature $\sigma_j = (\sigma_{j1}, \sigma_{j2}, \sigma_{j3})$ of the group U_j where

$$\sigma_{j1} = \prod_{i=1}^{n} \sigma_{j,i1}, \sigma_{j2} = \prod_{i=1}^{n} \sigma_{j,i2}, \sigma_{j3} = \prod_{i=1}^{n_j} \sigma_{j,i3}.$$

Then, $\sigma_j = (\sigma_{j1}, \sigma_{j2}, \sigma_{j3})$ is sent to the appointed collector C. The collector C will ensure the validity of σ_j and generate the MPMS

MP MS Gen. If (W, σ_{W_j}, U_j) in the PL, the collector C validates $\sigma_j = (\sigma_{j1}, \sigma_{j2}, \sigma_{j3})$ by the following equation. If it does, C accepts it.

$$\begin{aligned} e(\sigma_{j1}, g) &= \prod_{i=1}^{n_j} (e(OX_j, OY_j)e(PX_{j,l}, PY_{j,l})) \\ &\cdot e(u' \prod_{i=1}^{n} u_i^{m_i}, \sigma_{j2})e(Mv, \sigma_{j3}) \end{aligned} \quad (3)$$

If the collector C ensures the validity of σ_j where $j = 1, 2, ..., l$, C generates the MPMS $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ as follows:

$$\sigma_1 = \prod_{j=1}^{l} \sigma_{j1}, \sigma_2 = \prod_{j=1}^{l} \sigma_{j2} \cdot \prod_{z=1}^{m} VX_z, \sigma_3 = \prod_{j=1}^{l} \sigma_{j3} \cdot \prod_{z=1}^{m} VY_z.$$

Shared Ver. Given a signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$, the verifiers in G_v perform the following steps.

Each $V_z (1 \le z \le m) \in G_v$ computes his verification token $\sigma_{vz} = (\sigma_{vz1}, \sigma_{vz2})$, where $\sigma_{vz1} = (u' \prod_{j=1}^{n} u_j^{m_j})^{vx_z}$, $\sigma_{vz2} = (Mv)^{vy_z}$, ($z = 1, 2, ..., m$), and sends it to the designated collector, who is one of the honest verifiers in G_v. The collector combines these tokens to compute $\sigma_v = (\sigma_{v1}, \sigma_{v2})$, where $\sigma_{v1} = \prod_{z=1}^{m} \sigma_{vz1}$, $\sigma_{v2} = \prod_{z=1}^{m} \sigma_{vz2}$, and accepts $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ only when Eq.(4) holds:

$$\begin{aligned} e(\sigma_1 \cdot \sigma_{v1} \cdot \sigma_{v2}, g) &= \prod_{j=1}^{l} \prod_{i=1}^{n_j} (e(OX_j, OY_j)e(PX_{j,l}, PY_{j,l})) \\ &\cdot e(u' \prod_{i=1}^{n} u_i^{m_i}, \sigma_2)e(Mv, \sigma_3) \end{aligned}$$

III. EFFICIENCY

To this day almost all of the existing MPMS schemes are constructed according to discrete logarithm problem or large number factorization problem. Bilinear pairing has good property in cryptographic applications, so it is used in the proposed scheme. For more profound analysis, the efficiency is compared between this proposed scheme to [6], which is based on bilinear pairing too.

The number of modular exponentiation and modular multiplication determines the consumption of computation. We give the following definition of two symbols: T_e : time consumption of a modular exponentiation operation; T_m : time consumption of a modular multiplication operation.

Detailed data of the two schemes' comparison is listed in the Table I. The first column denotes the dimension of MPMS in G. The second column notes the count of system parameters needed in the scheme. The Proxy Signer and MPMS columns denote the computational cost of every proxy signer's signature and the MPMS, respectively. R.P. and A.P., specify whether the scheme is with revocation and addition properties. S.V shows whether the scheme is with shared verification. 2n+4 system parameters are required in [6] while only n+4 system parameters in our scheme. The new scheme is more efficient because the computational

consumption is decreased by $3T_e + 5T_m$ in PSG Proxy Signer phase and $(3l\text{-}3)T_m$ in MP MS Gen phase. Although the two schemes both have the property of revocation, our scheme has tighter security in important aspects of member addition and shared verification. In conclusion our proposed scheme is better for practical implementation.

TABLE I. COMPARISON OF COMPUTATIONAL COMPLEXITY

Schemes	Scheme of [6]	Our new scheme
Size	3G	3G
Parameter	2n+4	n+4
ProxySigner	$9T_e$+(2n+8) T_m	$3T_e$+5 T_m
MPMS	(3*l*-2) T_m	(3*l*-3) T_m
R.P.	Yes	Yes
A.P.	No	Yes
S.V.	No	Yes

IV. SECURITY ANALYSIS

We analyze that our scheme is secure under the computational Diffie-Hellman assumption.

A. Correctness

The following proving process shows that the scheme is correct in the phases of PSG of Proxy Signer and MP MS Gen.

(1)PSG of Proxy Signer. The proxy signature generated by proxy signer is correct because of the following:

$$e(\sigma_{j,1},g) = e(\sigma_{W_1} g^{PX_{j,1}PY_{j,1}}(v'\prod_{k\in\mathcal{M}} v_k)^{h_i k_i}, g)$$

$$= e(OX_j, OY_j)e(PX_{j,1}, PY_{j,1})e(u'\prod_{i=1}^{n} u_i^{m_i}, \sigma_{j,12})e(Mv, \sigma_{j,13})$$

(2) MP MS Gen. In the phase of MPMSGen, it can be proved that the signature $\sigma_j = (\sigma_{j1}, \sigma_{j2}, \sigma_{j3})$ is correct, which is accepted by the collector C according to Eq. (3). The proving process is as follows.

$$e(\sigma_{j1}, g) = \prod_{i=1}^{n} e(\sigma_{j,i1}, g) = \prod_{i=1}^{n} (e(OX_j, OY_j)e(PX_{j,i}, PY_{j,i})) \cdot e(u'\prod_{i=1}^{n} u_i^{m_i}, \sigma_{j,12})e(Mv, \sigma_{j,13})$$

(3) MultiVer. The verifier group G_v accepts the final signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ according to Eq. (4). The proving process is as follows.

$$e(\sigma_1 \cdot \sigma_{v1} \cdot \sigma_{v2}, g) = e(\sigma_1, g) \cdot e(\sigma_{v1}, g)e(\sigma_{v2}, g)$$

$$= \prod_{j=1}^{l}\prod_{i=1}^{n_j} (e(OX_j, OY_j)e(PX_{j,i}, PY_{j,i})) \cdot e(u'\prod_{i=1}^{n} u_i^{m_i}, \sigma_2)e(Mv, \sigma_3)$$

B. Game with the Adversary $\mathcal{A}_2$

Theorem 1. The adversary $\mathcal{A}_2$ is able to obtain the forgery signature with the probability ε after performing in time T. And the numbers of Delegation Gen, PSG of Proxy Signer, PSG of Proxy Group and MP MS Gen are requested at most q_{DG}, q_{PS}, q_{PG} and q_{MS}, respectively. The situation is called that $\mathcal{A}_2$ $(T, q_{DG}, q_{PS}, q_{PG}, q_{MS}, \varepsilon)$ - breaks the MPMS scheme. If the situation happens, we can format a new algorithm $\mathcal{B}$ which can (T', ε') -break the CDH problem where $\varepsilon' \geq \frac{1}{4(n+1)(q_{DG}+q_{PS}+q_{PG}+q_{MS})}\varepsilon$ and

$$T' = T + (2n+6+3l\cdot q_{DG} + 3l\cdot q_{PS})T_e + (n+3+2l\cdot q_{DG}$$

$$+(5l-1)\cdot q_{PS} + 3\prod_{j=1}^{l}(n_j-1)\cdot q_{PG} + 3(l-1)\cdot q_{MS})T_m .$$

$\mathcal{A}_2$ can be taken as a subroutine of $\mathcal{B}$.

Proof: Assume that $\mathcal{B}$ gains a CDH problem example (g, g^a, g^b) in G whose order is a prime number p, he wants to calculate g^{ab}. $\mathcal{B}$ is used as $\mathcal{A}_2$'s challenger and $\mathcal{A}_2$ is a subroutine of $\mathcal{B}$. When $\mathcal{A}_2$ makes requests, $\mathcal{B}$ responds to $\mathcal{A}_2$ through the following ways.

Setup: $\mathcal{B}$ lets $l = 2(q_{DG} + q_{PS} + q_{PG} + q_{MS})$. $\mathcal{B}$ randomly chooses k uniformly between 0 and n. Suppose $\mathcal{B}$ gets the values of q_{DG}, q_{PS}, q_{PG}, q_{MS} and n where $l(n+1) < p$. Then it selects a number x' and an n-dimensional vector $\vec{x} = (x_i)(i \in [1,n])$ where $x', x_i \in Z_l$. Then, $\mathcal{B}$ randomly picks a value $y' \in Z_p$ and an n-dimensional vector $\vec{y} = (y_i)(i \in [1,n])$ where $y_i \in Z_p$. Let $W = (w_1, w_2, \ldots, w_n)$ be a warrant and all the elements of $w_i = 1$ are collected to form the set $\mathcal{W}$. $\mathcal{B}$ assigns O_1 public key $Opk_1 = (OX_1, OY_1) = (g^a, g^{y'})$. Then, g^a and $g^{y'}$ are both used as the inputs of the CDH problem.

In order to illuminate our analysis, the functions are defined just as in[8].

$$F(W) = (p-lk) + x' + \sum_{i\in\mathcal{W}} x_i \qquad K(W) = \begin{cases} 0, x' + \sum_{j\in\mathcal{W}} x_j = 0 (\bmod \\ 1, otherwise \end{cases}$$

$$J(W) = y' + \sum_{j\in\mathcal{W}} y_j$$

Some public parameters are set as follows: $u' = OY_1^{r(p-lk)+x'} \cdot g^{y'}$, $u_j = OY_1^{x_j} \cdot g^{y_j} (1 \leq j \leq n)$.

Here, $u'\prod_{i\in\mathcal{W}} u_i^{m_i} = OY_1^{J(W)} g^{J(W)}$.

Delegation Gen Query:

(a) Delegation Query of O_1. $\mathcal{B}$ selects a random number $r_1 \in Z_p^*$. If $K(W) \neq 0$, which implies $F(W) \neq 0 (\bmod p)$ [9], $\mathcal{B}$ generates the proxy capacity of O_1 by calculating $\sigma_{W1} = (\sigma_{W_{11}}, \sigma_{W_{12}})$ where

$$\sigma_{W_{11}} = OX_1^{-\frac{J(W)}{F(W)}}(u'\prod_{i=1}^{n} u_i^{m_i})^{r_1}, \quad \sigma_{W_{12}} = OX_1^{-\frac{1}{F(W)}} g^{r_1}.$$

Let $\tilde{r}_1 = r_1 - \frac{a}{F(W)}$. Then we get

$$\sigma_{W_{i1}} = OX_1^{-\frac{J(W)}{F(W)}}(u'\prod_{i=1}^{n}u_i^{w_i})^{r_1} = OY_1^{a}(u'\prod_{i=1}^{n}u_i^{w_i})^{\tilde{r}_1}$$

$$\sigma_{W_{i2}} = OX_1^{\frac{-1}{F(W)}}g^{r_1} = g^{r_1-\frac{-a}{F(W)}} = g^{\tilde{r}_1}.$$

If $K(W)=0$, the simulation process is over. Then $\mathcal{B}$ returns the information of error.

(b) Delegation Query of others. $\mathcal{B}$ can compute the delegations of other original signers as follows:

$$\sigma_{W_j} = (\sigma_{W_{j1}}, \sigma_{W_{j2}}), \sigma_{W_{j1}} = g^{ax_j ay_j}(u'\prod_{i=1}^{n}u_i^{w_i})^{r_j},$$

$$\sigma_{W_{j2}} = g^{r_j} \quad (j=2,\cdots,l)$$

PSG of Proxy Signer Query:

(a) PSG of Proxy Signer Query of O_1. If $K(W)=0$, the simulation process is over. Then $\mathcal{B}$ returns the information of error. Otherwise, $K(W)\neq 0$. Since $\mathcal{B}$ obtains the secret keys of members' in U_1, the PSG of Proxy Signer algorithm is executed to gain $\sigma_{1,i}=(\sigma_{1,i1},\sigma_{1,i2},\sigma_{1,i3})$ of every proxy signer in U_1 where $\sigma_{1,i1}=\sigma_{W_{11}}g^{px_{1,i}py_{1,i}}(Mv)^{h_i k_i}$, $\sigma_{j,i2}=\sigma_{W_{12}}$, $\sigma_{1,i3}=g^{h_i k_i}$.

(b) PSG of Proxy Signer Query of others. $\mathcal{B}$ can compute the proxy signatures $\sigma_{j,i}$ of other proxy signers in U_j as he does in response to PSG of Proxy Signer query of O_1 where $\sigma_{j,i}=(\sigma_{j,i1},\sigma_{j,i2},\sigma_{j,i3})$, $\sigma_{j,i1}=\sigma_{W_{j1}}g^{px_{j,i}py_{j,i}}(Mv)^{h_i k_i}$, $\sigma_{j,i2}=\sigma_{W_{j2}}$, $\sigma_{j,i3}=g^{h_i k_i}$ $(j=2,\cdots,l;i=1,2,\cdots,n_j)$.

PSG of Proxy Group Query:

(a) PSG of Proxy Group Query of O_1. If $K(W)=0$, the simulation process is over. Then $\mathcal{B}$ returns the information of error. Otherwise, $K(W)\neq 0$. $\mathcal{B}$ can calculate the proxy signature $\sigma_1=(\sigma_{11},\sigma_{12},\sigma_{13})$ of the proxy group U_1 where

$$\sigma_{11}=\prod_{i=1}^{n}\sigma_{1,i1}=(OX_1^{-\frac{J(W)}{F(W)}}(u'\prod_{i=1}^{n}u_i^{w_i})^{r_1})^n\cdot g^{\sum_{i=1}^{n}px_{1,i}py_{1,i}}(Mv)^{\sum_{i=1}^{n}h_i k_i},$$

$$\sigma_{12}=\prod_{i=1}^{n}\sigma_{1,i2}=(OX_1^{\frac{-1}{F(W)}}g^{r_1})^n, \sigma_{13}=\prod_{i=1}^{n_1}\sigma_{1,i3}=g^{\sum_{i=1}^{n_1}h_i k_i}.$$

(b)PSG of Proxy Group Query of others. $\mathcal{B}$ can compute the proxy signatures $\sigma_j=(\sigma_{j1},\sigma_{j2},\sigma_{j3})$ of other proxy groups $U_j(j=2,\cdots,l)$ as he does in response to O_1 PSG of Proxy Group query.

$$\sigma_{j1}=\prod_{i=2}^{n_j}\sigma_{j,i1}, \sigma_{j2}=\prod_{i=2}^{n_j}\sigma_{j,i2}, \sigma_{j3}=\prod_{i=2}^{n_j}\sigma_{j,i3}.$$

MP MS Gen Query:

Finally, the adversary $\mathcal{A}_2$ outputs a multi-proxy multi-signature $\sigma=(\sigma_1,\sigma_2,\sigma_3)$.

$$\sigma_1=\prod_{j=1}^{l}\sigma_{j1}=((OY_1^{a}(u'\prod_{i=1}^{n}u_i^{w_i})^{\tilde{r}_1})^n\cdot g^{\sum_{i=1}^{n}px_{1,i}py_{1,i}}(Mv)^{\sum_{i=1}^{n}h_i k_i}\cdot\prod_{j=2}^{l}\sigma_{j1},$$

$$\sigma_2=\prod_{j=1}^{l}\sigma_{j2}\cdot\prod_{z=1}^{m}VX_z=g^{\tilde{r}_1 n_1}\cdot\prod_{j=2}^{l}\sigma_{j2}\cdot\prod_{z=1}^{m}VX_z,$$

$$\sigma_3=\prod_{j=1}^{l}\sigma_{j3}\cdot\prod_{z=1}^{m}VY_z=g^{\sum_{i=1}^{n_1}h_i k_i}\cdot\prod_{j=2}^{l}\sigma_{j3}\cdot\prod_{z=1}^{m}VY_z.$$

The following proving process shows that the signature σ is correct.

$$e(\sigma_1\cdot\sigma_{v1}\cdot\sigma_{v2},g)=\prod_{j=1}^{l}\prod_{i=1}^{n_j}(e(OX_j,OY_j)e(PX_{j,i},PY_{j,i}))$$
$$\cdot e(u'\prod_{i=1}^{n}u_i^{w_i},\sigma_2)e(Mv,\sigma_3)$$

Forgery: Assume the simulating play is not over, $\mathcal{A}_2$ will output $\sigma^*=(\sigma_1^*,\sigma_2^*,\sigma_3^*)$ according to the message $M^*\in\{0,1\}^*$ and the warrant $W^*=(w_1^*,w_2^*,\cdots,w_n^*)$ with an advantage at least ε under the following conditions:

(1) $\mathcal{A}_2$ never sent a Delegation Gen request on W^*.

(2)$\mathcal{A}_2$ never sent a PSG of Proxy Signer request on the message M^* and the warrant W^*.

If $F(W^*)\neq 0(\mathrm{mod}\ p)$, then the forgery process is over. Else, if $F(W^*)=0(\mathrm{mod}\ p)$, then $u'\prod_{j\in\omega}u_i^{w_i^*}=g^{J(W^*)}$.

$\mathcal{B}$ can outputs

$$g^{ab}=\sigma_1^*\cdot(M^*v)^{-\sum_{j=1}^{l}\sum_{i=1}^{n_j}(h_i k_j)}$$
$$\cdot g^{-(\sum_{i=1}^{m}(px_{1,i}+py_{1,i})+\sum_{j=2}^{l}\sum_{i=1}^{n_j}(ax_j+ay_j+px_{j,i}+py_{j,i})+J(W^*)\cdot\sum_{j=1}^{l}n_j r_j)}.$$

It is the answer of the CDH problem which is issued at the beginning of this section.

The overall depiction of simulation process is accomplished. Inspired by Waters' method[9], the possibility of solving CDH problem will be figured out. Assume $W_1,W_2,\cdots,W_{q_w}$ are the warrants in above requests, but does not include W^*. We have $q_w\leq q_{DG}+q_{PS}+q_{PG}+q_{MS}$. The following conditions should be fulfilled.

$A_i: K(W_i)\neq 0(\mathrm{mod}\ l), i=1,2,\cdots,q_w$, during Delegation Gen, PSG of Proxy Signer , PSG of Proxy Group and MP MS Gen queries.

B: $F(W^*)=0(\mathrm{mod}\ p)$ during MP MS Gen query.

Therefore, the success probability for $\mathcal{B}$ is

$A_i: K(W_i)\neq 0(\mathrm{mod}\ l), i=1,2,\cdots,q_w$

B: $F(W^*)=0(\mathrm{mod}\ p)$

$$\Pr(\bigcap_{i=1}^{q_w}A_i\bigcap B)=\Pr(B)\cdot\Pr(\bigcap_{i=1}^{q_w}A_i\mid B)$$
$$\geq 1/(4(n+1)(q_{DG}+q_{PS}+q_{PG}+q_{MS}))$$

Assume the probability that $\mathcal{A}_2$ wins the game is not less than ε. So, the possibility that the CDH problem can be solved is $\varepsilon'\geq\frac{1}{4(n+1)(q_{DG}+q_{PS}+q_{PG}+q_{MS})}\varepsilon$.

The time consumption of $\mathcal{B}$ is computed according to the exponentiations and multiplications performed in q_{DG} Delegatin Gen queries , q_{PS} PSG of Proxy Signer queries, q_{PG} PSG of Proxy Group queries, q_{MS} MP MS Gen queries. The running time of every stage is shown in Table II.

TABLE II. THE RUNNING TIME OF EVERY PHASE

Stage	T_e	T_m
Setup	$2n+6$	$n+3$
DelegatinGen	O_i : 3; Others: $3(l-1)$	O_i : 2; Others: $2(l-1)$
PSGofProxySigner	O_i : 3; Others: $3(l-1)$	O_i : $3(n_i-1)$; Others: $3\prod_{j=2}^{l}(n_j-1)$
PSGofProxyGroup	O_i : 0; Others: 0	O_i : (3l-3) T_m ; Others: $3\prod_{j=2}^{l}(n_j-1)$
MPMSGen	0	$3l+2m-5$

$2n+6$ exponentiation operations and $n+3$ multiplication operations in G are required in Setup stage. $3l$ exponentiation operations and $2l$ multiplication operations in G are needed in each Delegatin Gen query. In every PSG of Proxy Signer query, $3l$ exponentiation operations and $5l-1$ multiplication operations in G are required. In PSG of Proxy Group query and MPMSGen query, $3\prod_{j=1}^{l}(n_j-1)$ and $3l+2m-5$ multiplication operations in G are needed, respectively. Therefore, the time consumption of this simulation is about

$$T' = T + (2n+6+3l\cdot q_{DG}+3l\cdot q_{PS})T_e + (n+3+2l\cdot q_{DG}+(5l-1)\cdot q_{PS}$$
$$+3\prod_{j=1}^{l}(n_j-1)\cdot q_{PG}+3(l-1)\cdot q_{MS})T_m .$$

Thus, this completes the proof.

C. Game with the Adversary $\mathcal{A}_3$

Theorem 2. The adversary $\mathcal{A}_3$ is able to obtain the forgery signature with the probability ε after performing in time T. And the number of PSG of Proxy Signer, PSG of Proxy Group and MP MS Gen are requested at most q_{PS} , q_{PG} and q_{MS} , respectively. The situation is called that $\mathcal{A}_3(T,0,q_{PS},q_{PG},q_{MS},\varepsilon)$ - breaks the MPMS scheme. If the situation happens, we can format a new algorithm $\mathcal{B}$ which can (T',ε') -break the CDH problem where $\varepsilon' \geq \dfrac{\varepsilon}{2n(q_{PS}+q_{PG}+q_{MS})}$ and

$$T' = T + (2n+6+\prod_{j=1}^{l}(3n_j-1)\cdot q_{PS})T_e + (n+3+\prod_{j=1}^{l}5n_j\cdot q_{PS}$$
$$+3(n_j-1)q_{PG}+(3l+2m-5)q_{MS})T_m .$$

$\mathcal{A}_3$ can be taken as a subroutine of $\mathcal{B}$.

Proof: There are many similarities in the proof of Theorem 1. Limited by space, we only make the distinctions between them. In the first place, $\mathcal{A}_3$ obtains the public keys of the members in G_o and U_j where $j=1,2,...,l$; besides, he achieves the private keys of them except $P_{1,1}$, who is designated the proxy right by O_1 . Thus $\mathcal{A}_3$ needn't to make the Delegation Gen requests, and is able to construct the delegation independently. In the second place, the public key of $P_{1,1}$ is set to $Ppk_1=(OX_1,OY_1)=(g^a,g^b)$ in the Setup stage, where g^a and g^b are the inputs of the CDH problem at the beginning.

V. CONCLUSIONS

It is the important issue that every original signer can designate his own proxy group which is different from others'. Yet, few people consider the property in the MPMS schemes. In this article, a secure MPMS scheme with this characteristic is construct and demonstrate. Moreover, the scheme has the properties of proxy revocation, proxy authorization and shared verification. According to the comparison, the new scheme offers higher security and better computational efficiency.

ACKNOWLEDGMENT

This work was supported by the Fund for Natural Science Research of Liaoning Province Grant #201602887 to Dong Liu.

REFERENCES

[1] Tzeng S, Yang C, Hwang M. A new multi-proxy multi-signature scheme. Technical Report No. CYUT-IM-TR-2002-004, CYUT, 2002

[2] Tzeng SF, Yang CY, Hwang MS. A Nonrepudiable Threshold Multi-Proxy Multi-Signature Scheme with Shared Verification. Future Generation Computer Systems, 2004, 20(5): 887-893

[3] YuGuang Y, QiaoYan W. Threshold proxy quantum signature scheme with threshold shared verification. Sci China Ser G-Phys Mech Astron 2008;51:1079-88

[4] Zhang Z, Cai M, Xiao G-z. An efficient threshold shared verification signature scheme and its application. Journal of China Institute Communications 2003;24:134-9

[5] Sun Y, Xu CX, Yu Y, Yang B. Improvement of a proxy multi-signature scheme without random oracles. Computer Communications 2011;34:257-63

[6] Liu ZH, Hu YP, Zhang XS, Ma H. Provably secure multi-proxy signature scheme with revocation in the standard model. Computer Communications 2011; 34:494-501

[7] Kang BY, Boyd C, Dawson E. A Novel Nonrepudiable Threshold Multi-Proxy Multi-Signature Scheme with Shared Verification. Computers & Electrical Engineering, 2009, 35(1): 9-17

[8] Huang X, Susilo W, Mu Y, Wu W. Proxy signature without random oracles. Mobile Ad-hoc and Sensor Networks 2006:473-84

[9] Waters B. Efficient identity-based encryption without random oracles. Advances in Cryptology-EUROCRYPT 2005 2005:114-27