Strategies of Proactive (k, n) Threshold Secret Sharing and Applications in a Secure Message Exchange System

Shiuh-Jeng Wang[1,*] Yuh-Ren Tsai[2] Pin-You Chen[2]

[1] Department of Information Management

Central Police University

TaoYuan 33304, Taiwan

[2] Institute of Communications Engineering

National Tsing Hua University

Hsinchu 30013, Taiwan

sjwang@mail.cpu.edu.tw

Received 8 November 2007; Revised 18 December 2007; Accepted 12 January 2008

Abstract. A secret sharing scheme protects the secret by distributing it to a group of participants (nodes), and it allows for some groups of participants to collaborate together to reconstruct this secret. With regard to the (k, n) threshold scheme, k out of n or more participants can reconstruct the secret. The proactive secret sharing scheme periodically updates the distributed secret (shadow). Based on such a proactive secret sharing scheme, we propose a completely mobile proactive secret sharing scheme, which not only allows for the changing of participant number n, but also arbitrarily changes the threshold k. Furthermore, we apply our technical idea to a specified project exploitation in the real commerce transaction systems to fulfill a trust-based delivery.

Keywords: Threshold secret sharing, proactive secret sharing, sharing polynomial, decision-making, distributed certificate authority

1 Introduction

A secret sharing scheme protects a secret by distributing it to a group of participants (called nodes), and are held in by the nodes so the authorized group can collaborate together to reconstruct the secret. In the (k, n)-threshold scheme, we generate n shares (called shadows) and distribute them to n different nodes, at which point k or more nodes can reconstruct the secret. Thereby an intruder would need to compromise more than k-1 nodes to figure out the secret. The first (k, n)-threshold scheme was proposed by Shamir [1] and Blakley [2], and in order to prevent an inconsistent shadow from destroying the system, Feldman [3] and Pederson [4] expanded on this idea by also implementing a verifiable secret sharing (VSS) system based on Shamir's scheme. In the VSS scheme, nodes can verify any shadow that they have received.

For an important or long-term secret, the original (k, n)-threshold scheme may be insufficient. Herzberg et al. [5] proposed a proactive secret sharing scheme (PSS) based on Shamir's scheme. In the PSS scheme, the nodes update the shadows periodically, and the shadows from the previous periods are invalid for the current period and any subsequent period. Any intruder would need to compromise more than k-1 nodes in a single period to obtain the secret.

Herzberg et al.'s PSS scheme provides a recovery protocol, which k out of n nodes can collaborate on to help the node that loses its shadow by reconstructing it. This recovery protocol is a good method for generating new shadows for new, joining nodes. By including this protocol, we can allow the membership of the group of nodes to change.

Based on Herzberg et al.'s scheme, we propose an extended (k, n)-threshold scheme, which can not only change the membership of the group of nodes n, but the threshold k. For this adaptive scheme, we can change the threshold according to the amount of nodes to achieve complete mobility. The method of changing the threshold is also proposed in the Schultz's scheme [6]. Schultz's scheme reduces the threshold by using "virtual nodes" that hold shadows publicly, and increases the threshold by increasing the degree of the sharing polynomial. This method however has some limits, which will be discussed in Section 4.

The paper is organized as follows: In Section 2, we describe some methods that were used in our scheme. Our proposed protocols are introduced in Section 3. Some discussions are shown in Section 4. The applied scene with our scheme is given in Section 5. Finally, the conclusion is given in Section 6.

* Correspondence author

2 Preliminaries

In this section, we briefly describe the methods, such as Shamir's secret sharing scheme [1], Feldman's verifiable secret sharing scheme [3], and Herzberg et al.'s proactive secret sharing scheme [5], related to our work to guide us in the deployment of our contribution presentations.

2.1 Shamir's Secret Sharing Scheme

Shamir's secret sharing scheme is a (k, n)-threshold scheme. In the initial state, a dealer chooses a large prime number q, and makes q public. The following calculation is performed on Z_q: A secret, $s \in Z_q$ exists, so the dealer randomly generates a (k-1)th degree polynomial, in Z_q:

$$f(x) \equiv s + f_1 x + f_2 x^2 \cdots + f_{k-1} x^{k-1} \pmod{q},$$

where $f(0) = s$ is the secret. The dealer distributes the shadow, $x_i \equiv f(i) \pmod{q}$ to node P_i, where i is the ID number for P_i. Finally, in the secret reconstruction phase, k or more nodes can obtain the sharing polynomial f by using Lagrange interpolation, and learn the secret s by the output of $f(0)$.

2.2 Feldman's Verifiable Secret Sharing Scheme

In order to prevent inconsistent shadows fruiom destroying the system, Feldman proposes a scheme based on Shamir's secret sharing scheme in which the node can verify the shadow it has received. Feldman's scheme chooses two prime numbers p and q such that $q = mp + 1$, where m is an integer. It lets g be an element of Z_q of order p. For a piece of information, x, it lets $y \equiv g^x \pmod{q}$ become public, but it remains difficult for outsiders to compute x from y directly, because of it's logarithmic complexity: The dealer allows the values g^s, g^{f_1}, …, $g^{f_{k-1}}$, to be made public, (where s is the secret, and f_1, …, f_{k-1} are the coefficients for sharing polynomial). When the node P_i receives the shadow from the dealer, it can verify the shadow by checking the equation:

$$g^{x_i} \stackrel{?}{\equiv} \left(g^s\right)\left(g^{f_1}\right)^i \left(g^{f_2}\right)^{i^2} \cdots \left(g^{f_{k-1}}\right)^{i^{k-1}} \pmod{q}. \quad (1)$$

The sign, "?", above the equal sign means that if the equal hold. If the equation holds, the shadow it has received is correct.

2.3 Herzberg et al.'s Proactive Secret Sharing Scheme

For a long-term or otherwise important secret, the original secret sharing schemes may be insufficient. Herzberg et al. propose a proactive concept based on Shamir's secret sharing scheme. In Herzberg et al.'s scheme, all time is divided into shadow saving periods and shadow update phases. A shadow saving period is the time during which nodes hold their shadows, and a shadow update phase is the time during which all nodes engage in some interaction to update their shadows. This scheme needs to work under the constraints of a synchronous network. When the system triggers the update phase, all nodes need to execute the update phase simultaneously. All nodes finish their update phase and go into a new shadow saving period, at which point they discard their old shadows from the previous period. The old shadows from the previous period are invalid for the current period, and if an intruder does not destroy more than (n-k) nodes or discover more than (k-1) shadows from a single period, the system will be safe.

Before going through our scheme, we will briefly describe the protocols in Herzberg et al.'s scheme for a (k, n) threshold scheme. In the following, we use superscript (t) to denote the shadows and sharing polynomials computed during period t. It assumes that there is a pair of keys for each pair of nodes, and any two nodes send messages under their pairwise key.

2.3.1 The Initial Protocol

a. Shamir's secret sharing scheme is used to share the secret. For a secret s, the dealer randomly generates a (k-1)th degree sharing polynomial,

$$f^{(t_0)}(x) \equiv s + f_1 x + f_2 x^2 + \cdots + f_{k-1} x^{k-1} \pmod{q},$$

publicizes the coefficients of the sharing polynomial $g^s \pmod q$ and $g^{f_h} \pmod q$, where h = 1, 2, …, k-1, and distributes the shadow $x_i^{(t_0)} \equiv f^{(t_0)}(i) \pmod q$ to node P_i.

b. When the node P_i receives the shadow, it can verify the shadow by checking (1). If the equation holds, the shadow it has received is deemed correct.

c. The dealer then discards the sharing polynomial $f^{(t_0)}(x)$ and secret s.

2.3.2 The Shadow Update Protocol

a. During the update phase, of time t, there exists an honest nodes set, A; their ID numbers form a set, A'. Each honest node $P_i \in A$ randomly generates a (k-1)th degree polynomial

$$\delta_i^{(t)}(x) \equiv \delta_{i,1}^{(t)} x + \delta_{i,2}^{(t)} x^2 + \cdots + \delta_{i,k-1}^{(t)} x^{k-1} \pmod q,$$

where $\delta_i^{(t)}(0) \equiv 0 \pmod q$, and publicizes $g^{\delta_{i,h}^{(t)}} \pmod q$, where h = 1, 2, …, k-1, which $\delta_{i,h}^{(t_0)}$ are the coefficients of polynomial $\delta_i^{(t)}$. P_i computes $u_{ij}^{(t)} \equiv \delta_i^{(t)}(j) \pmod q$ and sends it to node $P_j \in A$, where j is the ID number for node P_j.

b. As each honest node $P_i \in A$ receives the message $u_{ji}^{(t)}$ from each honest node $P_j \in A$, P_i verifies the message using the equation

$$g^{u_{ji}^{(t)}} \stackrel{?}{\equiv} \left(g^{\delta_{j,1}^{(t)}}\right)^i \left(g^{\delta_{j,2}^{(t)}}\right)^{i^2} \cdots \left(g^{\delta_{j,k-1}^{(t)}}\right)^{i^{k-1}} \pmod q.$$

If the equation holds true, P_i updates the shadow by adding the received messages to the previous shadow. The new shadow is therefore

$$x_i^{(t)} \equiv x_i^{(t-1)} + \sum_{j \in A'} u_{ji}^{(t)} \pmod q.$$

c. During every period, the new shadows add a random number, and each node discards their previous shadows. An intruder does not have the chance to discover more previous shadows during current period and furthermore, the previous shadows are invalid in current period.

2.3.3 The Shadow Recovery Protocol

If the node P_r loses the shadow, k or more honest nodes can help P_r with the recovery of its shadow, where k is the threshold.

a. The node $P_i \in A$ randomly generates a (k-1)th de gree polynomial, $\alpha_i^{(t)}(x) \equiv \alpha_{i,0}^{(t)} + \alpha_{i,1}^{(t)} x + \cdots + \alpha_{i,k-1}^{(t)} x^{k-1}$ (mod q), where $\alpha_i^{(t)}(r) \equiv 0 \pmod q$, and publicizes $g^{\alpha_{i,h}^{(t)}} \pmod q$, h = 0, 1,…, k-1. P_i computes $w_{ij}^{(t)} \equiv \alpha_i^{(t)}(j) \pmod q$ and sends $w_{ij}^{(t)}$ to node $P_j \in A$, where j is the id number for P_j.

b. As node $P_i \in A$ receives the message, $w_{ji}^{(t)}$, from each node $P_j \in A$, P_i verifies the message by way of

$$g^{w_{ji}^{(t)}} \stackrel{?}{\equiv} \left(g^{\alpha_{j,0}^{(t)}}\right)\left(g^{\alpha_{j,1}^{(t)}}\right)^i \cdots \left(g^{\alpha_{j,k-1}^{(t)}}\right)^{i^{k-1}} \pmod q \text{ and}$$

$$\left(g^{\alpha_{i,0}^{(t)}}\right)\left(g^{\alpha_{i,1}^{(t)}}\right)^r \cdots \left(g^{\alpha_{i,k-1}^{(t)}}\right)^{r^{k-1}} \stackrel{?}{\equiv} 1 \pmod q.$$

If the equations hold, P_i computes the virtual shadow by adding the received messages to the shadow, where the virtual shadow is $x'^{(t)}_i \equiv x_i^{(t)} + \sum_{j \in A'} w_{ji}^{(t)} \pmod q$, and sends it to P_r.

c. Until node P_r receives k or more virtual shadows, P_r verifies the virtual shadow $x'^{(t)}_i$ from node $P_i \in A$ by

$$g^{x_i'^{(t)}} \stackrel{?}{\equiv} \left(y_i^{(t)}\right)\left(\prod_{j\in A'}\left(g^{\alpha_{j,0}^{(t)}}\right)\left(g^{\alpha_{j,1}^{(t)}}\right)^{i}\cdots\left(g^{\alpha_{j,k-1}^{(t)}}\right)^{i^{k-1}}\right) \pmod q$$

where $y_i^{(t)} \equiv g^{x_i^{(t)}} \pmod q$. If all virtual shadows are correct, P_r uses Lagrange interpolation to compute the virtual polynomial

$$f'^{(t)}(x) \equiv f^{(t)}(x) + \sum_{i\in A'} \alpha_i^{(t)}(x) \pmod q,$$

and the recovered shadow is therefore $x_r^{(t)} \equiv f'^{(t)}(r) \equiv f^{(t)}(r) + \sum_{i\in A'} \alpha_i^{(t)}(r) \equiv f^{(t)}(r) \pmod q$.

2.3.4 Reconstructing the secret

If the nodes want to reconstruct the secret, k or more nodes can reconstruct the polynomial

$$f(x) + \sum_{t\in \text{all time periods}} \sum_{i\in A'^{(t)}} \delta_i^{(t)}(x) \pmod q$$

by using Lagrange interpolation, where $A'^{(t)}$ is the honest nodes' ID set at time period, t. Finally, the secret is yielded by calculating

$$s \equiv f(0) + \sum_{t\in \text{all time periods}} \sum_{i\in A'^{(t)}} \delta_i^{(t)}(0) \equiv f(0) \pmod q.$$

3 Our Scheme

If a node was compromised, it may be insufficient to recover the shadow to it, because the intruder may stay in the node in order to discover its shadow. Replacing it with a new node is better. The recovery protocol is a good method by which to redistribute a new shadow to a new, joining node. At this time we have already developed a mobile membership of nodes. Our scheme can maintain the safety of the (k, n) threshold scheme by properly changing the threshold k according to the amount of nodes. When the nodes become fewer in number, we need to decrease the threshold and let the scheme satisfy with the condition $2k\text{-}1 \le n$.

Our scheme stores the random value, which is generated at each node during each update phase, by distributing some information to all nodes. Until we want to change the threshold, each node collects the distributed information, and computes accumulated random values. Before exchanging information, each node subtracts this value. Finally, the new shadows are generated from the new degree polynomial. Our scheme changes the threshold by changing the degree of the sharing polynomial directly, and the needed memories and computation amount adapt to the threshold.

In the following, we use superscript (t) to denote the shadows and sharing polynomials computed during period t. We assume that there is a pair of keys for each pair of nodes, and that any two nodes will send messages under their associated pairwise key. All time is divided in to shadow saving periods and shadow update phases, therefore the scheme needs to work under the confines of a synchronous network. Our scheme must maintain that k, the threshold number of nodes are safe and do not leave. When there are only k safe nodes, the system triggers the shadow update protocol or the threshold change protocol. Our proposal for such a protocol for a (k, n) threshold scheme is as follows.

3.1. The Initial Protocol

a. There are n nodes that form a set A, and they plan to share a secret s. We select k honest nodes from n nodes to form a set B. For a selected node P_i, where its ID number is i, we let $Nu(l) = i$ and $Nu(1) \le Nu(2) \le \cdots \le Nu(k)$, and those $Nu(l)$ form an ordered set B'.

b. Each node $P_i \in B$ randomly generates a (k-1)th degree polynomial $\delta_l^{(t_0)}$, where $\delta_l^{(t_0)}(0) \equiv 0 \pmod q$ and P_i's ID number is $i = Nu(l)$, and simultaneously sends it to the dealer. P_i publicizes $g^{\delta_{l,h}^{(t_0)}} \pmod q$, $h = 1, 2, \ldots, k\text{-}1$. Finally P_i selects n random numbers $m_{la} \in Z_q$, where $a = 1, 2, \ldots, n$, computes $v_{la}^{(t_0)} \equiv \delta_l^{(t_0)}(m_{la}) \equiv \gamma_l^{(t_0)}(m_{la}) \pmod q$, and sends a triple number $\left(l, m_{la}, v_{la}^{(t_0)}\right)$ to each node $P_a \in A$.

c. When the dealer receives all polynomials from each node in B, the sharing polynomial is computed as:

$$f^{(t_0)} \equiv s + \sum_{g(l)\in B'} \delta_l^{(t_0)} \pmod q .$$

Then the dealer distributes shadows to each node $P_a \in A$, and publicizes $g^s \pmod q$.

d. When node $P_a \in A$ receives the shadow and triple message, it can verify the shadow by equation

$$g^{x_a^{(t_0)}} \stackrel{?}{\equiv} \left(g^s\right) \prod_{g(l)\in B'} \left(g^{\delta_{l,1}^{(t_0)}}\right)^a \cdots \left(g^{\delta_{l,k-1}^{(t_0)}}\right)^{a^{k-1}} \pmod q ,$$

and the value $v_{la}^{(t_0)}$ which is in triple message by

$$g^{v_{la}^{(t_0)}} \stackrel{?}{\equiv} \left(g^{\delta_{l,1}^{(t_0)}}\right)^{m_{la}} \left(g^{\delta_{l,2}^{(t_0)}}\right)^{m_{la}^2} \cdots \left(g^{\delta_{l,k-1}^{(t_0)}}\right)^{m_{la}^{k-1}} \pmod q . \tag{2}$$

If the equation (2) holds, P_a saves $\left(m_{la}, v_{la}^{(t_0)}\right)$ to $C_{a,l}^{(t_0)}$, and $C_a^{(t_0)}$ is an $1\times k$ vector.

e. Dealer discards the sharing polynomial $f^{(t_0)}$ and secret s, and the nodes discard $\delta_l^{(t_0)}$ and $\gamma_l^{(t_0)}$.

3.2. Shadow Update Protocol

a. During update phase, of time t_1, we select k honest nodes from n nodes to form sets B and B' as former ones. One method of discriminating between safe and wrong nodes is proposed in [5].

b. Each node $P_i \in B$ sends $C_{i,o}^{(t_1-1)}$ to each node $P_j \in B$, where P_j's ID number $j = Nu(o)$. As node P_i, with ID number $i = Nu(l)$, receives $C_{j,l}^{(t_1-1)}$ from each node $P_j \in B$, P_i verifies the messages by (2) and uses Lagrange interpolation to compute $\gamma_l^{(t_1-1)}$. Then node P_i randomly generates a (k-1)th degree polynomial, $\delta_l^{(t_1)}$, where $\delta_l^{(t_1)}(0) \equiv 0 \pmod q$, lets $\gamma_l^{(t_1)} \equiv \gamma_l^{(t_1-1)} + \delta_l^{(t_1)} \pmod q$, and publicizes $g^{\delta_{l,h}^{(t_1)}} \pmod q$ and $g^{\gamma_{l,h}^{(t_1)}} \pmod q$, where h = 1, 2,…, k-1. Node P_i selects n random numbers $m_{la} \in Z_q$, a = 1, 2,…, n, and computes $u_{la}^{(t_1)} = \delta_l^{(t_1)}(a) \pmod q$ and $v_{la}^{(t_1)} \equiv \gamma_l^{(t_1)}(m_{la}) \pmod q$. Finally P_i sends a fourfold number $\left(u_{la}^{(t_1)}, l, m_{la}, v_{la}^{(t_1)}\right)$ to node $P_a \in A$.

c. When node $P_a \in A$ receives the message from each node $P_i \in B$, where P_i's ID number is $i = Nu(l)$, P_a verifies the message $u_{la}^{(t_1)}$ by

$$g^{u_{la}^{(t_1)}} \stackrel{?}{\equiv} \left(g^{\delta_{l,1}^{(t_1)}}\right)^a \left(g^{\delta_{l,2}^{(t_1)}}\right)^{a^2} \cdots \left(g^{\delta_{l,k-1}^{(t_1)}}\right)^{a^{k-1}} \pmod q \tag{3}$$

and verifies the message $v_{la}^{(t_1)}$ by way of (2). If the equations hold, P_a updates the shadow as

$$x_a^{(t_1)} \equiv x_a^{(t_1-1)} + \sum_{g(l)\in B'} u_{la}^{(t_1)} \pmod q ,$$

discards the previous shadow, and updates C_a.

d. Node $P_i \in B$ discards $\delta_l^{(t_1)}$ and $\gamma_l^{(t_1)}$.

3.3. The Threshold Change Protocol

As we consider changing the threshold from k to m, the two cases of $m < k$ (see Fig. 1) and $m > k$ will be discussed (see Fig. 2).

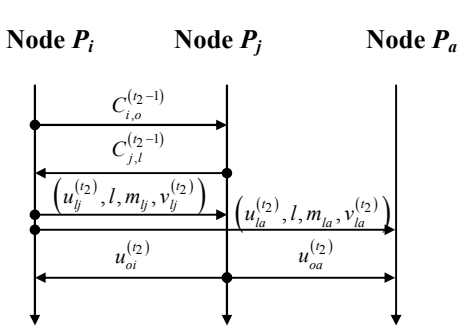

Fig. 1. The threshold change protocol (case 1): P_i and P_j are two honest nodes selected from n nodes to update the shadow, where P_i's ID number $i = Nu(l)$ and P_j's ID number $j = Nu(o)$. P_i is in set B_1 and P_j is in set B_2. P_a is the node which is not selected in set A.

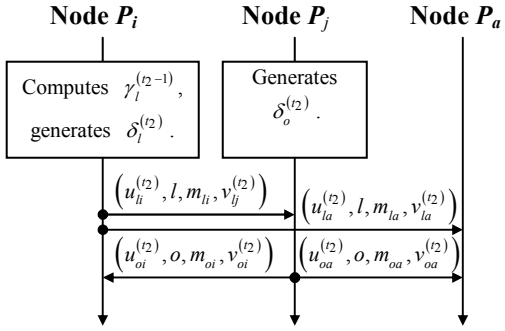

Fig. 2. The threshold change protocol (case 2): P_i and P_j are two honest nodes selected from n nodes to update the shadow, where P_i's ID number $i = Nu(l)$ and P_j's ID number $j = Nu(o)$. P_i is in set B_1 and P_j is in set B_2. P_a is the node which is not selected in set A.

3.3.1. Case 1: When $m < k$

a. During threshold change protocol, of time t2, we select k honest nodes from n nodes to form sets B and B' as former ones. In addition we let $Nu(1) \le \cdots \le Nu(m)$ form a set B'1 and let the corresponding nodes form a set B1. Finally, we let $Nu(m+1) \le \cdots \le Nu(k)$ form a set B'2 and we also let the corresponding nodes form a set B2.

b. Each node $Pi \in$ B, whose ID number is i = Nu(l), sends $C_i^{(t_2-1)}$ to each other and computes $\gamma_l^{(t_2-1)}$.

c. Node $Pi \in$ B1 randomly generates a (m-1)th degree polynomial $\delta_l^{(t_2)}$, where $\delta_l^{(t_2)}(0) \equiv 0 \pmod q$ and Pi's ID number is i = Nu(l), lets $\gamma_l^{(t_2)} = \delta_l^{(t_2)}$, and publicizes some information. Node Pi selects n random numbers as former, computes $u_{la}^{(t_2)} \equiv \delta_l^{(t_2)}(a) - \gamma_l^{(t_2-1)}(a) \pmod q$ and $v_{la}^{(t_2)} \equiv \gamma_{la}^{(t_2)}(m_{la})$, and sends a fourfold number to node $Pa \in$ A.

d. Node $Pj \in$ B2, whose its ID number is j = Nu(o), sends $u_{oa}^{(t_2)} \equiv -\gamma_o^{(t_2-1)}(a) \pmod q$ to node $Pa \in$ A.

e. As node $Pa \in$ A receives the message from each node $Pi \in$ B1 and each node $Pj \in$ B2, where Pi's ID number is i = Nu(l) and Pj's ID number is j = Nu(o), Pa verifies the message $u_{la}^{(t_2)}$ by equation

$$g^{u_{la}^{(t_2)}} \stackrel{?}{\equiv} \left\{\left(g^{\delta_{l,1}^{(t_2)}}\right)^a \cdots \left(g^{\delta_{l,m-1}^{(t_2)}}\right)^{a^{m-1}}\right\} \Big/ \left\{\left(g^{\gamma_{l,1}^{(t_2-1)}}\right)^a \cdots \left(g^{\gamma_{l,k-1}^{(t_2-1)}}\right)^{a^{k-1}}\right\} \pmod q \tag{4}$$

and verify the message $u_{oa}^{(t_2)}$ by way of equation

$$g^{u_{oa}^{(t_2)}} \stackrel{?}{\equiv} 1 \Big/ \left\{\left(g^{\gamma_{o,1}^{(t_2-1)}}\right)^a \cdots \left(g^{\gamma_{o,k-1}^{(t_2-1)}}\right)^{a^{k-1}}\right\} \pmod q .$$

Finally P_a updates the shadow and discards the previous ones. The new shadow is therefore,

$$x_a^{(t_2)} \equiv x_a^{(t_2-1)} + \sum_{g(l) \in B'_1} u_{la}^{(t_2)} + \sum_{g(o) \in B'_2} u_{oa}^{(t_2)} \pmod q .$$

Moreover, P_a verify the information of $\gamma^{(t_2)}$ by (2) and updates table C_a.

f. The node $P_i \in B_1$ discards $\delta_l^{(t_2)}$ and $\gamma_l^{(t_2)}$.

3.3.2. Case 2: When $m > k$

Case 2 is similar to Case 1, and the different parts are shown in following: In part a, we select m honest nodes from n nodes to form a set B and B', and let $Nu(1) \le \cdots \le Nu(k)$ to form sets B'_1 and B_1; let $Nu(k+1) \le \cdots \le Nu(m)$ to form sets B'_2 and B_2. In part d, node P_j randomly generates a (m-1)th degree polynomial $\delta_o^{(t_2)}$, and lets

$\gamma_o^{(t_2)} = \delta_o^{(t_2)}$. Node P_j sends $u_{oa}^{(t_2)} \equiv \delta_o^{(t_2)}(a) \pmod q$ and the information of $\gamma_o^{(t_2)}$ to $P_a \in A$. In step e, node P_a verifies the message $u_{la}^{(t_2)}$ by using (4) and verifies the message $u_{oa}^{(t_2)}$ by (3). Other differences are shown in Fig. 2.

3.4. Reconstruct the Secret

When the nodes want to reconstruct the secret, m or more nodes can reconstruct the sharing polynomial

$$f + \sum_{t \in \text{new threshold}} \sum_{g(l) \in B'^{(t)}} \delta_l^{(t)} \pmod q,$$

where $B'^{(t)}$ is the selected node's ID set during time period, t. The secret is therefore,

$$s \equiv f(0) + \sum_{t \in \text{new threshold}} \sum_{g(l) \in B'^{(t)}} \delta_l^{(t)}(0) \equiv s + 0 \pmod q.$$

4 Discussion

At this point we now discuss the security of our scheme. During every translating period, the previous shadow is changed by adding a random number which is generated from k selected nodes, the same as table C, and therefore, an intruder would not be able to figure out the shadows of the current time from previous shadows and vice versa. The nodes discard the previous shadows after they finish updating protocol. Besides, an intruder can not figure out the polynomials γ, if he does not compromise more then k-1 table C. Under the aforementioned constraints, if the intruder does not receive more than k-1 shadows during one time period, the system is therefore secure.

Now, we consider the node number in each time period. In each time period, there are some new nodes joining the group, some that are corrupted and some that leave the group. Our scheme must maintain k, the threshold number, the number of nodes not changed in each period. When there are only k safe nodes, we trigger the shadow update protocol or the threshold change protocol.

Our scheme has some advantages. Firstly, it is an adaptive scheme in that we can change the threshold according to the amount of nodes. Secondly, we do not need a trusted third party to complete the shadow update protocol and the threshold change protocol. Those protocols are finished by k nodes collectively. Finally, our scheme is more resilient as compared to Schultz's scheme [6]. Our scheme changes the threshold by directly changing the degree of the sharing polynomial, and the needed memories and computation amount adapt to the threshold. However, our scheme needs some rules to select k out of n honest nodes and extra memories in order to achieve a dynamic threshold.

Schultz's method has some limits: The node number can not be less than the two times the degree of the sharing polynomial. After increasing the degree of the sharing to achieve a higher threshold, more "virtual nodes" were used to decrease the threshold.

As Table 1, we therefore compare our current scheme with the original Herzberg et al.'s proactive secret sharing scheme [5] and Schultz's scheme [6] to see how it measures up. Moreover as the threshold increases from k to m (m>k) and then decreases from m to k, the number of nodes to hold the shadow and the number of the interactive message sent during one period are also shown in Table 1.

Table 1. Comparisons in complexity

	The number of the interactive message sent during one period.		The essential number of nodes to hold shadows after changing the threshold.
	Before changing the threshold.	After changing the threshold.	
Our scheme	k^2+kn	k^2+kn	$2k$
Herzberg et al.'s scheme	n^2	--	--
Schultz et al. scheme	$2k^2+(n-k)k$	$2k^2+(n-k)k+(m-k)m$	$m+k$ (include $m-k$ virtual nodes)

5 Applications

Recently, the distributed systems are widely used, such as ad-hoc networks and sensor networks. How to achieve the secure message exchanges in a realistic case is an interesting issue among the network security applications. Conventionally, Public Key Infrastructure (PKI) is used in the distributed systems, where a Certificate Authority (CA) which holds many public and private keys provides certificates enables a trusted message exchange in network systems. Nevertheless, the single CA is not capable of being considered in the ad-hoc systems due to the feature of non-centralized node propagations. Therefore, Dong et al. [7] and Kong et al. [8] proposed a distributed certificate authority service scheme, instead of the single CA service in ad-hoc systems. The schemes distribute the authority of CA to n nodes based on (k, n) secret sharing scheme. In other words, the role of CA is made of a number of nodes to work in the systems whenever the threshold k does encompass. The implemented work is briefly reviewed as follows right prior to the application as our proposal.

Kong et al.'s scheme

In Kong et al.'s scheme [8], there is a key-pair of public key PK and private key SK, where the certification is signed by SK and verified by PK in use. Firstly, a dealer distributes SK to n nodes by using Shamir's scheme [1]. The shared information, x_i, called shadow, is held by a node P_i. When one node wants to request a certification, it sends requests to all neighbor nodes of k-1 asking the partial certifications back, where the partial certifications are signed by shadows associated with the nodes. The genuine certification is therefore generated whenever more than k-1 partial information is collected by the request node. The details are shown as follows:

a. The dealer randomly generates a (k-1)th degree polynomial, in Z_q

$$f(x) \equiv SK + f_1 x + f_2 x^2 \cdots + f_{k-1} x^{k-1} \pmod{q},$$

where $f(0) = SK$ is assigned as a private key. The dealer distributes the shadow, $x_i \equiv f(i) \pmod{q}$ to node P_i, where i is treated as the ID number for P_i.

b. Once the shadows are received by nodes, the private key SK can be reconstructed by using Lagrange interpolation as the rule:

$$SK \equiv \sum_{j=1}^{k} \left(x_j \cdot lc_j(0) \right) \equiv \sum_{j=1}^{k} SK_j \pmod{q}, \tag{5}$$

where $lc_j(0)$'s are Lagrange coefficients, and are defined as

$$lc_j(x) \equiv \frac{(x-1)\cdots(x-(j-1))(x-(j+1))\cdots(x-k)}{(j-1)\cdots(j-(j-1))(j-(j+1))\cdots(j-k)} (\mathrm{mod}\, q). \tag{6}$$

c. When a node P_j wants to request a certification, it broadcasts a random number, X, to all neighbor nodes. The neighbor node P_i therefore computes the partial certification X^{SK_i}, where SK_i is derived from shadow x_i by the computations of (5) and (6).

d. The node P_j combines the k received partial certifications to generate the genuine certification, X^{SK} as follows:

$$X^{SK} \equiv X^{SK_1} \cdot X^{SK_2} \cdot \cdots \cdot X^{SK_k} (\mathrm{mod}\, q). \tag{7}$$

According to RSA implementations, the certification X^{SK} can be verified by the public key PK.

Implementation

In the distributed certificate authority service scheme, if the intrusion behavior is compassed, the attacker needs to compromise more than k-1 nodes so as to obtain the private key. It is possible to make it done as long as the systems keep running longer without any changing at shadows as well as threshold policy upon secret sharing applications. As the observations we studied, the proposed proactive secret sharing scheme is imposed to solve the compromise concerns in [8] and make the connection to the realistic managements in the network systems.

Consider a scenario that a company is composed of a managing director and some assistant managers. We model a company policy decision-making upon the secret sharing structure as follows: Initially the managing director chooses a key-pair, public key $PK = \{e, n\}$ and private key $SK = \{d\}$. There are n assistant managers, P_1, P_2,…, and P_n in the system. The private key is shared to n different assistant managers by using our secret sharing scheme presented in Section 3, where the shadow, x_i and the table, C_i, both are held by an assistant manager, P_i. We further assume that the staff members, E_j's, in this company, who dedicate to evaluate the requirements of products or service qualities to the customers in the business market investigations. Assuming

one of staff members, E_j, discovers something benefits with this company market exploitation. Firstly he/she needs to discuss it with the assistant manager, P_i. Once the evaluations are positive, P_i will assign a commitment, X^{SK_i}, to a report of project, where X^{SK_i} is computed from x_i and (6), X is a random code generated by E_j. The proposed project will finally submit to managing director via the delivery of E_j. The decision will be made as long as more than *k-1* commitments are received based on (k, n) secret sharing scheme, where the threshold is set as k among n assistant manger. In other words, the staff member E_j will collect k commitments, X^{SK_1}, X^{SK_2} …, and X^{SK_k}, from k assistant managers, P_i's, and then conduct a certification X^d to adhere to the proposed project, where X^d is computed by (7). Next, the proposed project is sent off and verified by the managing director using the public key set of $PK = \{e, n\}$. All the message exchanges and computations are under the protocols proposed in Section 3. The implementation with our scheme application is shown in Fig. 3.

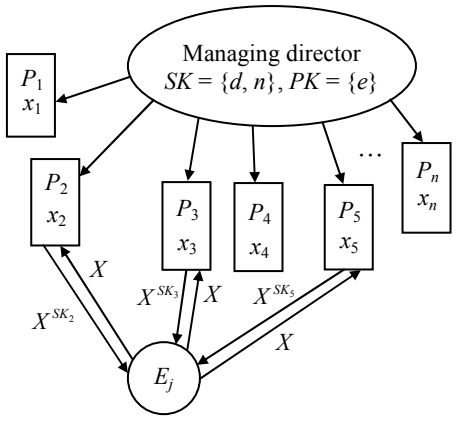

Fig. 3. A project generation organization upon the secret sharing application using our protocols in Sec. 3

6 Conclusions

In this paper, we have proposed a new (k, n) threshold secret sharing scheme which allows for the changing of participant number n and arbitrarily changing the threshold k. As analyzed to our scheme, the node number comparisons are less than the past studies in terms of interactive message exchanges and shadow counting. Furthermore, we propose a sort of application on the basis of the scene of commerce transaction in a secure message exchange manner, where the trusted message deliveries, flexible shadow distributions and exchanges enable the decision of business market investigations implemented successfully.

References

[1] A. Shamir, "How to share a secret," *Communications of the ACM*, Vol. 22, No.11, pp. 612-613, 1979.

[2] G. Blakley, "Safeguarding cryptographic keys," *Proceedings of the AFIPS 1979 National Computer Conference*, Vol. 48, Arlington, VA, pp. 313-317, June 1997.

[3] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pp. 427-437, 1987.

[4] T. P. Pederson, "Non-interactive and information- theoretic secure verifiable secret sharing," *Advances in Cryptology*, pp. 129-140, 1991.

[5] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing, or how to cope with perpetual leakage," *Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, pp. 339-352, August 1995.

[6] D. Schultz, B. Liskov, and M. Liskov, "MPSS: mobile proactive secret sharing," http://www.cs.wm.edu/~mliskov/full-paper.pdf, Nov. 2006.

[7] Y. Dong, H. W. Go, A. F. Sui, V. O. K. Li, L. C. .K. Hui, and S. M. Yiu, "Providing distributed certificate authority service in mobile ad hoc networks," *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pp. 149-156, 2005.

[8] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," *Proceedings of the International Conference on Network Protocols (ICNF)*, pp. 251-260, 2001.