A Proactive Secret Sharing Scheme in matrix projection method

Li Bai*

Department of Electrical and Computer Engineering,
Temple University,
Philadelphia, PA 19122, USA
E-mail: lbai@temple.edu
*Corresponding author

XuKai Zou

Department of Computer and Information Science,
Indiana University-Purdue University Indianapolis,
Indianapolis, IN 46202, USA
E-mail: xkzou@cs.iupui.edu

Abstract: Proactive Secret Sharing (PSS) scheme is a method to periodically renew n secret shares in a (k, n) threshold-based Secret Sharing Scheme (SSS) without modifying the secret, or reconstructing the secret to reproduce new shares. Traditionally, PSS schemes are developed for the Shamir's SSS which is a single SSS. Bai (2006) developed a multiple-secret sharing scheme using matrix projection. This paper presents a distributed PSS method for the matrix projection SSS. Once the new shares are updated, adversaries cannot discover the secrets from k shares which are mixed with past and present shares.

Keywords: cryptography; matrix projection; Pythagorean triples.

Reference to this paper should be made as follows: Bai, L. and Zou, X. (2009) 'A Proactive Secret Sharing Scheme in matrix projection method', *Int. J. Security and Networks*, Vol. 4, No. 4, pp.201–209.

Biographical notes: Li Bai is an Assistant Professor in the Electrical and Computer Engineering Department, Temple University. He completed his PhD Degree from Drexel University, Philadelphia, PA 2001. He has published over 30 papers on reliability, signal processing, and information security related topics. His research areas are reliability, wireless communication, dependable secure computing, parallel array signal processing, and information fusion.

Xukai Zou is a Faculty member with the Department of Computer and Information Sciences at Indiana University-Purdue University Indianapolis, USA. He completed his PhD Degree from University of Nebraska-Lincoln in 2000. His research focus is in applied cryptography, network security, and communication networks. He recently published two books and over 30 papers in security-related topics. His research has been supported by US government agencies such as National Science Foundation and Department of Veterans Affairs and industry such Cisco System Inc.

1 Introduction

Secret sharing is an effective way to distribute a secret among n parties, where each party holds one piece of the secret. Blakley (1979) and Shamir (1979) were credited for two initial designs of SSS in 1979 independently. Blakley's method regards a secret as a point in the k-dimensional hyperplane space. These n shares are constructed as k affine hyperplanes in this space. The solution of any k affine hyperplanes is the intersection point (or the secret). This scheme is not a perfect SSS because a person with a share knows that the secret is a point on the hyperplane (the share). In contrast, Shamir proposed a polynomial interpolation method, which satisfied two basic conditional information entropy requirements Karnin et al. (1983) in a perfect SSS. As a result, other secret sharing techniques were based on the Shamir SSS, including many techniques (Okada and Kurosawa, 2000; Beimel and Chor, 1998; Stinson, 1994) to show how the secret shares are authenticated. To avoid a share dealer distributing invalid secret shares (Feldman, 1987) and (Pedersen, 1991) proposed two

different non-interactive Verifiable Secret Sharing (VSS) schemes respectively. Since shareholders could also be blamed for wrong shares, they need to authenticate the shares when they receive the shares from the dealer. If shareholders cannot validate the secret shares with the dealer's multicast messages, they multicast an alert message that they have the wrong shares. However, the corrupt shareholders can purposely send out fake alert messages each time when the dealer distributes authenticated secret shares. Consequently, the interactive VSS schemes (Goldreich et al., 1987; Ben-Or et al., 1988) were proposed to detect who are the honest participants and who are the adversaries in the system.

Interestingly, a secret sharing system is still quite vulnerable (Herzberg et al., 1995) when a dynamic adversary determines to break into the system before the lifetime of the secret expires. Ben-Or et al. (1988) discussed a general theory for the distributed fault tolerance systems and presented some possible solutions to avoid such attacks. Among many different classifications of adversary attacks, one of the most notable ones is to classify the attacks as:

- passive adversary attacks
- active adversary attacks.

Where passive adversary attacks are primarily resulting in spoofing data without modification or corruption to the data. In contrast to the passive adversary attacks, the active adversary attacks are much more malicious wherein the adversaries can persistently attempt to infiltrate a system, and/or to damage or destroy data already stored in the system.

Herzberg et al. (1995) proposed the PSS scheme based on the Shamir SSS to address this problem. This method periodically renews the shares (without reconstructing the secret) so that it prevents an adversary from gaining the knowledge of the secret before it expires. To counter active adversary attacks, Herzberg et al. combined the ideas of the VSS technique to prevent dishonest participants (or compromised participants by active adversaries) from refusing to change the shares during the renew process, or introduce invalid secret shares. Marsh and Schneider (2004) developed an operational system – COrnell Data Exchange (CODEX). The system addressed the issue of dependability and proactive security of a secret sharing system in a distributed environment. Noticeably, all these schemes were based on the Shamir SSS.

This paper presents a PSS method on the SSS on the matrix projection method (Bai, 2006). The matrix projection method allows to share multiple secrets where the Shamir's technique permits only one secret to be shared. We have not seen anyone propose a different PSS method other than Herzberg's method on a multiple-secret sharing scheme in literatures. In lieu of this, our emphasis is on the passive attacks because we have not developed a suitable and efficient VSS scheme on the matrix projection secret sharing method for multiple secrets. This, however, does not mean that we cannot use Feldman's VSS to counter active adversary attacks. In this paper, we focused on developing a distributed PSS method to counter passive adversary attacks by using a renew matrix from the Pythagorean triples. We can still satisfy the properties of the PSS method:

- to update shares without reconstructing the secret
- to reveal the secret using any k updated shares
- to prevent the secret from being revealed by using k past and present shares.

To counter active adversary attacks like Herzberg's method, we need to implement similar VSS scheme like Feldman's method. However, it will be quite expensive because there are multiple secrets. In our future research, we will implement an effective VSS scheme before we can incorporate it with the proposed PSS scheme to counter active adversary attacks.

The rest of the paper will be organised as follows. Section 2 presents the proactive secret sharing model and some assumptions. Section 3 briefly describes the Shamir's SSS, the matrix projection secret sharing method and Herzberg's PSS method. Our proposed PSS method is discussed in Sections 4 and 5 presents the conclusion.

2 Model and assumptions

According to the model and assumptions described in Herzberg et al. (1995), we summarised them as follows:

1 The system consists of n servers[1] $\mathcal{A} = \{P_1, P_2, \ldots, P_n\}$ that will (proactively) share a secret s or a secret matrix S.

2 The system is securely and properly initialised.

3 All servers in $\mathcal{A}$ are connected to a common secure multicast medium C with the property that messages sent on C instantly reach every party connected to it and the property that an adversary cannot understand encrypted multicast messages even though the adversary can eavesdrop the multicast messages. It is worthy to mention that such a multicast channel is not difficult to implement since many mature secure group communication mechanisms have been developed (Zou et al., 2004).

4 The system is synchronised with a common global clock. The time is divided into time periods which is determined by the common global clock. At the beginning of each time period, the servers engage an update protocol. Once the update is completed, the servers hold new shares of the secret or secret matrix.

5 Each server in $\mathcal{A}$ has its own local source of randomness.

6 Adversary is connected to the channel C, and the adversary knows the non-secret data and the algorithm that each server performs. However, the adversary cannot modify messages sent to C by a server which the adversary does not have control.

7 Adversary is computationally bounded where it cannot break the underlying cryptographic primitives used for secure communication.

8 Once the server is updated, all information is refreshed. In other words, the adversary cannot leave a backdoor to get back into the server again.

Here, to simplify our discussion of the proposed PSS technique, we consider the case of passive adversary attacks where the adversary cannot inject or modify messages multicast on the channel C. In other words, all servers in $\mathcal{A}$ are 'honest' servers and they can reliably trigger and participate in an updating process. We will investigate the counter method for the active adversary attacks in future research when we find more suitable and efficient VSS for the matrix project technique.

3 Secret Sharing Schemes and Herzberg's schemes

In this section, we briefly review two SSS

- Shamir's method and its PSS scheme developed by Herzberg et al. (1995)
- matrix projection method.

3.1 Shamir's Secret Sharing Scheme

Shamir (1979) developed the idea of a (k, n) threshold-based secret sharing technique ($k \leq n$). The technique is to construct a polynomial function of order $(k-1)$ as,

$$f(x) = d_0 + d_1 x + d_2 x^2 + \cdots + d_{k-1} x^{k-1} \pmod{p},$$

where the value d_0 is the secret and p is a prime number. The secret shares are the pairs of values (x_i, y_i) where $y_i = f(x_i)$, $1 \leq i \leq n$ and $0 < x_1 < x_2 \cdots < x_n \leq p-1$.

The polynomial function $f(x)$ is destroyed after each server P_i possesses a pair of values (x_i, y_i) so that no single server knows what the secret value d_0 is. In fact, no groups of $(k-1)$ or fewer secret shares can be used to discover the secret d_0. On the other hand, when k or more secret shares are available, we can set up at least k equations $y_i = f(x_i)$ with k unknown parameters d_i's. The unique solution d_0 can be solved. Also, a Lagrange interpolation formula (Shamir, 1979) is commonly used to solve the secret value d_0 as the following formula

$$d_0 = \sum_{i=0}^{k} \left(\prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{-x_j}{x_i - x_j} \right) y_i \pmod{p}$$

where (x_i, y_j) are any k shares for $1 \leq i \leq k$. Shamir's SSS is regarded as a perfect SSS because knowing $(k-1)$ linear equations cannot expose any information about the secret.

3.2 Herzberg's Proactive Secret Sharing scheme

To periodically update shares is an effective way to protect a secret from being revealed by adversary attacks. Herzberg et al. (1995) developed a PSS technique for the Shamir's method. After the initialisation of Shamir's SSS, at the beginning of every time period, all 'honest' servers can trigger an update phase in which the servers perform a share renewal protocol. The shares computed in period t are denoted by using the superscript t, i.e., $(x_i, f^t(x_i))$, $t = 0, 1, \ldots$. We know that the secret d_0 at time $(t-1)$ is

$$d_0 = f^{(t-1)}(0).$$

The algorithm is to construct a new $(k-1)$ random polynomial function at each updating phase as,

$$\delta(x) = a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1} \pmod{p}, \qquad (1)$$

where $\delta(0) = 0$ so that $f^t(0) = f^{(t-1)}(0) + \delta(0) = d_0 + 0 = d_0$.

The Herzberg's share renew protocol for each server P_i, $i \in \mathcal{A}$, at the beginning of the time period t is as follows:

1 P_i picks $k-1$ random numbers $\{a_{im}\}$ from Z_p for $m = 1, 2, \ldots, (k-1)$. The numbers define a polynomail function $\delta_i(x) = a_{i1}x + a_{i2}x^2 + \cdots + a_{i(k-1)}x^{k-1} \pmod{p}$ in Z_p.

2 For all other servers P_j, P_i secretly sends $u_{ij} = \delta_i(x_j)$ to P_j.

3 After decrypting u_{ji}, $\forall j \in \{1, 2, \ldots, n\}$ P_i computes its new share as

$$f^t(x_i) = \left(f^{t-1}(x_i) + u_{1i} + u_{2i} + \cdots + u_{ni} \right) \pmod{p},$$

4 P_i erases all the variables it used except of its current secret share $y_i^t = f^t(x_i)$.

Since the $\delta(x)$ function does not have a constant term, consequently, any group of k or more servers can still compute d_0 by contributing their new shares. However, a combination of k shares using past and present shares cannot be used to reconstruct the secret. As a result, the secret is protected from being revealed by the passive adversaries.

3.3 Secret Sharing Scheme using matrix projection

Bai (2006) developed a SSS using matrix projection method. Here, we describe briefly about some basic properties of matrix projection.

Let A be an $m \times k$ matrix of rank k ($m \geq k > 0$), and

$$\mathbb{S} = A(A'A)^{-1}A',$$

where $(\bullet)'$ is the transpose of a matrix. The $m \times m$ matrix $\mathbb{S}$ is the projection matrix of matrix A.

We can also compute vectors v_i using k linearly independent $k \times 1$ vectors x_i,

$$v_i = Ax_i,$$

where $1 \leq i \leq k$. These $m \times 1$ vectors v_i can be placed in

$$B = \left[v_1\, v_2 \ldots v_k\right].$$

The projection of matrix A is the same as the projection of matrix B. We can show that in the following theorem.

Theorem 3.1 (Invariance Theorem)**:** *For an $m \times k$ matrix A of rank k $(m \geq k > 0)$ and an $m \times k$ matrix $B = [v_1\ v_2\ \ldots\ v_k]$ where $v_i = Ax_i$ for $i = 1, 2, \ldots, k$ and x_is are k linearly independent $k \times 1$ vectors. The projection of matrix A is the same as that of matrix B, or $\mathbb{S} = A(A'A)^{-1}A' = B(B'B)^{-1}B'$.*

Proof: Since $B = [v_1\ v_2\ \ldots\ v_k]$ and $v_i = Ax_i$,

$$\begin{aligned} B &= [v_1\ v_2\ \ldots\ v_k] \\ &= [Ax_1\ Ax_2\ \ldots\ Ax_k] \\ &= A[x_1\ x_2\ \ldots\ x_k]. \end{aligned} \tag{2}$$

To simplify notations, we write $X = [x_1\ x_2\ \ldots\ x_k]$. The $k \times k$ matrix X is a full rank matrix because it has k linearly independent column vectors x_is. The equation (2) becomes

$$B = AX. \tag{3}$$

Substitute equation (3) into the projection of matrix B

$$\begin{aligned} B(B'B)^{-1}B' &= AX((AX)'AX)^{-1}(AX)' \\ &= AX(X'A'AX)^{-1}(AX)'. \end{aligned}$$

Since the matrices X and X' are invertible,

$$\begin{aligned} B(B'B)^{-1}B' &= AXX^{-1}(A'A)^{-1}(X')^{-1}X'A' \\ &= A(A'A)^{-1}A'. \end{aligned}$$

The result shows that

$$\mathbb{S} = A(A'A)^{-1}A' = B(B'B)^{-1}B'. \tag{4}$$

□

Denote projection matrix $\mathbb{S} = (s_{ij})$ for $1 \leq i, j \leq m$, and it has following properties:

1 $\mathbb{S}$ is symmetric

2 $\mathbb{S}A = A$

3 $\mathbb{S}v_i = v_i$

4 $\mathbb{S}$ is idempotent, i.e., $\mathbb{S}^2 = \mathbb{S}$

5 $\mathrm{tr}(\mathbb{S}) \pmod p = \mathrm{rank}(\mathbb{S}) = k$ where $\mathrm{tr}(\mathbb{S}) = \sum_{i=1}^{m} s_{ii}$.

The invariance property of matrix projection can be used in secret sharing system to share multiple secrets. The detail of the scheme can be found in Bai (2006). Bai proved that the matrix projection is a strong ramp SSS with k access levels.

Here, we present the procedure in two phases including a numerical example. The procedure is as follows:

- Construction of shares from a secret matrix S

  1 Construct a random $m \times k$ matrix A of rank k where $m > 2k - 3$

  2 Choose n linearly independent $k \times 1$ random vectors x_i

  3 Calculate shares $v_i = (A \times x_i) \pmod p$ for $1 \leq i \leq n$

  4 Compute a projection matrix $\mathbb{S} = (A(A'A)^{-1}A') \pmod p$

  5 Solve a remainder matrix $R = (S - \mathbb{S}) \pmod p$

  6 Destroy the matrix A, the vector x_is, the projection matrix $\mathbb{S}$, the secret matrix S

  7 Distribute n shares v_i to servers in $\mathcal{A}$ and make the remainder matrix R publicly known.

- Secret reconstruction

  1 Collect k shares from any k servers in $\mathcal{A}$, say the shares are $v_1, v_2, \ldots, v_k$ and construct a matrix $B = [v_1\ v_2\ \ldots\ v_k]$

  2 Calculate the projection matrix $\mathbb{S} = (B(B'B)^{-1}B') \pmod p$

  3 Verify that the trace of the projection matrix $\mathrm{tr}(\mathbb{S}) \pmod p = k$

  4 Compute the secret $S = \mathbb{S} + R \pmod p$.

To demonstrate the method, we show a simple $(2, 4)$ threshold-based example with the prime modulus $p = 19$ and the secret matrix

$$S = \begin{bmatrix} 10 & 12 & 4 & 7 & 8 \\ 5 & 10 & 9 & 1 & 3 \\ 3 & 2 & 1 & 11 & 14 \\ 4 & 3 & 8 & 5 & 1 \\ 2 & 4 & 2 & 3 & 10 \end{bmatrix}.$$

To construct the shares, we choose a 4×2 random matrix A of rank 2 that

$$A = \begin{bmatrix} 10 & 1 \\ 7 & 2 \\ 8 & 4 \\ 1 & 1 \\ 3 & 5 \end{bmatrix}.$$

The values of $m = 5$ and $k = 2$ satisfy the condition of secret sharing where $m > 2k - 3$. It is a necessary

condition for strong protection of the secret matrix S. Choose $n = 4$ linearly independent vectors as

$$x_1 = \begin{bmatrix} 1 \\ 17 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 7 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{and} \quad x_4 = \begin{bmatrix} 1 \\ 9 \end{bmatrix}.$$

Next we compute $v_i = Ax_i$ for $i = 1, 2, 3, 4$,

$$v_1 = \begin{bmatrix} 8 \\ 3 \\ 0 \\ 18 \\ 12 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 17 \\ 2 \\ 17 \\ 8 \\ 0 \end{bmatrix}, \quad v_3 = \begin{bmatrix} 11 \\ 9 \\ 12 \\ 2 \\ 8 \end{bmatrix}, \quad \text{and} \quad v_4 = \begin{bmatrix} 0 \\ 6 \\ 6 \\ 10 \\ 10 \end{bmatrix}.$$

The projection matrix $\mathbb{S}$ is

$$\mathbb{S} = (A(A'A)^{-1}A') \pmod{19} = \begin{bmatrix} 8 & 8 & 5 & 1 & 14 \\ 8 & 14 & 11 & 11 & 5 \\ 5 & 11 & 2 & 13 & 14 \\ 1 & 11 & 13 & 16 & 1 \\ 14 & 5 & 14 & 1 & 0 \end{bmatrix},$$

then the remainder matrix R is equal to

$$R = (S - \mathbb{S}) \pmod{19} = \begin{bmatrix} 2 & 4 & 18 & 6 & 13 \\ 16 & 15 & 17 & 9 & 17 \\ 17 & 10 & 18 & 17 & 0 \\ 3 & 11 & 14 & 8 & 0 \\ 7 & 18 & 7 & 2 & 10 \end{bmatrix}.$$

The matrix R is made publicly known. We can destroy A, x_is, $\mathbb{S}$ and S, then we distribute four v_i shares to four different servers in $\mathcal{A}$.

When any two servers' shares are chosen, they can form a matrix B. For example, these two shares are v_1 and v_2 to form the matrix B as

$$B = \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} 8 & 17 \\ 3 & 2 \\ 0 & 17 \\ 18 & 8 \\ 12 & 0 \end{bmatrix}.$$

The projection matrix of B, $\mathbb{S}$ is

$$\mathbb{S} = (B(B'B)^{-1}B') \pmod{19} = \begin{bmatrix} 8 & 8 & 5 & 1 & 14 \\ 8 & 14 & 11 & 11 & 5 \\ 5 & 11 & 2 & 13 & 14 \\ 1 & 11 & 13 & 16 & 1 \\ 14 & 5 & 14 & 1 & 0 \end{bmatrix}.$$

We can validate that $\operatorname{tr}(\mathbb{S}) \pmod{19} = 40 \pmod{19} = 2 = k$. The secret matrix S is obtained by the remainder matrix R and the projection matrix $\mathbb{S}$ as

$$S = (R + \mathbb{S}) \pmod{19} = \begin{bmatrix} 10 & 12 & 4 & 7 & 8 \\ 5 & 10 & 9 & 1 & 3 \\ 3 & 2 & 1 & 11 & 14 \\ 4 & 3 & 8 & 5 & 1 \\ 2 & 4 & 2 & 3 & 10 \end{bmatrix}.$$

The reconstructed matrix is the same as the secret matrix, and the shares are $1/m$ of the size of the secret matrix (for our case, it is $1/5$ because $m = 5$). This matrix projection method is not a perfect SSS, but it is a multiple-secret sharing scheme and has a strong protection on the secrets.

4 Proposed Proactive Secret Sharing scheme

In contrast, the matrix projection method cannot be updated easily by using the Herzberg's PSS technique. For n secret shares v_i, we need to determine a different way to renew these vectors so that we can protect the secret. In this paper, we developed a different PSS technique by using the Pythagorean triples.

Here, we review the Pythagorean triples (Weisstein, 2003). The Pythagorean triples are the three integer values (Z_1, Z_2, Z_3) that satisfy the following equation:

$$Z_1^2 + Z_2^2 = Z_3^2.$$

The general form of a Pythagorean triples is

$$Z_1 = a^2 - b^2, \quad Z_2 = 2ab, \quad Z_3 = a^2 + b^2 \tag{5}$$

where a and b are positive integers $(a > b)$. If we use the triples to construct a $k \times k$ matrix $L = (l_{ij})$ with two random index numbers g and h for $1 \le i, j, g, h \le k$ and $g \ne h$, we can have

$$l_{ij} = \begin{cases} \dfrac{Z_1}{Z_3} \pmod{p} & i = j = g, \\ \dfrac{Z_1}{Z_3} \pmod{p} & i = j = h, \\ \dfrac{Z_2}{Z_3} \pmod{p} & i = g, j = h, \\ -\dfrac{Z_2}{Z_3} \pmod{p} & i = h, j = g, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

the matrix L is an orthonormal matrix because $L^{-1} = L'$.

For example, if $k = 2$, $Z_1 = 3, Z_2 = 4, Z_3 = 5$ and $p = 19$, we can express the matrix L as

$$L = \begin{bmatrix} \dfrac{3}{5} & \dfrac{4}{5} \\ -\dfrac{4}{5} & \dfrac{3}{5} \end{bmatrix} \pmod{19} = \begin{bmatrix} 12 & 16 \\ 3 & 12 \end{bmatrix}$$

because $(5^{-1} \pmod{19}) = 4$, and its inverse matrix

$$L^{-1} = L' = \begin{bmatrix} 12 & 3 \\ 16 & 12 \end{bmatrix}.$$

We can also verify that

$$L \times L' (\text{mod } 19) = \begin{bmatrix} 12 & 16 \\ 3 & 12 \end{bmatrix} \times \begin{bmatrix} 12 & 3 \\ 16 & 12 \end{bmatrix} (\text{mod } 19)$$
$$= \begin{bmatrix} 400 & 228 \\ 228 & 153 \end{bmatrix} (\text{mod } 19) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

It suggests that the matrix L is an orthonormal matrix.

For another example, if $k = 3, g = 1, h = 3, Z_1 = 3$, $Z_2 = 4, Z_3 = 5$ and $p = 19$, we can express the matrix L as

$$L = \begin{bmatrix} \frac{3}{5} & 0 & \frac{4}{5} \\ 0 & 1 & 0 \\ -\frac{4}{5} & 0 & \frac{3}{5} \end{bmatrix} (\text{mod } 19) = \begin{bmatrix} 12 & 0 & 16 \\ 0 & 1 & 0 \\ 3 & 0 & 12 \end{bmatrix}.$$

This matrix L is also an orthonormal matrix along two other forms as

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 12 & 16 \\ 0 & 3 & 12 \end{bmatrix}, \quad \text{or} \quad L = \begin{bmatrix} 12 & 16 & 0 \\ 3 & 12 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

After we get the matrix L, we can construct another $m \times m$ orthonormal matrix T as,

$$T = \begin{bmatrix} I_{m-k} & \vdots & 0_{(m-k)\times k} \\ \cdots\cdots & \vdots & \cdots\cdots \\ 0_{k\times(m-k)} & \vdots & L \end{bmatrix},$$

where I_{m-k} is an identity matrix with the dimension of $(m-k) \times (m-k)$. Similarly, $T^{-1} = T'$. For example, a 5×5 matrix T can be constructed by using the matrix L matrix as,

$$T = \begin{bmatrix} 1 & 0 & 0 & \vdots & 0 & 0 \\ 0 & 1 & 0 & \vdots & 0 & 0 \\ 0 & 0 & 1 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \vdots & \cdots & \cdots \\ 0 & 0 & 0 & \vdots & 12 & 16 \\ 0 & 0 & 0 & \vdots & 3 & 12 \end{bmatrix}, \tag{7}$$

for $k = 2$. Clearly, $TT' = I$. If we denote v_i^t as a renewed secret share at time t from an old share v_i for $i = 1, 2, \ldots, n$,

$$v_i^t = T \times v_i,$$

the renewed shares can produce the same projection matrix $\mathbb{S}$. We present a theorem as follows.

Theorem 4.1: *For any $m \times m$ orthonormal matrix T, if $v_i^t = Tv_i$ where v_i^ts and v_i are the past and present shares respectively for $i = 1, 2, \ldots, n$. Suppose the matrix $\mathbb{S}^t$ is the projection matrix of any k renewed secret shares $\{v_i^t\}$, and the matrix $\mathbb{S}$ is the projection matrix of any k past secret shares $\{v_i\}$, we have $\mathbb{S}^t = T\mathbb{S}T'$.*

Proof: Suppose we choose any k present shares and past shares. Say that they are v_i^t and v_i respectively, and

$$v_i^t = T \times v_i,$$

where $i = 1, 2, \ldots, k$.

The renewed k shares v_i^ts can be used to construct a matrix B^t as,

$$B^t = [v_1^t \ v_2^t \ \ldots \ v_k^t]$$
$$= [Tv_1 \ Tv_2 \ \ldots \ Tv_k] = T[v_1 \ v_2 \ \ldots \ v_k]$$
$$= TV.$$

Since the matrix $\mathbb{S}$ is the projection matrix of V, we have

$$\mathbb{S} = V(V'V)^{-1}V'.$$

To determine the renewed projection matrix $\mathbb{S}^t$, we have

$$\mathbb{S}^t = B^t((B^t)'B^t)^{-1}(B^t)'$$
$$= TV(V'T'TV)^{-1}V'T'$$
$$= TV(V'V)^{-1}V'T'$$
$$= T\mathbb{S}T'. \qquad \square$$

Apparently, a renewed projection matrix $\mathbb{S}^t$ is related with the original projection matrix $\mathbb{S}$. If we partition both matrices, we have

$$\mathbb{S} = \begin{bmatrix} \mathbb{S}_{11} & \vdots & \mathbb{S}_{12} \\ \ldots & \vdots & \ldots \\ \mathbb{S}_{21} & \vdots & \mathbb{S}_{22} \end{bmatrix} \quad \text{and} \quad \mathbb{S}^t = \begin{bmatrix} \mathbb{S}_{11}^t & \vdots & \mathbb{S}_{12}^t \\ \ldots & \vdots & \ldots \\ \mathbb{S}_{21}^t & \vdots & \mathbb{S}_{22}^t \end{bmatrix},$$

where matrices $\mathbb{S}_{11}$ and $\mathbb{S}_{11}^t$ are $(m-k) \times (m-k)$ matrices, $\mathbb{S}_{12}$ and $\mathbb{S}_{12}^t$ are $(m-k) \times k$ matrices, $\mathbb{S}_{21}$ and $\mathbb{S}_{21}^t$ are $k \times (m-k)$ matrices, and $\mathbb{S}_{22}$ and $\mathbb{S}_{22}^t$ are $k \times k$ matrices. Accordingly, we have the following theorem.

Theorem 4.2: *Consider the partitioned matrices $\mathbb{S}^t$ and $\mathbb{S}$, the following relationships hold as*

$$\mathbb{S}_{11}^t = \mathbb{S}_{11} \quad \mathbb{S}_{12}^t = \mathbb{S}_{12}L'$$
$$\mathbb{S}_{21}^t = L\mathbb{S}_{21} \quad \mathbb{S}_{22}^t = L\mathbb{S}_{22}L'.$$

Proof: From Theorem 4.1, we know that $\mathbb{S}^t = T\mathbb{S}T'$, and the matrix T as

$$T = \begin{bmatrix} I_{m-k} & \vdots & 0_{(m-k)\times k} \\ \cdots\cdots & \vdots & \cdots\cdots \\ 0_{k\times(m-k)} & \vdots & L \end{bmatrix}.$$

Therefore,

$$\begin{bmatrix} \mathbb{S}_{11}^t & \vdots & \mathbb{S}_{12}^t \\ \cdots & \vdots & \cdots \\ \mathbb{S}_{21}^t & \vdots & \mathbb{S}_{22}^t \end{bmatrix} = \begin{bmatrix} I_{m-k} & \vdots & 0_{(m-k)\times k} \\ \cdots & \vdots & \cdots \\ 0_{k\times(m-k)} & \vdots & L \end{bmatrix} \times \begin{bmatrix} \mathbb{S}_{11} & \vdots & \mathbb{S}_{12} \\ \cdots & \vdots & \cdots \\ \mathbb{S}_{21} & \vdots & \mathbb{S}_{22} \end{bmatrix} \begin{bmatrix} I_{m-k} & \vdots & 0_{(m-k)\times k} \\ \cdots & \vdots & \cdots \\ 0_{k\times(m-k)} & \vdots & L' \end{bmatrix} = \begin{bmatrix} \mathbb{S}_{11} & \vdots & \mathbb{S}_{12}L' \\ \cdots & \vdots & \cdots \\ L\mathbb{S}_{21} & \vdots & L\mathbb{S}_{22}L' \end{bmatrix}.$$

Hence,

$$\mathbb{S}_{11}^t = \mathbb{S}_{11} \quad S_{12}^t = S_{12}L'$$
$$\mathbb{S}_{21}^t = L\mathbb{S}_{21} \quad \mathbb{S}_{22}^t = L\mathbb{S}_{22}L'. \qquad \square$$

To proactively share the projection matrix (or secret), we can see that we cannot share the whole projection matrix. Rather, we can only share the partitioned $(m-k)\times(m-k)$ matrix $\mathbb{S}_{11}$. Similar, we cannot share the original matrix S, but its partitioned matrix S_{11} respectively.

4.1 An example for proactive matrix projection sharing

Consider the previous example shown in Section 3.3, we can only proactively share the secret matrix

$$S_{11} = \begin{bmatrix} 10 & 12 & 4 \\ 5 & 10 & 9 \\ 3 & 2 & 1 \end{bmatrix}$$

where the updating matrix T is shown in equation (7). The servers in $\mathcal{A}$ can compute the following new shares at time t,

$$v_1^t = T \times \begin{bmatrix} 8 \\ 3 \\ 0 \\ 18 \\ 12 \end{bmatrix} = \begin{bmatrix} 8 \\ 3 \\ 0 \\ 9 \\ 8 \end{bmatrix}, \quad v_2^t = T \times \begin{bmatrix} 17 \\ 2 \\ 17 \\ 8 \\ 0 \end{bmatrix} = \begin{bmatrix} 17 \\ 2 \\ 17 \\ 1 \\ 5 \end{bmatrix},$$

$$v_3^t = T \times \begin{bmatrix} 11 \\ 9 \\ 12 \\ 2 \\ 8 \end{bmatrix} = \begin{bmatrix} 11 \\ 9 \\ 12 \\ 0 \\ 7 \end{bmatrix}, \text{ and } \quad v_4^t = T \times \begin{bmatrix} 0 \\ 6 \\ 6 \\ 10 \\ 10 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \\ 6 \\ 14 \\ 17 \end{bmatrix}. \quad (8)$$

If two shares (say 1 and 3) are combined to reconstruct the secret S_{11}, we can form a matrix B^t using shares v_1^t and v_3^t as

$$B^t = [v_1^t \ v_3^t] = \begin{bmatrix} 8 & 11 \\ 3 & 9 \\ 0 & 12 \\ 9 & 0 \\ 8 & 7 \end{bmatrix},$$

the projection matrix $\mathbb{S}^t$ is computed as

$$\mathbb{S}^t = B^t((B^t)'B^t)^{-1}(B^t)' \pmod{19} = \begin{bmatrix} 8 & 8 & 5 & 8 & 0 \\ 8 & 14 & 11 & 3 & 17 \\ 5 & 11 & 2 & 0 & 17 \\ 8 & 3 & 0 & 9 & 8 \\ 0 & 17 & 17 & 8 & 7 \end{bmatrix}.$$

Clearly, even $\mathbb{S}^t \neq \mathbb{S}$, but we have

$$\mathbb{S}_{11}^t = \mathbb{S}_{11} = \begin{bmatrix} 8 & 8 & 5 \\ 8 & 14 & 11 \\ 5 & 11 & 2 \end{bmatrix}.$$

We can use the remainder matrix R to calculate

$$S_{11} = \mathbb{S}_{11}^t + R_{11} \pmod{19} = \begin{bmatrix} 10 & 12 & 4 \\ 5 & 10 & 9 \\ 3 & 2 & 1 \end{bmatrix}.$$

As a result, we renew the shares without reconstructing the secret (or the projection matrix), and we can still obtain the same secret matrix S_{11}. When m is a larger number, the process becomes more efficient because the matrix S_{11} has more elements to be shared.

However, if a passive adversary has

- an older share v_1
- a renewed share v_3^t,

then the adversary thinks that he/she obtained enough number of shares. A matrix $\widehat{B}$ is constructed by using these two vectors as

$$\widehat{B} = \begin{bmatrix} 8 & 11 \\ 3 & 9 \\ 0 & 12 \\ 18 & 0 \\ 12 & 7 \end{bmatrix}.$$

The adversary computes the projection matrix of $\widehat{B}$, the result is

$$\widehat{\mathbb{S}} = \widehat{B}(\widehat{B}'\widehat{B})^{-1}\widehat{B}' \pmod{19} = \begin{bmatrix} 16 & 12 & 6 & 7 & 5 \\ 12 & 7 & 12 & 7 & 18 \\ 6 & 12 & 5 & 2 & 9 \\ 7 & 7 & 2 & 14 & 1 \\ 5 & 18 & 9 & 1 & 17 \end{bmatrix}$$

$$\Longrightarrow \widehat{\mathbb{S}}_{11} = \begin{bmatrix} 16 & 12 & 6 \\ 12 & 7 & 12 \\ 6 & 12 & 5 \end{bmatrix} \neq \mathbb{S}_{11} = \begin{bmatrix} 8 & 8 & 5 \\ 8 & 14 & 11 \\ 5 & 11 & 2 \end{bmatrix}.$$

Consequently, the secret cannot be determined by simply obtaining a combination of k past shares and present shares. The secret matrix S_{11} is protected after the shares are renewed. This process will ensure the adversaries cannot learn the projection matrix $\mathbb{S}$ when the shares are periodically renewed.

4.2 Periodic share renewal scheme using matrix projection technique

In the initialisation of SSS of an $m \times m$ secret matrix S (or essentially share the $(m-k) \times (m-k)$ secret matrix S_{11}), a centralised dealer can generate a projection matrix $\mathbb{S}$ to produce n pieces of shares $v_i \in Z_{m\times 1}$ store in n servers in $\mathcal{A}$ where we leave the remainder matrix R_{11} to be publicly known (cannot be modified by anyone).

After the initialisation, at the beginning of every time period, all honest servers trigger an update phase in which the servers perform a share renewal protocol. The shares computed in period t are denoted as v_i^t, $t = 0, 1, \ldots$

The share renew protocol for each server P_i, $i \in \mathcal{A}$, at the beginning of the time period t is as follows:

1 P_i picks four nonzero random numbers: g_i & h_i from Z_k and a_i & b_i from Z_p.

2 P_i multicasts g_i, h_i, a_i and b_i on the secure channel C so that all other P_j can get these values.

3 After decrypting g_j, h_j, a_j and b_j, $\forall j \in \{1, 2, \ldots, n\}$ P_i uses these values to generate a set of Pythagorean triples to produce an orthonormal matrix L_j as shown in equations (5) and (6). Then, P_i derives its new share as

$$v_i^t = \begin{bmatrix} (v_i^{t-1})_{1:m-k} \\ \left(\prod_{j=1}^{n} L_j\right)(v_i^{t-1})_{m-k+1:m} \end{bmatrix},$$

where $(v_i^{t-1})_{1:m-k}$ is the vector contains the first $(m-k)$ elements in vector v_i^{t-1} and $(v_i^{t-1})_{m-k+1:m}$ is the vector contains the last k elements in vector v_i^{t-1}.

4 P_i erases all the variables it used except of its current secret share v_i^t.

4.2.1 Correctness of the PSS scheme using matrix projection

It is easy to prove that our PSS scheme is correct. Because each server can obtain the same $L = \left(\prod_{j=1}^{n} L_j\right)$ matrix where L_is are orthonormal matrices for $i = 1, 2, \ldots, n$, we can show that L is also an orthonormal matrix because

$$LL' = \left(\prod_{j=1}^{n} L_j\right)\left(\prod_{j=1}^{n} L_j\right)' = L_1 L_2 \ldots L_n L_n' \ldots L_2' L_1' = I.$$

Consequently, we can see that the server P_i's new shares

$$\begin{aligned} v_i^t &= \begin{bmatrix} (v_i^{t-1})_{1:m-k} \\ L(v_i^{t-1})_{m-k+1:m} \end{bmatrix} \\ &= \begin{bmatrix} I_{m-k} & \vdots & 0_{(m-k)\times k} \\ \cdots\cdots & \vdots & \cdots\cdots \\ 0_{k\times(m-k)} & \vdots & L \end{bmatrix} v_i^{t-1} \\ &= T v_i^{t-1}. \end{aligned}$$

for $i = 1, 2, \ldots, n$. According to Theorems 4.1 and 4.1, the matrix $\mathbb{S}_{11}$ remains the same after the updating process.

4.2.2 Secrecy of the PSS scheme using matrix projection

Using the similar argument as in Herzberg et al. (1995), we can prove the security of the proposed scheme as follows. Let $\mathfrak{A}$ be an adversary. Let K_1 be the set of k_1 servers whose shares in period $(t-1)$ (but not in period t) are compromised by $\mathfrak{A}$; let K_2 be the set of k_2 servers whose shares in both period $(t-1)$ and period t are compromised by $\mathfrak{A}$; let K_3 be the set of k_3 servers whose shares in period t (but not in period $(t-1)$) are compromised by $\mathfrak{A}$. Since k is the threshold, we can assume $k_1 + k_2 < k$ and $k_2 + k_3 < k$. Moreover, we will assume a clear worst case where $k_1 = k_3 = k - 1 - k_2$. We also denote $V_1 = \{v_i^{t-1}\}_{k_1}$ and $V_2 = \{v_i^{t-1}\}_{k_2}$ as sets of the shares in period $(t-1)$ corresponding to the servers in K_1 and K_2, respectively; and $\widehat{V}_2 = \{v_i^t\}_{k_2}$ and $V_3 = \{v_i^t\}_{k_3}$ as sets of the shares in period t corresponding to the servers in K_2 and K_3, respectively. We know that the shares in servers K_2 are related by an updating orthonormal matrix L as

$$\underbrace{(\widehat{V}_2)_{m-k_2:m}}_{k\times(k_2)} = \underbrace{L}_{k\times k}\underbrace{(V_2)_{m-k_2:m}}_{k\times(k_2)}.$$

The matrix L cannot be determined uniquely because $(\widehat{V}_2)_{m-k_2:m}$ and $(V_2)_{m-k_2:m}$ are $k \times (k_2)$ matrices and even in the worst case $k_2 = k - 2$ (and $k_1 = k_3 = 1$), they are not full rank matrices. A passive adversary $\mathfrak{A}$ needs to recover the updating matrix T in order to reconstruct the correct projection matrix. Since the updating matrix T cannot be determined without the corrected L matrix, the adversary cannot get a renewed share of a server in K_1 or an old share of a server in K_3. Consequently, the secret matrix will be protected from the passive attack. Also, the secret projection matrix $\mathbb{S}_{11}$ is independent from how the matrix L is chosen. In other words, the above argument holds for any projection matrix $\mathbb{S}_{11}$, or no information of $\mathbb{S}_{11}$ will be revealed to the adversary $\mathfrak{A}$.

4.3 Computation complexity of the PSS schemes

Also, we can compare the computation complexities of both PSS schemes. Herzberg's method requires to

compute equation (1) n times for n different shareholders. There are $\left(n\frac{(k-1)k}{2}\right)$ multiplications. Consequently, the computation complexity of Herzberg's method is $O(nk^2)$.

In contrast, it appears that our method requires to multiple matrix T with n different v_is. However, a simple updating method is to compute the $k \times k$ matrix L with the last k elements in v_is. As a result, the number of multiplication is nk^2 for protecting $(m-2)^2$ secrets. Per each secret, the computation complexity is $O\left(\frac{nk^2}{(m-2)^2}\right)$. Clearly, our method has much less computation complexity compared to Herzberg's method.

5 Conclusion

In this paper, we introduced a new, secure and distributed PSS scheme for the matrix projection secret sharing method. We demonstrate a different PSS scheme other than Herzberg's method. The procedure is archived by constructing a renewal matrix L from the Pythagorean triples. After the shares are updated, any k shares of past and present shares cannot be used to reveal the secret matrix. Our method is emphasised on protection against the passive attacks. In future research, we are interested in developing methods to countermeasure active attacks by implementing an effective VSS before we incorporate it with our PSS scheme.

Acknowledgement

The authors would like to thank the paper reviewers and the editor for their constructive comments which have lead to significant improvements in the paper.

References

Bai, L. (2006) 'A strong ramp secret sharing scheme using matrix projection', *Second International Workshop on Trust, Security and Privacy for Ubiquitous Computing*, Niagara-Falls, Buffalo, NY, pp.652–656.

Beimel, A. and Chor, B. (1998) 'Secret sharing with public reconstruction', *IEEE Trans. Inform. Theory*, Vol. 44, pp.1887–1896.

Ben-Or, M., Goldwasser, S. and Wigderson, A. (1988) 'Completeness theorems for non-cryptographic fault-tolerant distributed computation', *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, 2–4 May, Chicago, Illinois, pp.1–10.

Blakley, G. (1979) 'Safeguarding cryptographic keys', *Proceedings of the AFIPS 1979 National Computer Conference*, June, Arlington, VA, Vol. 48, pp.313–317.

Feldman, P. (1987) 'A practical scheme for non-interactive verifiable secret sharing', *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science (FOCS' 87)*, 12–14 October, IEEE Computer Society, Los Angeles, California, pp.427–437.

Goldreich, O., Micali, S. and Wigderson, A. (1987) 'How to play any mental game or a completeness theorem for protocols with honest majority', *Proceedings of the Nineteen Annual ACM Symposium on Theory of Computing*, New York City, NY, pp.218–219.

Herzberg, A., Jarecki, S., Krawczyk, H. and Yung, M. (1995) 'Proactive secret sharing or: how to cope with perpetual leakage', in Don Coppersmith (Ed.): *Advances in Cryptology – Crypto '95*, August, Santa Barbara, CA, pp.339–352.

Karnin, E.D., Greene, J.W. and Hellman, M.E. (1983) 'On secret sharing systems', *IEEE Trans. Inform. Theory*, Vol. IT-29, pp.35–41.

Marsh, M.A. and Schneider, F.B. (2004) 'CODEX: a robust and secure secret distribution system', *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, pp.34–47.

Okada, K. and Kurosawa, K. (2000) 'MDS secret-sharing scheme secure against cheaters', *IEEE Trans. Inform. Theory*, Vol. 46, pp.1078–1081.

Pedersen, T.P. (1991) 'Non-interactive and information-theoretic secure verifiable secret sharing', in Feigenbaum, J. (Ed.): *Advances in Cryptology – Crypto'91*, IACR, Springer-Verlag, University of California in Santa Barbara, 11–15 August, pp.129–140.

Shamir, A. (1979) 'How to share a secret', *Communications of the ACM*, Vol. 22, pp.612–613.

Stinson, D.R. (1994) 'Decomposition constructions for secret-sharing schemes', *IEEE Trans. Inform. Theory*, Vol. 40, pp.118–125.

Weisstein, E.W. (2003) *Pythagorean Triple – From Mathworld*, May, Website at http://mathworld.wolfram.com/PythagoreanTriple.html

Zou, X.K., Ramamurthy, B. and Magliveras, S. (2004) *Secure Group Communication over Data Networks*, Springer, October, ISBN: 0387229701.

Note

[1]Servers and shareholders are used interchangeably in this paper.