Publicly Verifiable Secret Sharing Member-join Protocol For Threshold Signatures

Jia Yu[1]
[1]College of Information Engineering, Qingdao University, Qingdao, P. R. China
Email: qduyujia@gmail.com

, Fanyu Kong[2], Rong Hao[1], Xuliang Li[3], Guowen Li[4]
[2]Institute of Network Security, Shandong University, Jinan, P. R. China
[3]Network Center, Qingdao University, Qingdao, P. R. China
[4]School of Computer Science and Technology, Shandong Jianzhu University, Jinan, P. R. China
Email: {sdukongfanyu, hr, xll, gwl}@gmail.com

Abstract—**Publicly verifiable secret sharing (PVSS) allows not only shareholders themselves but also everyone verify the shares of a secret distributed by a dealer. It has a lot of electronic applications. In this paper, we propose a publicly verifiable member-join protocol for threshold signatures. In our proposal, a new member can join a PVSS scheme to share the secret only with the help of old shareholders. What's more, everyone besides the new member can verify the validity of the new member's share, while only the new member knows his share. Different from previous protocols, our protocol can tolerate a mobile adversary. This proposal adapts to many electronic applications. Finally, we analyze the security of our scheme.**

Index Terms—**verifiable secret sharing, publicly verifiable secret sharing, verifiable secret redistribution, verifiable encryption**

I. Introduction

A secret sharing scheme (SS) can make a secret be divided into many shares that are shared among a set of shareholders. The secret construction needs the cooperation of some qualified subset. The secret sharing scheme is composed of two phases. The first is distribution phase, in which a dealer distributes secret shares into many shareholders or shareholders jointly generate their shares by a distributed protocol. The second phase is reconstruction phase, in which some qualified subset of the shareholders reconstructs the secret by their shares. The secret sharing scheme was firstly introduced by Blakley [1] and Shamir [2] in 1979, independently. It has wide applications in distributed computations. However, the secret sharing scheme assumes that the dealer and all the shareholders are honest. If the dealer distributes false shares in distribution phase or dishonest shareholders provide false shares in reconstruction phase, the secret can't be computed correctly. The verifiable secret sharing (VSS) [3~5] aims at resolving this problem. It can verify the validity of the shares in distribution and reconstruction phases. It plays an important role in design of protocols of distributed key generation [6,7] and secure multi-party computation [8~11]. Publicly verifiable secret sharing (PVSS) [12~17] is a special VSS in which not only the shareholders but also everyone can verify whether the shares are valid or not.

Secret sharing scheme, however, can only be applied to the condition that the group of shareholders is static. If the group of shareholders is dynamic, computation of the new shares is necessary. The schemes [18] and [19] can enroll and disenroll shareholders from the access structure, respectively. Martin *et al.* [20] introduced some bounds and techniques for efficient secret redistribution schemes. A secret redistribution protocol was proposed by Desmedt and Jajodia [21], which can distribute the new shares from a group of old shareholders to another disjoint groups of new shareholders. Wong *et al.* [22] gave a protocol with verifiable ability through improving protocol [21]. In these schemes, when a new member joins a secret sharing scheme, all old shareholders have to change their shares. Refs. [23,24] proposed two protocols which can verifiably distribute a share to a new member. And old shareholders don't change their shares after distribution, which can bring great convenience to key management.

However, faced with a mobile adversary, how to publicly verifiably join a new member in a secret sharing scheme is an interesting problem. One ideal method is to set up a trusted party (dealer) always available. We can let the trusted party hold the secret and distribute a new share to the new member. Unfortunately, the trusted party is easy to become a target to be attacked by an adversary in electronic society, so it is impossible for the trusted party to be online always. We wish the share for the new member can be computed with the help of a group of old shareholders. Thus it is very important to design a publicly verifiable secret sharing member-join protocol for electronic applications. Ref. [25] proposed a publicly verifiable secret redistribution protocol, however, in this protocol all old shares needs change if a new member join the system similarly to [22]. It will bring the burden of key management. Refs. [17,26] proposed two protocols that can publicly verifiably join a new member without changing old shares in a PVSS scheme. However, they can only tolerate a static adversary. If the adversary can corrupt different players in different time points, the protocols can't get the correct result. The motivation of this paper is to put forward a protocol for threshold signatures to resolve the problem.

The rest of this paper is organized as follows. In Section 2, we introduce the preliminaries of our work including the definitions of secret sharing scheme, publicly verifiable secret sharing scheme and publicly verifiable secret sharing member-join protocol, notations and building blocks. A concrete description of our proposal is given in Section 3. In addition, we give the security theorems in section 4 and give the method of how to decide the value of m in section 5. Finally, Section 6 concludes the paper.

II. PRELIMINARIES

A. Definitions

We say access structure Γ is monotone if it follows that if $A \in \Gamma$ and $A \subseteq B$ then $B \in \Gamma$.

Definition 1. A Secret Sharing (SS) Scheme is composed by a dealer, n participants $P_1,...,P_n$, and a monotone access structure $\Gamma \subseteq 2^{\{1,...,n\}}$. There are two algorithms in SS scheme.

One is **algorithm *Share***. The dealer runs this algorithm

$$Share(s) = \{s_1, s_2, ..., s_n\}$$

to compute and distribute shares to participants $P_1,...,P_n$.

The other is **algorithm *Reconstruct***. When some participants want to reconstruct the secret, they run the algorithm having this property that

$$\forall A \in \Gamma: \quad \mathrm{Re}construct(\{s_i \mid i \in A\}) = s$$

and that for $\forall A \notin \Gamma$, it is computationally infeasible to calculate s from $\{s_i \mid i \in A\}$.

Definition 2. A Publicly Verifiable Secret Sharing (PVSS) Scheme is a SS scheme with an expanded *Share* algorithm, a *Reconstruct* Algorithm and an additional *PubVerify* algorithm that are described as following:

Algorithm *Share*: The dealer computes $S_i = E_i(s_i)$ for $1 \le i \le n$ with the encryption functions E_i, distributes the shares $s_1,...,s_n$ to $P_1,...,P_n$, and publishes $S_1,...,S_n$. Where E_i are public encryption functions.

Algorithm *Reconstruct*: When some participants want to reconstruct the secret, they run the algorithm having this property that

$$\forall A \in \Gamma: \quad \mathrm{Re}construct(\{s_i \mid i \in A\}) = s$$

and that for $\forall A \notin \Gamma$, it is computationally infeasible to calculate s from $\{s_i \mid i \in A\}$.

Algorithm *PubVerify*: This algorithm can verify the validity of all encrypted shares. It has the property that

$$\exists u \forall A \in 2^{\{1,...,n\}}:$$
$$(PubVerfiy(\{S_i \mid i \in A\}) = 1) \Rightarrow$$
$$\mathrm{Re}construct(\{D_i(S_i) \mid i \in A\}) = u$$

and $u = s$ if the dealer is honest, where D_i are decryption functions.

A PVSS scheme is called non-interactive if algorithm *PubVerify* requires no interaction with the dealer at all.

Definition 3. Publicly Verifiable Secret Sharing Member-join (PVSSMJ) Protocol is composed by a dealer, n participants $P_1,...,P_n$, and a monotone access structure $\Gamma \subseteq 2^{\{1,...,n\}}$. This protocol consists of two phases:

The first is **the secret distribution phase**. In this phase, the dealer runs this algorithm $Share(s) = \{s_1, s_2, ..., s_n\}$ to compute and publicly verifiably distribute shares to participants $P_1,...,P_n$. The secret s is shared by a (t,n) secret sharing scheme.

The second is **the member-join phase**. In this phase, a new member firstly selects a group of shareholders $A \in \Gamma$ to help him generate new share. All shareholders $P_i \in A$ blind shares s_i using functions $Blind(s_i) = s_i'$. And then publicly verifiably send the blinded shares s_i' to the new member. The validity of the blinded shares can be verified by everyone. The new member selects a group B of t shareholders who provide correct blinded shares. Using a *construct* algorithm, the new member computes his share $Construct(\{s_i' \mid i \in B\}) = s_{n+1}$ according to these blinded shares s_i'. Finally, the secret is shared by a $(t,n+1)$ secret sharing scheme.

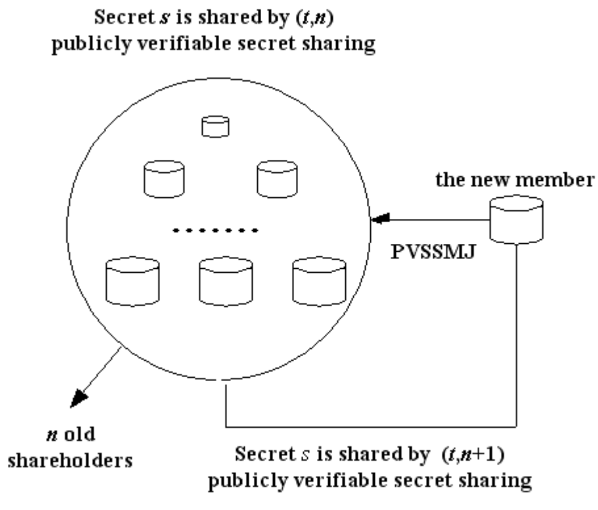

Figure 1 The PVSSMJ Protocol

B. Notations

p and q are primes *s.t.* $q \mid p-1$. Let *G* denote a group with prime order *p* and *g* be a generator of group *G*. Let $h \in Z_p^*$ be an element of order *q*. The secret s is shared by a (t,n) publicly verifiable secret sharing scheme among n participants $P_1, P_2, ..., P_n$. The new member to join the system is P_{n+1}.

C. Building Blocks

(1) Verifiable Secret sharing scheme

The shared secret k is in Z_p. Randomly choose a polynomial

$$f(x) = a_0 + \sum_{j=1}^{t-1} a_j x^j \pmod p \in Z_p[x] \quad (1)$$

where $a_0 = k$, $a_j \in_R Z_p$

Compute the secret shares

$$s_i = f(i) = a_0 + \sum_{j=1}^{t-1} a_j i^j \pmod p \quad (2)$$

for each member $P_i \in P$. At the same time, the dealer broadcasts commits

$$\varepsilon_j = g^{a_j}, \ (0 \le j < t) \quad (3)$$

Member P_i use Eq.

$$g^{s_i} = \prod_{j=0}^{t-1} \varepsilon_j^{i^j} \quad (4)$$

to verify whether s_i is right or not.

The secret reconstruction: According to some subset *B* (|*B*|=*t*), compute

$$k = \sum_{P_i \in B} C_{Bi} s_i \pmod p \quad (5)$$

where $C_{Bi} = \prod_{P_j \in B \setminus \{P_i\}} \frac{j}{(j-i)} \pmod p$.

According to some subset *B* (|*B*|=*t*), any shares for $P_j \notin B$ can be computed by the following Eq.

$$s_j = \sum_{P_i \in B} C_{Bi}(j) s_i \pmod p \quad (6)$$

where $C_{Bi}(j) = \prod_{P_l \in B \setminus \{P_i\}} \frac{j-l}{i-l}$.

(2) Verifiable Encryption of Discrete Logarithms $VEDL(D, P_i, s_i, E_i)$ [13]

Where D is a sender, P_i is a receiver, and s_i is a secret that is encrypted by the sender *D* and verifiably sent to the receiver P_i. E_i is a commit of s_i satisfying $E_i = g^{s_i}$.

In this protocol, receiver P_i selects $x_i \in_R Z_p$ as her secret key, and then publishes her public key $y_i = h^{x_i}$. The sender *D* distributes an encrypted secret s_i to P_i while everyone can verify the validity of the encrypted s_i. The commit $E_i = g^{s_i}$ is published. Let $H: \{0,1\}^* \to \{0,1\}^l$ be a collision-resistant hash function.

(1) The sender *D* encrypts s_i by a variation of ElGamal encryption algorithm:

She selects $l_i \in_R Z_q$, computes

$$\gamma_i = h^{l_i} \pmod p \quad (7)$$

$$\delta_i = s_i^{-1} y_i^{l_i} \pmod p \quad (8)$$

and publishes (γ_i, δ_i) as the ciphertext of value s_i. And then selects $w_k \in Z_q$, $k = 1,2,...,l$, computes and broadcasts

$$T_{h,i,k} = h^{w_k} \pmod p \quad (9)$$

$$T_{g,i,k} = g^{(y_i^{w_k})} \quad (10)$$

where $i = 1,2,...,n$.

She computes

$$c_i = H(g \parallel h \parallel \gamma_i \parallel \delta_i \parallel T_{h,i,1} \parallel T_{h,i,2} \parallel ... \parallel T_{h,i,l} \parallel T_{g,i,1} \parallel T_{g,i,2} \parallel ... \parallel T_{g,i,l}) \quad (11)$$

(2) Let $c_{i,k}$ denote the *k*-th bit of c_i. The dealer computes $r_{i,k} = w_k - c_{i,k} l_i$, where $k = 1,2,...,l$ and publishes $\Pr oof_D = (c_i, r_{i,1}, ..., r_{i,l})$.

(3) P_i decrypts (γ_i, δ_i) to get

$$s_i = \gamma_i^{x_i} \cdot \delta_i^{-1} \pmod p \quad (12)$$

and verifies the following equation

$$E_i = g^{s_i} \quad (13)$$

holds or not. If it holds, P_i believes her share is correct. Otherwise, publishes s_i and broadcasts a complaint against the dealer.

(4) Everyone P_j can check the validity of each share s_i $(i \ne j)$ by verifying

$$T_{h,i,k} = h^{r_{i,k}} \gamma_i^{c_{i,k}} \quad (14)$$

$$T_{g,i,k} = (g^{1-c_{i,k}} E_i^{c_{i,k}\delta_i})^{y_i^{r_{i,k}}} \quad (15)$$

And by verifying whether equation (11) holds. If it holds, then believes s_i is correct. Otherwise, generates a complaint against the dealer.

Theorem 1 Under the assumption that computing discrete logarithms in *G* is infeasible, and that breaking the ElGamal cryptosystem is hard, computing s_i from E_i and $(h^{l_i}, s_i^{-1} y_i^{l_i})$ is at least as hard as solving the Decision-Diffe-Hellman problem to the base *h* in Z_p^*.

Theorem 2 The described non-interactive protocol above is perfectly zero-knowledge.

The above protocol and theorems are taken from [13] with slight modification.

III. THE PROPOSED PVSSMJ PROTOCOL

The protocol is composed of two phases. The first phase is secret distribution phase. In this phase, a dealer publicly verifiably distributes the shares of a secret into a group of shareholders $P_1, P_2, ..., P_n$. This procedure is similar to Stadler's PVSS [13]. The second phase is member-join phase that is the core phase of our protocol. In this phase, a group of old shareholders that are selected by the new member help the new member publicly verifiably generate a share. The both phases are described as follows:

① The Secret Distribution Phase

(1) The dealer D randomly selects a polynomial

$$f(x) = s + \sum_{i=1}^{t-1} a_i x^i \in Z_p[x] \qquad (16)$$

and computes $s_i = f(i)$, $i = 1, 2, ..., n$. The dealer broadcasts g^s , $g^{a_i} (i = 1, 2, .., t-1)$.

(2) The dealer encrypts each s_i : She selects $l_i \in_R Z_q$, computes

$$\gamma_i = h^{l_i} (\bmod p) \qquad (17)$$

$$\delta_i = s_i^{-1} y_i^{l_i} (\bmod p) \qquad (18)$$

and publishes (γ_i, δ_i) as the ciphertext of s_i . And then selects $w_k \in Z_q$, $k = 1, 2, ..., l$, computes and broadcasts

$$T_{h,i,k} = h^{w_k} (\bmod p) \qquad (19)$$

$$T_{g,i,k} = g^{(y_i^{w_k})} \qquad (20)$$

where $i = 1, 2, ..., n$.

She computes

$$c_i = H(g \| h \| \gamma_i \| \delta_i \| T_{h,i,1} \| T_{h,i,2} \| ... \| T_{h,i,l} \| T_{g,i,1} \| T_{g,i,2} \| ... \| T_{g,i,l}) \qquad (21)$$

(3) Let $c_{i,k}$ denote the k-th bit of c_i. The dealer computes $r_{i,k} = w_k - c_{i,k} l_i$, where $k = 1, 2, ..., l$ and publishes $\Pr oof_D = (c_i, r_{i,1}, ..., r_{i,l})$.

(4) Each participant $P_i (i = 1, 2, ..., n)$ decrypts (γ_i, δ_i) to get

$$s_i = \gamma_i^{x_i} \cdot \delta_i^{-1} (\bmod p) \qquad (22)$$

and verifies the following equation

$$g^{s_i} = g^s \prod_{j=1}^{t-1} (g^{a_j})^{i^j} \qquad (23)$$

holds or not. If it holds, P_i believes his share is correct and sets $E_i = g^{s_i}$. Otherwise, publishes s_i and broadcasts a complaint against the dealer.

(5) Each participant $P_j (j = 1, 2, ..., n)$ checks the validity of share s_i $(i \neq j)$. She computes

$$E_i = g^s \prod_{j=1}^{t-1} (g^{a_j})^{i^j} \qquad (24)$$

$$T_{h,i,k} = h^{r_{i,k}} \gamma_i^{c_{i,k}} \qquad (25)$$

$$T_{g,i,k} = (g^{1-c_{i,k}} E_i^{c_{i,k} \delta_i})^{y_i^{r_{i,k}}} \qquad (26)$$

And then verifies whether equation (21) holds. If it holds, then believes s_i is correct. Otherwise, generates a complaint against the dealer.

② Member-join Phase

When a new member P_{n+1} asks for joining the system. Firstly, she randomly selects a secret $x_{n+1} \in_R Z_p$ and publishes the commit $y_{n+1} = h^{x_{n+1}}$. And then she randomly chooses $m (t \le m \le 2t-1)$ active members from $P_i (i = 1, 2, ..., n)$. W.l.o.g, assume the players $P_1, P_2, \ldots, P_m$ are selected. Let $A = \{P_1, P_2, ..., P_m\}$ and set $F = \varnothing$.

(1) Each $P_i (i = 1, 2, ..., m)$ selects a random polynomial

$$f_i(x) = \sum_{l=0}^{t} a_{il} x^l (\bmod p) \qquad (27)$$

to compute $u_{ij} = f_i(j)$, $j = 1, 2, ..., m$ and $u_{i(n+1)} = f_i(n+1)$.

Each P_i broadcasts message: $g^{a_{il}}$ $(l = 0, 1, ..., t)$, $\varepsilon_{ij} = g^{u_{ij}}$ ($j = 1, 2, ..., m, n+1$).

(2) And then each member $P_i (i = 1, 2, ..., m)$ selects $l_{i,j} \in_R Z_q$, computes

$$\gamma_{i,j} = h^{l_{i,j}} (\bmod p) \qquad (28)$$

$$\delta_{i,j} = u_{i,j}^{-1} y_j^{l_{i,j}} (\bmod p) \qquad (29)$$

and broadcasts $(\gamma_{i,j}, \delta_{i,j})$ as the ciphertext of the share $u_{i,j}$.

She selects $w_k \in Z_q$, $k = 1, 2, ..., l$, computes and broadcasts

$$T_{h,i,j,k} = h^{w_k} (\bmod p) \qquad (30)$$

$$T_{g,i,j,k} = g^{(y_j^{w_k})} \qquad (31)$$

where $j = 1, 2, ..., n$.

She computes

$$c_{i,j} = H(g \| h \| \gamma_{i,j} \| \delta_{i,j} \| T_{h,i,j,1} \| T_{h,i,j,2} \| ... \| T_{h,i,j,l} \| T_{g,i,j,1} \| T_{g,i,j,2} \| ... \| T_{g,i,j,l}) \qquad (32)$$

(3) Let $c_{i,j,k}$ denote the k-th bit of $c_{i,j}$. P_i computes $r_{i,j,k} = w_{i,k} - c_{i,j,k} l_{i,j}$, where $k = 1, 2, ..., l$, and broadcasts $\Pr oof_D = (c_{i,j}, r_{i,j,1}, ..., r_{i,j,l})$.

(4) Each $P_j (j = 1, 2, ..., m)$ decrypts $(\gamma_{i,j}, \delta_{i,j})$ to get

$$u_{i,j} = \gamma_{i,j}^{x_j} \cdot \delta_{i,j}^{-1} (\bmod p) \qquad (33)$$

and verifies the following equation

$$g^{u_{ij}} = g^{a_{i0}} \prod_{r=1}^{t-1} (g^{a_{ir}})^{j^r} \qquad (34)$$

holds or not. If it doesn't hold, abort.

(5) Other members verify the validity of value $u_{i,j}$ $(j \neq i)$. They compute

$$T_{h,i,j,k} = h^{r_{i,j,k}} \gamma_i^{c_{i,j,k}} \quad (35)$$

$$T_{g,i,j,k} = (g^{1-c_{i,j,k}} \varepsilon_{i,j}^{c_{i,j,k}\delta_{i,j}})^{y_j^{r_{i,j,k}}} \quad (36)$$

And then verifies whether equation (32) holds. If it holds, then believes $u_{i,j}$ is correct. Otherwise, not. If more than t-1 members in set A believe that u_{ij} is invalid, set $F = F \cup \{P_i\}$.

(6) Each $P_j (j = 1,2,...,m)$ computes

$$s_j' = s_j + \sum_{i\in A-F} u_{ij} (\bmod p) \quad (37)$$

She selects $l_{j,n+1} \in_R Z_q$, computes

$$\gamma_{j,n+1} = h^{l_{j,n+1}} (\bmod p) \quad (38)$$

$$\delta_{j,n+1} = s_j'^{-1} y_j^{l_{j,n+1}} (\bmod p) \quad (39)$$

and broadcasts $(\gamma_{j,n+1}, \delta_{j,n+1})$ as the ciphertext of share s_j'. Member P_j computes and broadcasts $E_j' = g^{s_j'}$.

She selects $w_k \in Z_q$, $k = 1,2,...,l$, computes and broadcasts

$$T_{h,j,n+1,k} = h^{w_k} (\bmod p) \quad (40)$$

$$T_{g,j,n+1,k} = g^{(y_{n+1}^{w_k})} \quad (41)$$

She computes

$$c_{j,n+1} = H(g \| h \| \gamma_{j,n+1} \| \delta_{j,n+1} \| T_{h,j,n+1,1} \| T_{h,j,n+1,2} \| \dots \| T_{h,j,n+1,l} \| T_{h,j,n+1,l}) \quad (42)$$

(7) Let $c_{j,n+1,k}$ denote the k-th bit of $c_{j,n+1}$. P_j computes $r_{j,n+1,k} = w_{j,k} - c_{j,n+1,k} l_{j,n+1}$, where $k = 1,2,...,l$, and publishes $\Pr oof_D = (c_{j,n+1}, r_{j,n+1,1},..., r_{j,n+1,l})$.

(8) New member P_{n+1} decrypts $(\gamma_{j,n+1}, \delta_{j,n+1})$ to get

$$s_j' = \gamma_{j,n+1}^{x_j} \cdot \delta_{j,n+1}^{-1} (\bmod p) \quad (43)$$

and verifies the following equation

$$g^{s_j'} = E_j \prod_{i\in A-F} \varepsilon_{ij} \quad (44)$$

holds or not. If it holds, then believes s_j' is correct. If more than t-1 members give the correct s_j', then P_{n+1} selects a set B with t members who give the right s_j'. She computes her share

$$s_{n+1} = \sum_{i\in B} C_{Bi}(n+1) s_i' - \sum_{i\in A-F} u_{i(n+1)} (\bmod p) \quad (45)$$

where $C_{Bi}(n+1) = \prod_{P_j \in B\setminus\{P_i\}} \frac{n+1-j}{i-j}$.

Otherwise, increase the value of m and go to step (1).

(9) Other members verify the validity of value s_j'. They compute

$$T_{h,j,n+1,k} = h^{r_{j,n+1,k}} \gamma_i^{c_{j,n+1,k}} \quad (46)$$

$$T_{g,j,n+1,k} = (g^{1-c_{j,n+1,k}} (E_j \prod_{i\in A-F} \varepsilon_{ij})^{c_{j,n+1,k}\delta_{j,n+1}})^{y_{n+1}^{r_{j,n+1,k}}} \quad (47)$$

And then verify whether equation (42) holds. If it holds, then believe s_j' is correct.

IV. Security Theorems

Theorem 3 If the members that the new member P_{n+1} selects to help her to generate the share are honest, then member P_{n+1} can get the right share by executing the presented protocol.

Proof .

It is because:

$$\begin{aligned} s_{n+1} &= \sum_{i\in B} C_{Bi}(n+1) s_i' - \sum_{i\in A-F} u_{i(n+1)} \\ &= \sum_{i\in B} C_{Bi}(n+1)(s_i + \sum_{j\in A-F} u_{ji}) - \sum_{i\in A-F} u_{i(n+1)} \\ &= \sum_{i\in B} C_{Bi}(n+1) s_i + \sum_{j\in A-F} \sum_{i\in B} C_{Bi}(n+1) u_{ji} \\ &\quad - \sum_{i\in A-F} u_{i(n+1)} \\ &= \sum_{i\in B} C_{Bi}(n+1) s_i + \sum_{i\in A-F} \sum_{j\in B} C_{Bj}(n+1) u_{ij} \\ &\quad - \sum_{i\in A-F} u_{i(n+1)} \\ &= s_{n+1} + \sum_{i\in A-F} u_{i(n+1)} - \sum_{i\in A-F} u_{i(n+1)} \\ &= s_{n+1} \end{aligned}$$

Theorem 4 The dishonest participants can be discovered in the proposed protocol. And when $n \ge 2t-1$, even if an adversary can corrupt t-1 old shareholders at one time-period, the new member still can get the right share.

Proof .

In secret distribution phase, the participating shareholders can verify whether the shares distributed by the dealer are right or not by verifying equation (23) in step (4) and equation (21) in step (5).

In member-join phase, a dishonest participating shareholder can deceive other members as follows:

Case 1: She can give other shareholders false value (values) in step (1) such as u_{ij}, or $g^{a_{il}}$, or $\varepsilon_{ij} = g^{u_{ij}}$. It can be discovered by verifying equation (34) in step (4) and equation (32) in step (5).

Case 2: She can give P_{n+1} false s_j' or other shareholders false E_j'. However, it can be discovered by verifying equation (44) in step (8) and equation (42) in step (9).

Therefore, the dishonest participating shareholders can be discovered in the proposed protocol.

When $n \ge 2t-1$, if fewer than t members give the correct s_j', the value of m will be increased up to 2t-1. At that time, even if an adversary can corrupt t-1 old shareholders, there are still no fewer than t honest shareholders. So these participants can help the new member get right share.

Lemma 1. For any polynomial $f(x)=k+\sum_{i=1}^{t-1} a_i x^i \pmod p$, *s.t.* $f(i)=s_i$, $(i\in\{1...t-1\})$, when taken as input $s_1,s_2,...,s_{t-1}$ and g^k, there is an algorithm A that can compute $g^{a_1},g^{a_2},...,g^{a_t}$ and an algorithm B that can compute $g^{s_k}\ (t\le k\le n)$.

Proof.
We define the polynomial in another format:

$$f(x)=\sum_{i=0}^{t-1} s_i \prod_{j\in\{0...t-1\}\backslash\{i\}} \frac{x-j}{i-j} = \sum_{i=0}^{t-1} \frac{s_i}{\prod_{j\in\{0...t-1\}\backslash\{i\}} i-j} \prod_{j\in\{0...t-1\}\backslash\{i\}} (x-j),$$

where $s_0=k$.

Thus the coefficient of x^k is $a_k=\sum_{i=0}^{t-1}\frac{s_i}{\prod_{j\in\{0...t-1\}\backslash\{i\}}(i-j)}\lambda_{k,i}$, where $k\in\{1...t-1\}$, and $\lambda_{k,i}$ are computable constants. Now we construct an algorithm A to compute as follows for all $k=1...t-1$:

$$g^{a_k}\equiv g^{\sum_{i=0}^{t-1}\frac{s_i}{\prod_{j\in\{0...t-1\}\backslash\{i\}}(i-j)}\lambda_{k,i}}\equiv \prod_{i=0}^{t-1} g^{\frac{s_i}{\prod_{j\in\{0...t-1\}\backslash\{i\}}(i-j)}\lambda_{k,i}}\equiv g^{s_0\lambda_{k,0}\frac{(-1)^{t-1}}{(t-1)!}}\prod_{i=1}^{t-1} g^{\frac{s_i}{\prod_{j\in\{0...t-1\}\backslash\{i\}}(i-j)}\lambda_{k,i}}\equiv (g^k)^{\lambda_{k,0}\frac{(-1)^{t-1}}{(t-1)!}}\prod_{i=1}^{t-1}(g^{s_i\lambda_{k,i}})^{\frac{1}{\prod_{j\in\{0...t-1\}\backslash\{i\}}(i-j)}}$$

Algorithm B is easy to be constructed as follows: Lets $s_0=k$, for all $t\le k\le n$:

$$g^{s_k}\equiv g^{\sum_{i=0}^{t-1}(\prod_{j\in\{0...t-1\}\backslash\{i\}}\frac{k-j}{i-j})\cdot s_i}\equiv g^{\sum_{i=1}^{t-1}(\prod_{j\in\{0...t-1\}\backslash\{i\}}\frac{k-j}{i-j})\cdot s_i}\cdot(g^k)^{\prod_{j\in\{0...t-1\}}\frac{j}{j-i}}$$

Above theorem and proof is taken from [27].

Theorem 5 The proposed protocol satisfies that:
(1) If an adversary corrupts t-1 members, she can't get any useful information about the secret and other members' shares in the protocol.
(2) The new member P_{n+1} can't get any information about the shares of old shareholders in the protocol.

Proof.
(1) From theorems 1 and 2, we can know that the verifiable encryption of discrete logarithms will not leak any useful information about the shares. W.l.o.g, assume the adversary corrupts members $P_1,P_2,...,P_{t-1}$.

In secret distribution phase, except the information from the verifiable encryption protocol, she knows information including $s_1,s_2,...,s_{t-1}$, g^s, $g^{a_i}(i=1,2,..,t-1)$. From lemma 1, we can know $g^{a_i}(i=1,2,..,t-1)$ will not expose any useful information about the secret and other members' shares in the protocol.

In member-join phase, the adversary knows $g^{a_{il}}(i=0,1,...,m;l=0,1,...,t-1)$, $\varepsilon_{ij}=g^{u_{ij}}$ ($i=1,2,...,m;$ $j=1,2,...,m,n+1$), $u_{ij}(i=1,2,...,m;j=1,...,t-1)$, $s_j'(j=1,...,t-1)$, and the ciphertext $(\gamma_{i,j},\delta_{i,j})$, $(\gamma_{j,n+1},\delta_{j,n+1})$, where $(i=1,2,...,m;$ $j=1,2,...,m)$. Because $u_{ij}(i=1,2,...,m;j=1,...,t-1)$ are random and independent of the secret and other members' shares, they will not expose useful information through the message. From lemma 1, $g^{a_{il}}(i=0,1,...,m;l=0,1,...,t-1)$, $\varepsilon_{ij}=g^{u_{ij}}$ ($i=1,2,...,m;$ $j=1,2,...,m,n+1$) can be computed from $u_{ij}(i=1,2,...,m;$ $j=1,...,t-1)$. Furthermore, the secret and other members' shares cannot be computed through $s_j'(j=1,...,t-1)$ according to the property of secret sharing. Therefore, if an adversary corrupts t-1 members, she can't get any useful information about the secret and other members' shares in the protocol.

(2) What the new member P_{n+1} gets from the old shareholders are values s_j' and $\Pr oof_D=(c_{j,n+1},r_{j,n+1,1},...,r_{j,n+1,l})$, $(j=1,2,...,t)$. Because s_j' is random and independent of share s_j of shareholder P_j, and $\Pr oof_D$ has no relation to s_j, new member P_{n+1} can't get any information about the shares of old shareholders in the scheme.

V. How to Decide the Value of M

m is a variable value between t and 2t-1. If m is chosen as t, then the protocol has to restart from step (1) when a participant is corrupted, however, it needs very few communication data and interactions when all participants are honest. If m is chosen as 2t-1, the protocol will never restart even if t-1 participants are corrupted, however, it needs many communication data and interactions when all participants are honest.

Therefore, the value of *m* is decided by actual circumstance. If participants are not easy to be corrupted, *m* should be chosen as a smaller value. Otherwise, *m* should be chosen as a lager value.

How much should *m* be increased when the protocol needs to be restarted in step (8)? Similarly to what we have discussed above, if participants are not easy to be corrupted, *m* should be increased slightly. Otherwise, *m* should be increased greatly. A proposed method in common circumstance is as follows: Firstly, let the value of *m* equate *t*; if the protocol needs to be restarted in step (8), we then increase *m* up to 2*t*-1 directly. Therefore the protocol assures to be finished by twice execution at most. From above mentioned, the choice of *m* is very important for the efficiency of the protocol.

When *m* is chosen as 2*t*-1, the proposed scheme can tolerate a mobile adversary as long as the periodical operation of refreshing shares is added to the scheme. It is because more than *t*-1 members being honest in each period can recover the secret and the dishonest members will be rebooted to remove the control of the mobile adversary. It is impossible for scheme [26].

VI. Conclusions

In this paper, we propose a publicly verifiable secret sharing member-join protocol for threshold signatures. This protocol solves the problem of how to dynamically publicly verifiably join members without changing old shares even if it is faced to mobile adversary. It is especially useful in many electronic applications including key-escrow systems, electronic voting, anonymity-revocation in e-cash systems and so on. It also is applied to threshold signatures to make schemes more flexibly.

Acknowledgment

This research is supported by Natural Science Foundation of China (60703089), the National High-Tech R & D Program (863 Program) of China (2006AA012110) and National Cryptologic Development Foundation of China.

References

[1] G. R. Blakley, “Safeguarding cryptographic keys, ” In Proc. AFIPS 1979 National Computer Conference. AFIPS, 1979, pp. 313-317.

[2] A. Shamir, “How to Share a Secret,” Communications of the ACM, vol. 22, no. 11, pp. 612-613, 1979.

[3] B. Chor, S. Goldwasser, S. Micali., and B. Awerbuch, “Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults, ” In: Proc. 26th IEEE Symposium on Foundations of Computer Sciences(FOCS’85), 1985, pp. 383-395.

[4] P. Feldman, “A Practical Scheme for Non-Interactive Verifiable Secret Sharing, ” In Proc. 28th Annual FOCS, 1987, pp. 427-437.

[5] T.P. Pedersen, “Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing,” In: J. Feigenbaum ed., Advances in Cryptology-Crypto’91 proceedings, 1992, pp. 129-140.

[6] Y. Frankel, P. D. Mackenzie, and M. Yung, “Robust efficient distributed RSA-key generation,” In Proceedings of the 30th Annual ACM Symposium on the Theory of Computing (STOC’98), 1998, pp. 663-672.

[7] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, “Secure distributed key generation for discrete-log based cryptosystems,” Advances in Cryptology-Eurocrypt’99, LNCS 1592, J. Stern ed., 1999, pp. 295-310.

[8] A. C. Yao, “Protocols for secure computations,” In Proc. 23rd IEEE Symp. on the Foundation of Computer Science, 1982, pp. 160-164.

[9] O. Goldreich, S. Micali, and A. Wigderson, “How to Play Any Mental Game,” In: Proc. 19th ACM Symposium on the Theory of Computing (STOC’87), 1987, pp. 218-229.

[10] D. Chaum, C. Crepeau, and I. Damgard, “Multiparty unconditionally secure protocols, ” In Proc. 20th ACM Symp. On the Theory of Computing, 1988, pp.11-19.

[11] S. Goldwasser and L. Levin, “Fair computation of general functions in presence of immoral majority,” In Advances in Cryptology-CRYPTO ’ 90, A. Menezes and S. Vanstone eds., 1990, pp. 77-93.

[12] B. Schoenmakers, “A simple Publicly Verifiable Secret Sharing Scheme and its Application to Electronic Voting,” Advances in Cryptology-Crypto’99, M. Wiener ed., 1999, pp. 148-164.

[13] M. Stadler, “Public verifiable secret sharing, ” Advances in Cryptology- EUROCRYPT’96, U. Maurer ed., 1996, pp. 190-199.

[14] E. Fujisaki, and T. Okamoto, “A practical and provably secure scheme for publicly verifiable secret sharing and its applications,” Advances in Cryptology-Eurocrypt’98, 1998, pp. 32-47.

[15] F. Boudot, and J. Traore, “Efficient Publicly Verifiable Secret Sharing Schemes with Fast or Delayed Recovery, ” 2nd International Conference on Information and Communication Security, 1999, pp. 88-102.

[16] A. Young, and M Yung. “A PVSS as Hard as Discrete Log and Shareholder Separability,” Advances in 4th International Workshop on Practice and Theory in Public Key Cryptosystems, 2001, pp. 287-299.

[17] J. Yu, F. Y. Kong, R. Hao, “Publicly Verifiable Secret Sharing with Enrollment Ability, ” In the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007, pp. 194--199.

[18] C. Cachin, “On-line secret sharing, ” Proc. Of the 5th IMA Conf. On Cryptography and Coding, 1995, pp. 90-198.

[19] B. Blakley, G. R. Blakley, A. H. Chan, and J. L. Massey, “Threshold schemes with disenrollment,” Proc. Of CRYPTO’1992, the 12th Ann. Intl. Cryptology Conf, 1992, pp. 540-548

[20] K. M. Martin, R. S. Naini, and H. Wang, “Bounds and Techniques for Efficient Redistribution of Secret Shares to New Access Structures, ” Comput. J. vol. 42, no. 8, pp. 638-649, 1999.

[21] Y. Desmedt and S. Jajodia, “Redistributing secret shares to new access structures and its application,” Technical Report ISSE TR-97-01, George Mason University, Fairfax, VA, 1997.

[22] T. M. Wong, C. X. Wang, and J. M. Wing, “Verifiable secret redistribution for archive systems,” Proc. of the 1st International IEEE Security in Storage Workshop, 2002, pp. 94-105.

[23] X. Li, and M. X. He. “A protocol of member-join in a secret sharing scheme, ” In Proc. of the 2th information security practice and experience, 2006, pp. 134-41.
[24] J. Yu, D. X. Li, and Y. L. Fan. “Verifiable secret redistribution protocol based on additive sharing,” Journal of Computer Research and Development, vol. 43, no. 1, pp. 23-27, 2006. (in Chinese).
[25] Z. W. Tan, and Z. J. Liu, “Publicly Verifiable Secret Redistribution for Threshold Secret Sharing Scheme,” Journal of the Graduate School of the Chinese Academy of Sciences, Vol.21 No.2, pp. 210-217, 2004.
[26] J. Yu, F. Y. Kong, R. Hao, and X. L. Li. “How to Publicly Verifiably Expand a Member without Changing Old Shares in a Secret Sharing Scheme,” 2008 Pacific Asia Workshop on Intelligence and Security Informatics (PAISI 2008), 2008, pp. 138-148.
[27] J. Yu, F. Y. Kong, and D. X. Li. Verifiable Secret Redistribution for PSS Schemes. The 2nd Information Security Practice and Experience Conference (ISPEC 2006). Journal of Shanghai Jiaotong University (Science), Vol. E-11, No. 2, pp. 236~241,.2006.

Jia Yu was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer, an associate professor of computer science in the College of Information Engineering at Qingdao University, China, in 2006 and 2007, respectively. He is currently an associate professor in the College of Information Engineering at Qingdao University, China. His research interests include encryption, digital signature, cryptographic protocol and network security.

Dr. Yu currently is a member of Chinese Association for cryptologic Research and Chinese Computer Federation.

Fanyu Kong was born in China in 1978. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer of computer science in the institute of Network Security at Shandong University, China, in 2006. He is currently a fellow in the institute of Network Security at Shandong University, China. His research interests include cryptography and network security.

Dr. Kong currently is a member of Chinese Association for cryptologic Research.

Rong Hao was born in China in 1976. He received the BS, MS degrees in computer science from Jinan University and Shandong University, Shandong, China, in 1998 and 2006, respectively.

She became a lecturer of computer science in the College of Information Engineering at Qingdao University, China, in 2006. She is currently a fellow in the College of Information Engineering at Qingdao University, China. Her research interests include cryptography and network security.