DOI: 10.2298/CSIS120801048T

Distributed Secret Sharing Scheme Based on Personalized Spherical Coordinates Space

Zhenhua Tan[1], Guangming Yang[1], Wei Cheng[1], and Xingwei Wang[2]

[1] Software College of Northeastern University, POB 349#, 110819 Shenyang, China
{tanzh, yanggm, chengw}@mail.neu.edu.cn
[2] Faculty of Information Science and Engineering of Northeastern University 11 0819 Shenyang, China
wangxw@mail.neu.edu.cn

Abstract. Threshold secret sharing schemes are ideal to protect confidential information. In this paper, we propose a novel distributed <k, n> threshold secret sharing scheme based on spherical coordinates. As four non-coplanar points can determine a unique sphere, we design transformation algorithms to generate secret as sphere center and mapping algorithms to convert participants to be sphere surface points. An algorithm to generate shadow secrets is proposed based on spherical coordinates. Verifiability and proactivity secret sharing are considered during the procedures of generating shadows and recovering secret and four or more participants could recover secret in our scheme. Performance analysis proves that the proposed scheme has relatively advantage in computation complexity, storage space and communication amounts during distribution and reconstruction processes, and it can tolerate collusion attacks and detect dishonest participants.

Keywords: secret sharing, distributed security, secret protection.

1. Introduction

Security of crucial information, such as credit evidences and confidential documents, has become one of the most important issues nowadays. Usually, we encrypt these pieces of critical information to protect data. However, how to protect the encryption key? Shamir mentioned the question in 1979 [1]. In practical, to improve encryptions can not solve this problem. Traditional methods for encryptions are ill-suited for simultaneously achieving high levels of confidentiality and reliability as they usually keep keys in a single and well-guarded location where single-point-failure exists. Moreover, it is also very significant that keys not be lost or exposed.

A (k, n) threshold secret sharing scheme is a method to store secrets in distributed form, usually used in a distributed network to protect highly sensitive and important information.

However, a secret sharing scheme should have ability to avoid dishonest behaviors. A dishonest participant may provide incorrect shadow so that dealer couldn't reconstruct the original correct secret. Moreover, attacker may also steal enough shadows from participants so as to get the secret. Thus, researchers studied hotly on verifiable secret scheme (VSS) and proactivity secret scheme (PSS) [6-21], while earlier schemes such as [1] and [2] have no such performance. Some VSS schemes use signature technology to ensure the verifiability and some PSS schemes use dynamic shares to ensure proactivity. Nevertheless, they depended on time cost or a trusted third party. In this paper, we think a distributed secret sharing scheme needs a fully distributed way, and performance of verifiability and proactivity should rely on distribution algorithm and reconstruction algorithm.

In this paper, we focus on the basic sharing algorithms in order to find new secret sharing scheme with verifiability and proactivity performance in fully distributed way. The main purpose of this paper is to protect secret key, and algorithms could be applied to big confidential information if some corresponding changes are made.

Thus, based on personalized sphere space, we propose a novel distributed secret sharing scheme, and secret and shadows are denoted as 3-d spherical coordinates. It is a geometry-based threshold scheme with the following contributions. To our knowledge, it is the first and quite new secret sharing scheme based on spherical coordinates.

(1) Algorithms are proposed to convert secret as a sphere center. During the converting processes, we use coordinates transformations and splitting operations to protect original secret against collusion attacks and dishonest participants.

(2) An algorithm is proposed to convert participants to be points on a unique and personalized sphere surface. On the basis of this, we design methods to generate shadows and distribute them to n participants.

(3) We prove a theorem that four non-coplanar points can recover a unique sphere center, and conclude its computational equations. At last, a <k, n> threshold secret scheme is proposed with $k \geqslant 4$.

The remainder of the paper is organized as follows. The next section briefly presents the related work. Section 3 introduces the secret distribution procedures of proposed scheme while the reconstruction processes are introduced in section 4. We analyze capabilities of proposed scheme in section 5 and the conclusions are stated in Section 6.

2. Related Work

Secret sharing was invented independently by Adi Shamir [1] and George Blakley [2] in 1979, based on Lagrange interpolating and liner project geometry respectively.

A secret sharing scheme consists of one dealer, n participants (or players), an original secret, a secret distribution algorithm and a secret reconstruction

algorithm. The dealer shares a secret among a given set of n participants, such that every k of those participants ($k \leqslant n$) could reconstruct the secret by their shares together, while any group of fewer than k participants gets no information about secret [1-3].

Take Shamir's scheme for instance, in this scheme, any t out of n shares may be used to recover the secret. The system relies on the idea that 2 points are sufficient to define a line, 3 points are sufficient to define a parabola, 4 points to define a cubic curve and so forth. That is, it takes k points to define a polynomial of degree (k-1). The method is to create a polynomial of degree k-1 with the secret as the first coefficient and the remaining coefficients are picked at random. Next find n points on the curve and give one to each of the players. When at least t out of the n players reveals their points, there is sufficient information to fit a (k-1) degree polynomial to them, the first coefficient being the secret.

At the very beginning, the implicit assumption in the original formulation of secret sharing scheme is that each participant is either honest or corrupt, and honest participants are all willing to cooperate when dealer requests reconstruction of the secret. Such schemes include Shamir's threshold secret sharing scheme [1] based on polynomial interpolation, Blakley's geometric threshold secret sharing scheme [2], Mignotte's scheme [4] and Asmuth-Bloom's scheme [5] based on the Chinese remainder theorem in 1983.

Gradually, researchers began to consider the security problems of secret sharing itself, such as how to verify dishonest players. HARN proposed a threshold secret sharing scheme [6] based on digital signature with certification functions in 1995. HSU and WU's scheme [7] improved HARN's scheme and proposed an encryption secret sharing scheme based on discrete logarithms with more efficiency for signature verification. Scheme of Han et al [8] used a certification center to verify dishonest shares. Marsh and Schneider designed a secret sharing system [9] which was fault-tolerant and attack-tolerant. All of these made the research on secret sharing scheme more and more mature. Works [10][11][12] were also secret sharing schemes with verification mechanisms.

Starting with the work of Halpern and Teague [13], participants in secret sharing are neither honest nor corrupt but are instead viewed as rational and are assumed to act in their own self-interest. Rational secret sharing is a problem at the intersection of cryptography and game theory. In essence, a dealer wishes to engineer a communication game that, when rationally played, guarantees that each of the players learns the dealer's secret. Yet, all solutions proposed so far did not rely solely on the players' rationality, but also on their beliefs, and were also quite inefficient. Micali and Shelat exhibited a very efficient and purely rational solution to it with a verifiable trusted channel in their scheme [14]. Fuchsbauer et al also proposed a new methodology for rational secret sharing leading to various instantiations in both the two-party and multi-party settings [15].

Of course, because of its generality, secret sharing schemes step into many fields, such as image secret sharing [16][17]. More and more information

security applications consider secret sharing scheme to protect confidential key data, and verifiable and proactive sharing mechanisms are hotly studied [18-21].

3. Secret Sharing Procedures of Proposed Scheme

In a certain distributed network, such as an ad hoc network or P2P (peer-to-peer) network, any node should have a global and unique identifier. We call such identifiers ID_0, ID_1, ID_2, etc. Particularly, we assume that the dealer's identifier is ID_0, and let

$$IDSet = \{ID_1, ID_2, ID_3, ..., ID_n\}. \quad (1)$$

IDSet is a set of participants' identifiers. In this paper, the running environment is a P2P file sharing network, and unique *ID* is an integer provided by a one-way trapdoor function. In our proposed scheme, we select *SHA-2* with 512 bits (so-called *SHA-512*) to generate *ID* numbers, which is the current standard secure hash algorithm claimed by NIST (the National Institute of Standards and Technology) [22]. We use "*SHA"* to denote our selected algorithm *SHA-2* here. Maybe we will try *SHA-3* hash algorithm [23] in future.

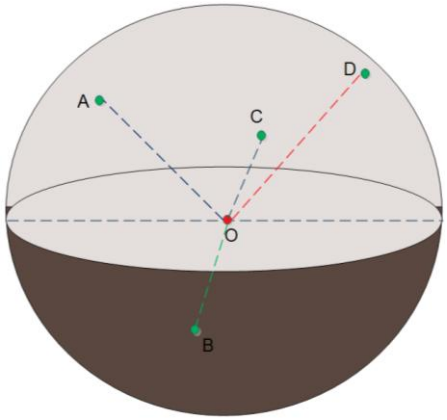

Fig. 1. Sphere with secret center.

The proposed scheme will make full use of the *IDSet* in both distribution and reconstruction procedures. Let S_0 be an original secret, which is an arbitrary length binary number. All information should be converted to a 3-d spherical coordinate as <*x, y, z*> firstly, and secret S_0 will be converted to a sphere center while participants in *IDSet* will be transformed into points on sphere surface, just as shown in figure 1.

Next, we will introduce how the dealer shares S_0 with n participants using n different shadows based on spherical coordinates, including algorithms to get coordinates and procedures to generate shadow secrets.

3.1. Convert Secret S_0 to Be Sphere Center

Any information in computer could be converted to a series of binary number, whether it is a big image or only a short secret key. Thus, we assume the original secret S_0 is an arbitrary length binary number. There are two steps to convert S_0 to be a sphere center $<x_0, y_0, z_0>$. The first step is to convert secret and node identifiers to initialized 3-d coordinates which are appropriate for coordinate computing, and then to run some transformation operation with these coordinates to get a sphere center.

Firstly, we convert the original secret into an initialized coordinate <x, y, z> by dividing binary number into two blocks and a verifying hash code. Figure 2 shows the method.

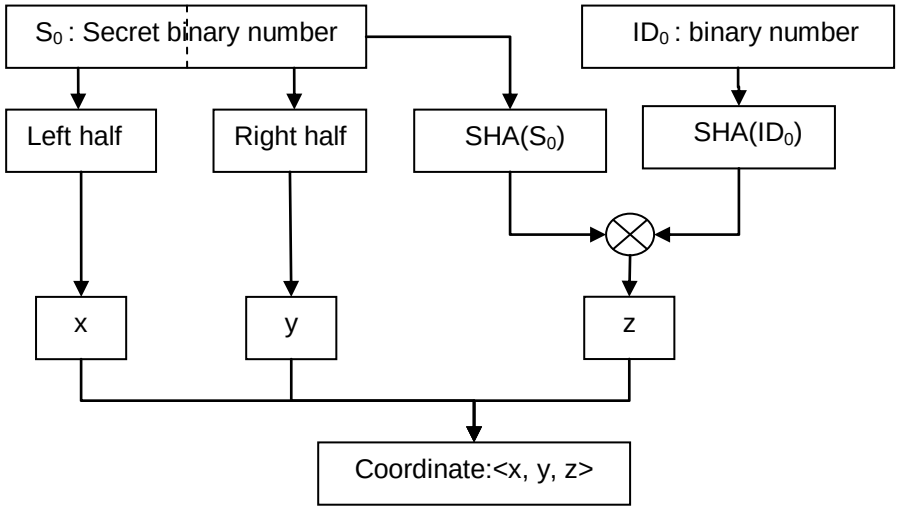

Fig. 2. Processes to initialize S_0 to be a coordinate $<x, y, z>$. The value of z is resulted from $SHA(S_0) \otimes SHA(ID_0)$, while the real secret is only divided into two parts x and y. However, z is very important for the coordinate as it is used for verifying the rightness of $<x, y>$ during secret recovering processes. We will discuss more in section 4 about this mechanism.

In principle, any length of the input number can be accepted by the proposed algorithm. However, with consideration of performance, we suggest a limited length is required, such as 256 bits and 512bits. Surely, our algorithm could also be applied to big length input number by dividing the big input into blocks with given length. We do not mention such a block-algorithm in this

paper while focusing on the following algorithms which will convert a secret series into an initialized coordinate data.

```
Algorithm 1:program InitializedSecret(S0, ID0)
  (1)Padding zeros to S0 if its length less than limited
length
  (2)x←(Stemp ←S0) SHR (length(S0)/2)
  (3)y←(Stemp ←S0) AND (2^(length(S0)/2)-1)
  (4)z←SHA(S0)⊗SHA(ID0) //⊗ here is bitwise exclusive or
  (5)return <x, y, z>
end.
```

In the above algorithm, step (2) is used to get half length high bits of the original secret S_0, and step (3) to get half length of low bits. At last, a binary secret is transformed into a coordinate <*x, y, z*>. By the way, in our proposed scheme, we do not consider more detailed computing process, such as large number problems. We only consider the logic, and anything else could be solved during the actual engineering. In fact, S_0 and ID_0 are large numbers in real networks, but we could use some auxiliary methods to improve computing efficiency in real engineering.

Meanwhile, the node *ID* is also an integer number with a given length (is 512 bits, 1024 bits or else). For example, ID_0 = *01110101010110110001001010010010101010111001001 0*. So, we design a simple algorithm to convert it into coordinate data by dividing the *ID* to trisection. The following pseudo code shows the procedures.

```
Algorithm 2: program GetCoordinateForID (ID)
  (1)x←(IDtemp ←ID) SHR (2*length(ID)/3) // to get the first
1/3 high bits
  (2)IDtemp←((IDtemp ←ID) AND (2^(2*length(ID)/3)-1))
  (3)y←IDtemp SHR (length(ID)/3)//to get the second 1/3 high
bits
  (4)z←(ID AND (2^(length(ID)/3)-1))
  (5)return <x, y, z>
end.
```

Step (1) is used to get the first 1/3 high bits of an *identifier* by shift right operation while step (2) and (3) is used to get the second 1/3 high bits, and step (4) to get the last parts. For example, the above ID_0= *01110101010110110001001010010010101010111001001 0* could be generated into a coordinate with *x=0111010101011011*, *y=0001001010010010 and z=1010101110010010* by algorithm 2.

Then, based on the above two algorithms, we could generate the sphere center by S_0 and ID_0 according to algorithm 3 named *GetSphereCenter*. Let the sphere center point be <x_0, y_0, z_0>.

```
Algorithm 3: program GetSphereCenter (S0 , ID0)
  (1)Stemp←(S0 ⊗ ID0)
  (2)<xtemp, ytemp, ztemp>←InitializedSecret(Stemp, ID0)
```

(3) $ID_{temp} \leftarrow SHA(ID_0)$
(4) $<x_{id0}, y_{id0}, z_{id0}> \leftarrow GetCoordinateForID(ID_{temp})$
(5) $P_0' \leftarrow <x_0', y_0', z_0'> \leftarrow <x_{temp} \otimes x_{id0}, y_{temp} \otimes y_{id0}, z_{temp} \otimes z_{id0}>$
(6) $Pri_{x0} \leftarrow x_0'$ AND $(2^{16}-1)$ //used to get 16 low bits
(7) $Pri_{y0} \leftarrow y_0'$ AND $(2^{16}-1)$
(8) $Pri_{z0} \leftarrow z_0'$ AND $(2^{16}-1)$
(9) $x_0 \leftarrow x_0'$ SHR 16
(10) $y_0 \leftarrow y_0'$ SHR 16
(11) $z_0 \leftarrow z_0'$ SHR 16
(12) Node ID_0 stores $Pri_0 \leftarrow <Pri_{x0}, Pri_{y0}, Pri_{z0}>$ as a private coordinate
(13) return $<x_0, y_0, z_0>$
end.

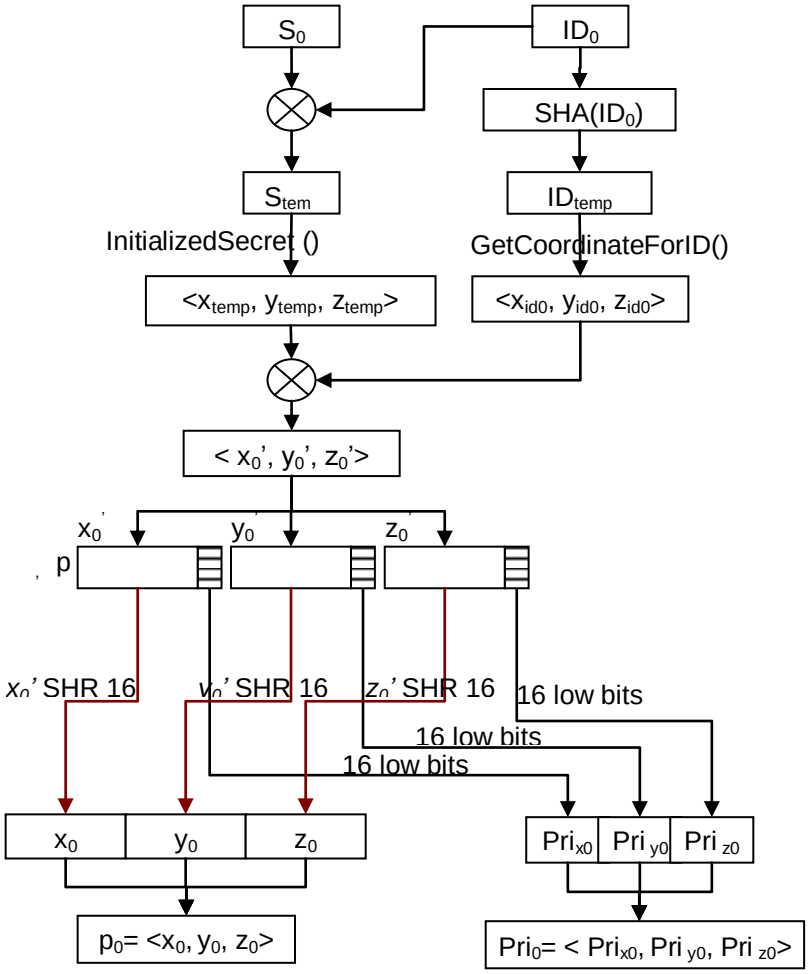

Fig. 3. Processes to generate sphere center point $<x_0, y_0, z_0>$.

In order to describe this algorithm more clearly and directly, we make the following figure 3 to show the algorithm flow.

In algorithm 3, secret S_0 could be hidden and protected by the dealer's ID_0 by two phases' operations of bitwise exclusive or, and the first $\otimes$ operation is made by ID_0 itself while the second $\otimes$ is based on the hash result of ID_0. Based on an intermediate coordinate P_0'=<x_0', y_0', z_0'> in step (5) of the algorithm, we truncate the 16 low bits respectively from three coordinates x_0', y_0', and z_0' via "and 00ffh" operation. These three parts are generated to be a private coordinate data Pri_0= < Pri_{x0}, Pri_{y0}, Pri_{z0}> which is stored in dealer. By the way, this private coordinate could also be distributed to some trusted nodes from dealer's perspective. How to get dealer's trusted participants can refer to our former work [24], and this paper won't expand it instead of only store this private coordinate in dealer itself.

3.2. Convert Participants to Points on Sphere Surface

Based on the above section, we will construct a sphere space with sphere center <x_0, y_0, z_0> and participant *ID*s in this section. Based on the initialized coordinates generated by algorithm 2, all of the participant *ID*s will be converted to 3-d sphere coordinates like <x_i, y_i, z_i> with same center and radius to appear on a same sphere surface.

Firstly, we preprocess the *IDSet* using $\otimes$ operation with ID_0 in order to hide the *ID* information in coordinate data. So that,

$$ID_i = ID_i \otimes SHA(ID_0). \tag{2}$$

We use $SHA(ID_0)$ instead of ID_0 itself. This process is beneficial to protect information and does not easily expose the dealer ID_0. Then, *algorithm 2* is designed to convert the participant *ID*s to initial sphere coordinates. That is,

$$p_i =< x_i, y_i, z_i >= GetCoordinateForID(ID_i). \tag{3}$$

And store them in another set named *CSet* as

$$CSet = \{p_1, p_2, ..., p_n\}. \tag{4}$$

Then, compute the Euclidean distance between points p_0 and p_i in *CSet*, that is,

$$r_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2 + (z_0 - z_i)^2}, p_i =< x_i, y_i, z_i >\in CSet. \tag{5}$$

Thus now we choose the longest distance with an increase to be the radius *R* of dealer's sphere via

$$R = \underset{i=1}{\overset{n}{Max}}(r_i) + \varepsilon. \quad (6)$$

The increase ε here is used to hide information of point with max radius. By the way, we couldn't choose the longest distance or even the average distance to be sphere radius here. Purpose of this sphere radius *R* is to construct a sphere space so that all of the participants could be mapped into a same sphere surface. So mapped points on sphere surface will be shadow points, and their coordinates should be quite different to former coordinates generated by participant identifiers. The later content in this section will introduce how to map participant coordinates into a same sphere surface. Therefore, we couldn't use the longest distance to be the radius directly here, or else the point with such max value would reveal its coordinate with security vulnerability and can not to be shadow point. If we use average distance in equation (6), the same situation may appear also.

Assume that the longest distance appears between points p_0 and p_m. We need to extend p_m to a new virtual point p_m^E with a ε increase so that the distance between p_0 and p_m^E equal to sphere radius *R*. Therefore, all actual points are interior points in the sphere at present and need to be extended onto the sphere surface so that all points will have the same radius. Just as shown in figure 4.

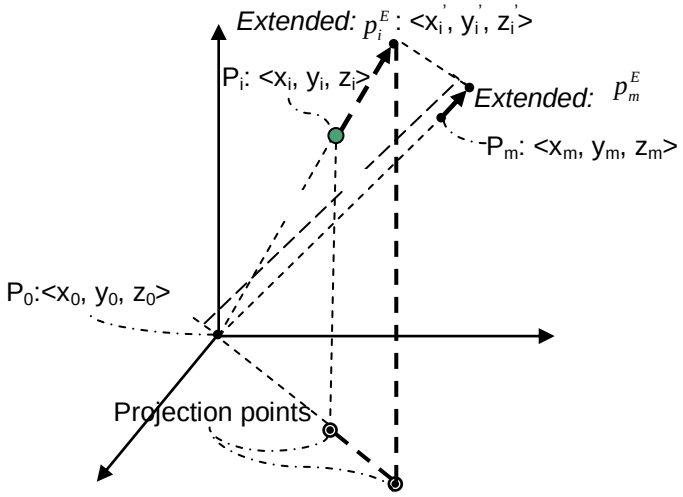

Fig. 4. Spherical coordinates with center P_0=<x_0, y_0, z_0>. We take one of the interior points p_i for instance, once we extend p_i to sphere surface, its coordinate data <x_i, y_i, z_i> can be updated with the same rate $\frac{R}{r_i}$.

Let's take point p_i as an example. As you can see from figure 4, we project points p_i and p_i^E to the plane with x and y axis. So, $x_i = \frac{R}{r_i}(x_i - x_0) + x_0$ as distance between p_0 and p_i^E is *R*. We can get y_i and z_i with the same method. Thus, it is easy to get extended coordinate data for point p_i. That is,

$$p_i^E = < \left(\frac{R}{r_i}(x_i - x_0) + x_0\right), \left(\frac{R}{r_i}(y_i - y_0) + y_0\right), \left(\frac{R}{r_i}(z_i - z_0) + z_0\right) > \quad (7)$$

.

After updating all of the coordinates, a new set $CSet^E = \{p_1^E, p_2^E, ..., p_n^E\}$ will store all of the new coordinate data to replace $CSet$. All of the participants could be mapped into the sphere surface with a same sphere center p_0. Figure 5 is a demonstration of such a mapping process. These n p_i^E in $CSet^E$ are shadow secrets.

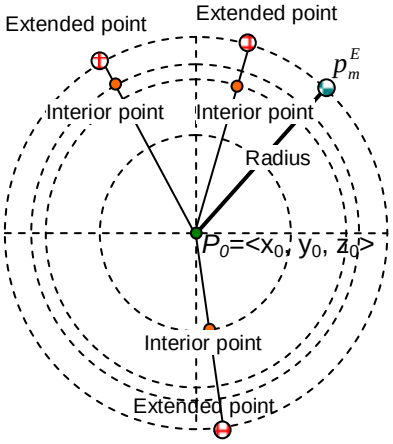

Fig. 5. This is demonstration of constructing participants onto sphere surface. Every interior point is mapped into a same sphere surface after the sphere radius is computed out.

Up to now, all of the participants have been mapped into a same sphere surface. One of the most remarkable things about the sphere space is personalized. In a specific distributed network, every node can share their special secret in our proposed scheme. Different secret will generate different sphere center. Therefore, every node can construct a personalized sphere space themselves. In other words, different node has different sphere space. This fact helps to hide the secret source well. However, a most important prerequisite to construct a sphere space for a node is that there needs at least

four active participants in the networks, or else, the secret is unable to be recovered.

3.3. Secret Distribution Algorithm

Based on the above two sections, algorithms have generated *n* coordinate points. In fact, these coordinates are shadow secrets for secret S_0. In this section, we will conclude the secret distribution procedures in algorithm 4 according to our analysis above.

```
Algorithm 4: program ShareSecret (S0 , ID0)
  (1)P0←<x0, y0, z0>←GetSphereCenter(S0, ID0)
  (2)IDSet←{ID1,ID2,…,IDn}
  (3)For i←1 to n do
  (4)  IDi←IDi⊗SHA(ID0)
  (5)  Pi←<xi, yi, zi>←GetCoordinateForID(IDi)
  (6)  ri←sqrt((P0.x0−Pi.xi)^2 +(P0.y0−Pi.yi)^2 +(P0.z0−Pi.zi)^2)
       //equation (5)
  (7)End for
  (8)R←(Max(ri)+ε)                          //equation (6)
  (9)For i←1 to n do
  (10)  xi←(R*(Pi.xi−P0.x0)/ri + P0.x0)
  (11)  yi←(R*(Pi.yi−P0.y0)/ri + P0.y0)
  (12)  zi←(R*(Pi.zi−P0.z0)/ri + P0.z0)
  (13)  PEi←<xi, yi, zi>                    //equation (7)
  (14)End for
  (15)For i←1 to n do
  (16)  node(ID0).send(PEi) to node(IDSet[i])
  (17)End for
end.
```

As described in algorithm 4, the dealer ID_0 generates *n* coordinate shadows for secret S_0 and distributes them to participants after a series of operations. The dealer could remove its original secret after distributing shadows to participants.

4. Secret Reconstruction Procedures of Proposed Scheme

This section will introduce how to recover the original secret S_0 from participants. In section 3, the secret has been converted to sphere center and its shadows are constructed onto sphere surface respectively corresponding to a participant. So, during the secret reconstruction processes, we need to recover a sphere center by shadows from participants. We prove that at least four shadows would reconstruct a unique sphere by analytic geometry in this

section firstly, and then we could easily get the original secret from the sphere center.

4.1. Four Non-coplanar Points Determine a Unique Sphere

Using analytic geometry principle [25], we firstly prove that there is a unique sphere that passes through four non-coplanar points if, and only if, they are not on the same plane. Based on this theorem, we could easily recover the secret by four or more than four non-coplanar points.

Theorem 1. Four non-coplanar points determine a unique sphere.

*Proof***.** Assume A, B, C and D are four non-coplanar points as shown in figure 6.

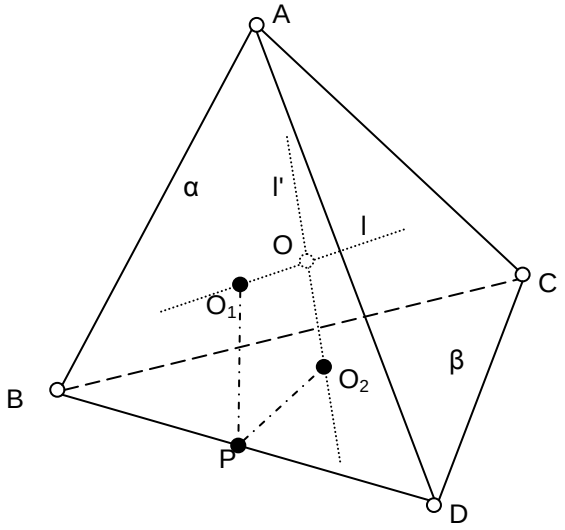

Fig. 6. Points A, B, C, and D are four non-coplanar points. We want to prove that there is a unique sphere center *O* determined by these four points.

Let points A, B, D be in plane α, and O_1 be the center of the circumcircle that passes the three points. So,

Result [1] $d(O_1, A) = d(O_1, B) = d(O_1, D)$.

Let points B, C, D be in plane *β*, and O_2 be the center of their circumcircle. Therefore,

Result [2] $d(O_2, B) = d(O_2, C) = d(O_2, D)$.

Let line *l* be the vertical line of plane α through point O_1; and *l'* be the vertical line of plane β through point O_2; and *P* be the midpoint of segment BD. Then, we link O_1P and O_2P. Assume line *l* and line O_1P are in plane *α'*. According to result [1], we can get that

Result [3] $BD \perp O_1P, BD \perp l \Rightarrow BD \perp \alpha'$.

Let line *l'* and O_2P be in plane *β'*. According to result [2], we can get that

Result [4] $BD \perp O_2P, BD \perp l' \Rightarrow BD \perp \beta'$.

According to results [3] and [4], BD is perpendicular to two planes *α'* and *β'*, and the two planes are intersecting across point *p*. So, the two planes are same.

Result [5] $\alpha' = \beta'$

Thus, line *l* and *l'* are in the same plane and must intersect with each other. Let *O* be their intersection point. According to the above results, we can easily get

Result [6] $d(O,A) = d(O,B) = d(O,C) = d(O,D)$.

Result [6] means the four non-coplanar points A, B, C, D are in a unique sphere with center *O*.

Proof completed.

Once we could get four non-coplanar shadows from participants, we are sure to infer a unique sphere center.

4.2. K≥4 Sphere Coordinates Reconstruct a Unique Sphere Center

Theorem 1 tells us that we need to find 4 or more non-coplanar points to recover sphere center. In this section, we design methods to fetch *k* (⩾4) available coordinates from participants. Thus, we define a theorem to compute a unique sphere center coordinate by four points in the following.

Theorem 2. Assume there are four points <x_1, y_1, z_1>, <x_2, y_2, z_2>, <x_3, y_3, z_3>, and <x_4, y_4, z_4>, and $\det(A) = \begin{bmatrix} (x_2 - x_1) & (y_2 - y_1) & (z_2 - z_1) \\ (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) \\ (x_4 - x_1) & (y_4 - y_1) & (z_4 - z_1) \end{bmatrix}$. If $|A| \neq 0$, then the four points could determine a unique sphere center <x_0, y_0, z_0>, where $x_0 = \frac{\det(A_1)}{\det(A)}$, $y_0 = \frac{\det(A_2)}{\det(A)}$, and $z_0 = \frac{\det(A_3)}{\det(A)}$.

Proof. In analytic geometry, a sphere with center <x_0, y_0, z_0> and radius *R* is the locus of all points (x, y, z) such that

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2 \quad (8)$$

Thus, for the four given points<x_1, y_1, z_1>, <x_2, y_2, z_2>, <x_3, y_3, z_3>, and <x_4, y_4, z_4>, we can get

$$\begin{cases}(x_1-x_0)^2+(y_1-y_0)^2+(z_1-z_0)^2=R^2 & (e1)\\ (x_2-x_0)^2+(y_2-y_0)^2+(z_2-z_0)^2=R^2 & (e2)\\ (x_3-x_0)^2+(y_3-y_0)^2+(z_3-z_0)^2=R^2 & (e3)\\ (x_4-x_0)^2+(y_4-y_0)^2+(z_4-z_0)^2=R^2 & (e4)\end{cases}$$

Let $(e1)-(e2),(e1)-(e3),(e1)-(e4)$, then

$$\begin{cases}x_0(x_2-x_1)+y_0(y_2-y_1)+z_0(z_2-z_1)=0.5\left(\left(x_2^2+y_2^2+z_2^2\right)-\left(x_1^2+y_1^2+z_1^2\right)\right)\\ x_0(x_3-x_1)+y_0(y_3-y_1)+z_0(z_3-z_1)=0.5\left(\left(x_3^2+y_3^2+z_3^2\right)-\left(x_1^2+y_1^2+z_1^2\right)\right)\\ x_0(x_4-x_1)+y_0(y_4-y_1)+z_0(z_4-z_1)=0.5\left(\left(x_4^2+y_4^2+z_4^2\right)-\left(x_1^2+y_1^2+z_1^2\right)\right)\end{cases}$$

This is a standard determinant equation $AX=b$ for unknown numbers x_0, y_0 and z_0. So, we can get

$$\det(A)=\begin{bmatrix}(x_2-x_1) & (y_2-y_1) & (z_2-z_1)\\ (x_3-x_1) & (y_3-y_1) & (z_3-z_1)\\ (x_4-x_1) & (y_4-y_1) & (z_4-z_1)\end{bmatrix}, \tag{9}$$

$$X=\begin{bmatrix}x_0\\ y_0\\ z_0\end{bmatrix}, \tag{10}$$

and,

$$b=\begin{bmatrix}0.5\left((x_2^2+y_2^2+z_2^2)-(x_1^2+y_1^2+z_1^2)\right)\\ 0.5\left((x_3^2+y_3^2+z_3^2)-(x_1^2+y_1^2+z_1^2)\right)\\ 0.5\left((x_4^2+y_4^2+z_4^2)-(x_1^2+y_1^2+z_1^2)\right)\end{bmatrix}=\begin{bmatrix}b_1\\ b_2\\ b_3\end{bmatrix}. \tag{11}$$

According to Cramer's Rule, determinant equation $AX=b$ will get a unique solution only if $|A|\neq 0$. In fact, we could easily prove that $|A|\neq 0$ means four points are coplanar. So, when $|A|\neq 0$, then we can get a unique solution for $AX=b$. That is

$$x_0=\frac{\det(A_1)}{\det(A)},\quad y_0=\frac{\det(A_2)}{\det(A)},\quad z_0=\frac{\det(A_3)}{\det(A)} \tag{12}$$

A_i is the matrix formed by replacing the i[th] column of by the column vector.

For example, $A_2 = \begin{bmatrix} (x_2 - x_1) & b_1 & (z_2 - z_1) \\ (x_3 - x_1) & b_2 & (z_3 - z_1) \\ (x_4 - x_1) & b_3 & (z_4 - z_1) \end{bmatrix}$, and so forth.

Proof completed.

Now, we could recover sphere center based on theorem 2. Dealer requests 3 participants shadow coordinates firstly, and mark them as $p_1^E =< x_1, y_1, z_1 >$, $p_2^E =< x_2, y_2, z_2 >$ and $p_3^E =< x_3, y_3, z_3 >$. As is well-known, three points must be in the locus of a same circular ring. We are sure that the circular ring based on p_1^E, p_2^E and p_3^E are on the destination sphere. Then, dealer only needs to select the fourth participant coordinate data $p_4^E =< x_4, y_4, z_4 >$ to satisfy

$$\det(A) = \begin{bmatrix} (x_2 - x_1) & (y_2 - y_1) & (z_2 - z_1) \\ (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) \\ (x_4 - x_1) & (y_4 - y_1) & (z_4 - z_1) \end{bmatrix} \neq 0 \quad (13)$$

,

which means these four points are non-coplanar. We use theorem 2 to compute the sphere center $p_0 =< x_0, y_0, z_0 >$ by equation (12) if the given four points meet equation (13), or else the dealer needs to reselect the fourth node and repeat the computing process to reconstruct the sphere center.

Since four shadows can recover the sphere center, *K* (*4*≤*K*≤*n*) shadows can also recover it in proposed scheme. The computing processes are similar.

4.3. Reconstruct Original Secret from Sphere Center

Once we have recovered sphere center $p_0 =< x_0, y_0, z_0 >$ in the above section, the secret can be reconstructed by algorithm 5 which is a reverse process to algorithm 3.

```
Algorithm 5: program GetSecretFromCenter (ID0,P0)
  (1)Get private coordinate <Prix0, Priy0, Priz0> from
node(ID0)
  (2)x0′←(Prix0 OR (P0.x0 SHL 16))
  (3)y0′←(Priy0 OR (P0.y0 SHL 16))
  (4)z0′←(Priz0 OR (P0.z0 SHL 16))
  (5)P0′←<x0′, y0′, z0′ >
  (6)IDtemp←SHA(ID0)
  (7)<xid0,yid0,zid0>←GetCoordinateForID(IDtemp)
```

```
  (8)<x_temp,y_temp,z_temp>←<x_0' ⊗x_id0, y_0' ⊗y_id0, z_0' ⊗z_id0>
  (9)S_temp←((x_temp SHL length(y_temp)) OR y_temp)
  (10)z_verify←SHA(S_temp) ⊗SHA(ID_0)
  (11)If (z_verify <> z_temp) then
  (12)  return "Error Secret!"
  (13)End if
  (14)S_0←S_temp ⊗ID_0
  (15)return S_0
end.
```

Up to now, we can reconstruct the original secret.

5. Analysis for Capabilities of the Proposed Scheme

In this section, we will discuss the capabilities of our proposed scheme like effectiveness, verifiability and proactivity. At the end of this section, we will analyze performance of the proposed scheme over computation, storage and communication.

5.1. Effectiveness

Traditional secret sharing schemes usually need a mutually trusted third party (TTP) to generate and distribute shares to participants. It has single-point-of-failure and is not appropriate to completely distributed systems which have no TTPs. Our proposed scheme is adapted to quite distribute networks, such as peer-to-peer networks and ad hoc networks. Any node could be a secret dealer and also could be a participant to hold shadow for other nodes.

Nodes in real networks are dynamical. Comparing to traditional scheme, dealer needs not to select all participants to generate and distribute shadows during real engineering, and the dealer can tolerate participants' fault or breakdown. According theorem 1 and theorem 2, it is enough to select only 4 non-coplanar points in theory. Thus, our proposed scheme is effective in enduring dynamic networks.

In practice, all the data to be processed in our algorithms is binary serial. Thus, the proposed scheme could be used for many Internet applications. During the processes of our scheme, secret could be easily converted to shadows, and K ($\geqslant 4$) shadows could easily recover secret according a series of equations.

5.2. Verifiability to Dishonest Participants Behaviors

Secret reconstruction depends on participants' honest feedbacks. However, a participant might lie about his own share to gain access to other shares. The verifiable secret sharing schemes [10-12] can detect, with some probability, a dishonest behavior of some users in the reconstructing phase. This feature is very important, for instance, in e-voting protocols based on secret sharing schemes.

In our scheme, once one or more participants provide fault shadows, the recovered result can detect the error. For example, there are four shadows provided by participants, p_1^E, p_2^E, p_3^E, and p_4^E. Then, equations (12) and (13) get a sphere center $p_0 =< x_0, y_0, z_0 >$. According to algorithm 2, a correct result $< x_0, y_0, z_0 >$ should satisfy

$$SHA(x_0 y_0) = z_0 \otimes SHA(ID_0). \tag{14}$$

As long as one of the shadows is incorrect, the recovered result $< x_0, y_0, z_0 >$ won't meet equation (14).

5.3. Proactivity to Attackers

If the participants store their shares on insecure computer servers, an attacker could crack in and steal the shares. Since it is not often practical to change the secret, the uncompromised (Shamir-style) shares should be updated in a way that they generate the same secret, yet the old shares are invalidated.

Proactive secret sharing is a method to update shares in a secret sharing scheme periodically such that an attacker has less time to compromise shares. This contrasts a non-proactive scheme where if the threshold number of shares is compromised during the lifetime of the secret, the secret is compromised [18][19].

In our case, assume there are four (or more) participants, so-called node ID_1, node ID_2, node ID_3, and node ID_4, and store their shadow coordinates (shares) on insecure computer. We also assume the four shadow coordinates (p_1^E, p_2^E, p_3^E, and p_4^E) are non-coplanar, and an attacker steals all of the four shares from participants. So, we analyze the proactivity of our scheme in the following two kinds of attacks.

(1) Simple collusion attack.

In this case, attacker doesn't know about our recovering algorithm and doesn't know how to compute the coordinates. At the same time, no information in a single shadow is useful as it is handled by algorithm (1), (2), (3) and (4) and extended by equation (7). Thus, any arbitrary combination for the shadow doesn't work.

(2) Algorithm based collusion attack.

In this case, attacker masters our scheme well, and can recover a sphere center while shadows are non-coplanar based on equations (12). Then, attacker could attempt

$$\left[p_1^E, p_2^E, p_3^E, p_4^E\right] \Rightarrow p_0 :< x_0, y_0, z_0 > . \tag{15}$$

Although $p_0 :< x_0, y_0, z_0 >$ here is a right sphere center, it is not a real secret. The attacker needs to do more according to algorithm (3) and (4) as you can see from figure 3. In fact, our scheme deals with secret by

$$\begin{aligned} &(S_0 \otimes ID_0) \\ &\rightarrow \left(S_{temp} :< x_0, y_0, z_0 >\right) \otimes \left(SHA(ID_0) :< x_{id0}, y_{id0}, z_{id0} >\right) \\ &\xrightarrow{IntermediateCoordinate} p_0' :< x_0', y_0', z_0' > \\ &\rightarrow \begin{cases} Pri_0 :< Pri_{x0}, Pri_{y0}, Pri_{z0} > \\ p_0 :< x_0, y_0, z_0 > \end{cases} \end{aligned} \tag{16}$$

In the above inference equation, a temporary sphere center $S_{temp} :< x_0, y_0, z_0 >$ is shifted to be a quite new intermediate center $p_0' :< x_0', y_0', z_0' >$ after a series of transformations. It is just like one sphere shifts to another space in the spherical coordinates. Moreover, the coordinate $p_0' :< x_0', y_0', z_0' >$ is divided into two coordinates; one is the final sphere center $p_0 :< x_0, y_0, z_0 >$, and another is private coordinate $Pri_0 :< Pri_{x0}, Pri_{y0}, Pri_{z0} >$ which is stored locally in dealer. Besides to test which node is dealer, attacker needs to attempt numerous private coordinates in order to get a correct private key at the same time. Thus, relatively speaking, the proposed scheme can be proactive to such attackers.

By the way, there exists another attack type. Probably, some participants may collude secretively to defraud secret from their own shadows. This phenomenon is usually called collusion attack, which is a main attack type today to Internet applications. According to the analysis above, the proposed scheme can also be proactive and robust to such collusion attacks based on its special design.

5.4. Performance of Computation, Storage and Communication

We have introduced our proposed scheme in the above sections, including how to initialize secret to be a sphere center, how to generate shadows around a personalized sphere, and how to distribute shadows. In this section, we will discuss the capabilities of our proposed scheme like effectiveness, verifiability

and proactivity. At the end of this section, we will analyze performance of the proposed scheme over computation, storage and communication.

We focus on computation amounts for proposed scheme firstly. SHA (it is SHA-2 with 512 bits in this paper) and bit-operations are the main computation style in both distribution and reconstruction processes while matrix operations are only used in reconstruction procedures. As the running time of bit-operations could be ignored, we mark a computation of SHA as "C_a" and a matrix operation as "C_m". Such as, the distribution algorithm (4) needs about "$n*C_a$" computation amounts while reconstruction algorithm (5) needs about "$3*C_a + 3*C_m$".

At the same time, storage space should be allocated during the computation. As you can see, all of the coordinates are same in scale, whether the initialized secret or the final shadow coordinates. So, we assume that one coordinate is a storage unit. Thus, n storage units are allocated during distribution while (n+k) storage units allocated to reconstruction.

Communications between nodes also occur. The distribution procedure needs almost n times to share shadows with n participants. It is a little difficult during reconstruction. As fault secret may exist in algorithm (5), we set p ($p \geqq 0$) to stand for fault probable times here. Once fault secret occurs, we need another k communications to recover the original secret.

Finally, we conclude the performance in table 1 below.

Table 1. Performance of computation, storage space and communications of proposed scheme

Algorithms	Computation during distribution	Computation during reconstruction	Storage Space	Communication
Algorithm 1 (A1)	$2*C_a$	NONE	1	0
Algorithm 2 (A2)	Only bit-operations	Only bit-operations	1	0
Algorithm 3 (A3)	A1+A2+C_a	NONE	2	1
Algorithm 4 (A4)	A3+n*A2+ $n*C_a$ = $O(n*C_a)$	NONE	O(n)	O(n)
Theorem 2 (T2)	NONE	$O(3*C_m)$	O(n+k)	O(k*(1+p))
Algorithm 5 (A5)	NONE	A2+T2+$3*C_a$ = $O(3*C_a+3*C_m)$	O(n+k)	O(k*(1+p))

From table 1, we could conclude that the proposed scheme has a relatively rational performance of computation and communication. The storage space is also rational. Then we continue to compare proposed scheme with other schemes.

As Shamir's scheme or Blakley's scheme has no verifiability and proactivity, we compare the proposed scheme with other scheme that has same

capability, such as HERZBERG's scheme [18], HARM's scheme [20] and TANG's scheme [21].

The following table 2 shows our conclusions in computation complexity and communication amounts of these schemes. By the way, we mark a modular exponentiation operation as C_e which appears in HARM's scheme. As we can see from it, our scheme has advantage in computation and communication complexity while a modular exponentiation has higher order of magnitude than a matrix operation or a SHA operation.

Table 2. Comparison of computation complexity and communication amounts

Schemes	Computation distribution	recover	Communication distribution	recover
Scheme [18]	$O(n*lb^2(n))$	$O(n*lb^2(n))$	$O(n*lb(q))$	$O(n*lb(q))$
Scheme [20]	$O(3n*C_e)$	$O(2k*C_e)$	$O(3n)$	$O(k)$
Scheme [21]	$O(nk)$	$O(nk)$	$O(3n+2k)$	$O(3n+2k)$
Proposed Scheme	$O(n*C_a)$	$O(3*C_a+3*C_m)$	$O(n)$	$O(k*(1+p))$

6. Conclusions and Future Work

In this paper, we propose a novel distributed secret sharing scheme based on spherical coordinates. We prove the theorem that four non-coplanar points can determine a unique sphere center using analytic geometry principle. And algorithms are proposed to generate sphere center and sphere surface based on this theorem. Via coordinate transformations, we convert original secret to sphere center to construct sphere surface with participants' identifier information. Verifiability and proactivity mechanisms are considered to protect shares. In our scheme, four or more participants could recover the secret. Analysis for proposed scheme proves the robustness.

However, there are also a number of technical problems that merit attention but were not fully addressed in this paper, such as how to deal with very large numbers efficiently in our scheme, how to balance dealer's load during reconstruction dynamically and how to ensure all shadow coordinates are non-coplanar. We believe that some new and interesting approaches will be found by investigating and studying this problem.

Acknowledgement. Thank anonymous reviewers for their careful and constructive suggestions to this paper. This work is supported by the Doctor Program Foundation of Education Ministry of China under Grant No. 20110042120027; China Postdoctoral Science Foundation under Grant No. 2012M511166; the Fundamental Research Funds for the Central Universities of China under Grant No. N110417006 and No.

N110204003; the National Natural Science Foundation of China under Grant No. 61070162, No. 71071028 and No. 70931001; the Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20100042110025 and No. 20110042110024; the Specialized Development Fund for the Internet of Things from the ministry of industry and information technology of China.

References

1. Shamir, A.: How to share a secret. Communications of the ACM, Vol. 22, No.11, 612-613. (1979)
2. Blakley, G.: Safeguarding cryptographic keys. In Proceedings of AFIPS National Computer Conference. IEEE Computer Science, NEW YORK, USA, 313-317. (1979)
3. Tamir T.: Hierarchical Threshold Secret Sharing. Journal of Cryptology, Vol. 20: 237–264. (2007)
4. Mignotte, M.: How to share a secret. Lecture Notes in Computer Science, Vol. 149. Springer-Verlag, Berlin Heidelberg New York. (1983)
5. Asmuth, C. A., Bloom J.: A modular approach to key safeguarding. IEEE Transactions on Information Theory, Vol. 29, No. 2, 208-210. (1983).
6. HARN, L.: Digital signature with (t,n) shared verification based on discrete logarithms. ELECTRONICS LEITERS, Vol. 31, No. 3, 177. (1995)
7. HSU, C. L., WU, T. C.: Authenticated encryption scheme with (t, n) shared verification. IEE ProcComput Digit Tech, Vol. 145, No. 2, 117-120. (1998)
8. HAN, Y. L., YANG X.Y., SUN, J., Li, D.L.: Verifiable threshold cryptosystems based on elliptic curve. In the proceedings of the International Conference on Computer Network and Mobile Computing. Vol.10, 334-337. (2003)
9. Marsh, M. A., Schneider, F. B.: CODEX: a robust and secure secret distribution system. IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, 34-47. (2004)
10. Chor, B., Goldwasser, S., Micali, S, Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults. In Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science. Portland, Oregon, USA, 383–395. (1985)
11. Feldman P.: A practical scheme for non-interactive verifiable secret sharing. In Proceedings of the 28th IEEE Symposium on the Foundations of Computer Science. IEEE press, Los Angeles, California, USA, 427–437. (1987)
12. Pedersen, T. P.: Non-interactive and information-theoretic secure verifiable secret sharing. Lecture Notes in Computer Science, Vol. 576. Springer-Verlag, Berlin Heidelberg New York. (1992)
13. Halpern, J., Teague, V.: Rational secret sharing and multiparty computation:Extended abstract. In: 36th Annual ACM Symposium on Theory of Computing (STOC), pp. 623–632. ACM Press, New York. (2004)
14. Micali, S., Shelat, A.: Purely rational secret sharing. LNCS, vol. 5444. Springer-Verlag, Berlin Heidelberg New York. (2009)
15. Fuchsbauer, G., Katz, J., Naccache, D.: Efficient Rational Secret Sharing in Standard Communication Networks. LNCS Vol. 5978. Springer-Verlag, Berlin Heidelberg New York. (2010)
16. Guo, C., Chang, C. C., Qin, C.: A hierarchical threshold secret image sharing [J]. Pattern Recognition Letters, Vol. 33, No. 1, 83–91. (2012)

17. Wang, R. Z., Shyu, S. J.: Scalable secret image sharing. Signal Processing: Image Communication, Vol. 22, No. 4, 363–373. (2007)
18. Herzberg, A., Jarecki, S., Hugo, K., Yung, M.: Proactive Secret Sharing Or: How to Cope With Perpetual Leakage. In the proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology. Springer-Verlag, London, UK, 339–352. (1995)
19. Yevdokimov, A. A.:Dynamic system of proactive security. In proceedings of International Conference on Application of Information and Communication Technologies. IEEE Press, Baku, Azerbaijan, 1–4. (2009)
20. HARM L.: Efficient sharing (broadcasting) of multiple secrets. IEEE Proc Comput Digital Tech, Vol. 142, No. 3, 237-240. (1995)
21. TANG C M, WU D O, CHRONOPOULOS A T, et al. Efficient multi-party digital signature using adaptive secret sharing for low-power devices in wireless networks. IEEE Transactions on Wireless Communications, Vol. 8, No. 2, 882-889. (2009)
22. National Institute of Standards and Technology (NIST): Secure Hash Standard (SHS). (2002). [Online]. http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf. (Current 2012)
23. National Institute of Standards and Technology (NIST): NIST Selects Winner of Secure Hash Algorithm (SHA-3) Competition. NIST Tech Beat, October 2, 2012. [Online]. Available: http://www.nist.gov/itl/csd/sha-100212.cfm. (2012)
24. Tan, Z. H., Cheng W., Ma, Y., Zhu Z. L., Chang G. R.: P2PETrust: A novel distributed transaction history vector-based trust appraisal model for peer-to-peer e-commerce networks. Scientific Research and Essays, Vol. 6, No. 18, 3845-3857. (2011)
25. Stillwell J.: Mathematics and Its History (Third Edition). Springer Science+Business Media, New York, USA. (2010)

Zhenhua Tan was born in Hunan, China, in Jan, 1980. He received his M.Sc. and Ph.D. degrees in Computer Science from Northeastern University, China, in 2006 and 2009 respectively. He is now a postdoctor of Northeastern University, works for trust computing and secret protection. He has published more than 15 scientific and professional papers on security direction, and was an invited session chair of the ninth International Conference on Hybrid Intelligent System. He also teaches courses in Discrete Mathematics, Distribted Systems and Programming. His current research interests are in the area of information security, especially in security of distributed networks. **(tanzhenhua192@126.com, tanzh@mail.neu.edu.cn)**

Guangming Yang was born in Liaoning, China, in May, 1961. He is now a professor of Northeastern University, works for information security. He has published more than 30 scientific and professional papers. His current research interests are in the area of information security and cloud networks.

Wei Cheng was born in Liaoning, China, in Oct, 1970. He is now an associate professor of Northeastern University, works for complex networks. He has published more than 10 scientific and professional papers. His current research interests are in the area of information security and cloud networks.

Xingwei Wang was born in China in 1968. He received his M.Sc. and Ph.D. degrees in Computer Applications from Northeastern University, China, in 1992 and 1998 respectively. He is now a professor and a Ph.D. supervisor of Northeastern University, works for mobile wireless Internet, distributed networks and information security. He has published more than 100 scientific and professional papers on security direction. (**Corresponding author**)

Received: August 1, 2012; Accepted: December 6, 2012