Dealer-Free Dynamic Secret Sharing Schemes

Mehrdad Nojoumian and Douglas R. Stinson

David R. Cheriton School of Computer Science
University of Waterloo, Waterloo, ON, N2L 3G1, Canada
{mnojoumi, dstinson}@uwaterloo.ca

Abstract. This research proposes dealer-free dynamic secret sharing constructions where different parameters of the scheme can be changed after the initialization. In practice, the ability of the adversary might be enhanced over time, for instance, by compromising more players. A possible solution to this problem is to increase the threshold or change the secret. In the literature, there exist some techniques to address this issue. These solutions assume the existence of a trusted authority, or have a large storage requirement, or are limited to a predefined modification. We apply some of the existing tools to tackle this problem. First, we formally analyze the well-known re-sharing techniques, also called 2-level sharing. After that, we present our solutions for dealer-free dynamic schemes in both the passive and active adversary models. In our constructions, players do not need to save extra shares beforehand, and the secret or threshold can be changed multiple times to any arbitrary values. Moreover, the presented protocols are unconditionally secure and realize a proactive secret sharing scheme.

Keywords: unconditional security, threshold changeability, secret changeability

1 Introduction

In *secret sharing schemes*, a secret is divided into different shares for distribution among several participants, and a subset of participants then collaborate to recover the secret [22, 3]. In particular, the (t, n)-*threshold secret sharing scheme* is proposed in which the secret is divided into n shares in such a way that any t players can combine their shares to reveal the secret, but any set of $t-1$ participants cannot learn anything about the secret.

We consider various types of adversaries in our constructions. In the *passive adversary* model, players follow protocols correctly but are curious to learn the secret. On the other hand, in the *active adversary* model, participants may deviate from protocols while trying to learn the secret. In addition, the passive or active adversary can be *static* or *mobile*. The former refers to the adversary who corrupts players ahead of time, while in the latter case, the adversary may corrupt different players at different stages of the protocols' executions. Finally, the entire security model can be *conditional* where the security of the protocols relies on computational assumptions such as the hardness of factoring, or *unconditional* where the adversary has unlimited computation power.

1.1 Motivation

Secret sharing schemes are an essential tool used in *secure multiparty computation* [27] where various participants cooperate in order to perform a computation task based on the private data they each provide. It is also applied in a variety of other applications such as joint signature and decryption [11], shared RSA keys [5], and electronic auctions with sealed bid [12]. In practice, the ability or computational power of the adversary might be enhanced over time, for instance, by compromising more players. In other words, changing the secret or threshold may be required throughout the lifetime of a secret. This problem can be tackled by changing these parameters in the absence of the *dealer* (i.e., the entity who initiates the scheme).

In the literature, there exist some solutions to address this issue. They assume the existence of a trusted authority, or have a large storage requirement, or are limited to a predefined modification. In addition, they may only modify the threshold at the side of the *combiner* (i.e., the entity who recovers the secret) rather than the entire scheme. In that case, the secret can be reconstructed if an adversary attacks a subset of players instead of the combiner. Our motivation is to apply existing methods in order to tackle these problems and add more utility to the secret sharing schemes.

1.2 Contribution

The contribution of this paper is to construct dynamic secret sharing schemes with a variety of desirable properties. Our protocols are *dealer-free* in which players change both the secret and the threshold based on a group agreement after the initialization. They are *unconditionally* secure in the sense that they do not rely on any computational assumptions. They have the minimum *storage cost* since players do not need to save extra shares beforehand in order to modify those parameters. They are *flexible* because the secret and the threshold can be changed to arbitrary values multiple times with the presence of enough participants.

First, we formally analyze the well-known re-sharing techniques. Our main contribution in this part is three theorems. We first show the security and correctness of the re-sharing method under the passive adversary model, and then illustrate two major drawbacks of the re-sharing approach in the active adversary setting. More precisely, we show that the 2-level sharing under the active adversary model is not secure against a mobile adversary. Moreover, each player has to store several shares unless participants agree on a set of good players.

Second, we propose a new multiplication-based solution for a dealer-free dynamic scheme in both passive and active adversary models. The proposed protocols change the secret as well as the threshold at the same time. In the first construction, we apply the enrollment protocol of [20] and the degree reduction and randomization method of [10] to change the secret and then adjust the threshold accordingly. In the second one, we develop a new protocol to generate a verified random symmetric polynomial to change the secret, and also extend the degree reduction and randomization approach presented in [10] to the symmetric bivariate polynomial for the threshold adjustment.

1.3 Organization

The paper is organized as follows. Section 2 reviews existing constructions for changing the threshold in the secret sharing schemes. Section 3 provides the required background for this paper. Section 4 formally analyzes the well-known re-sharing approach. Section 5 illustrates the proposed dealer-free dynamic secret sharing schemes. Finally, Section 6 outlines concluding remarks.

2 Related Work

K. M. Martin et al. [16] design a threshold changeable secret sharing scheme in the absence of secure channels based on two methods. The first one can be implemented by the Shamir approach and the second one is a geometrical construction. They have two strong assumptions. First, the original shares must contain the required information for extracting both the shares of the initial scheme and the shares of the future scheme, known as shares and subshares. Consequently, the size of the stored shares grows linearly with the number of required modifications to the threshold. Second, the proposed construction assumes that shareholders behave honestly in the sense that they only use the subshares that are relevant to the threshold in current use.

Using the prior approach, A. Maeda et al. [15] propose an unconditionally secure verifiable scheme where the threshold can be changed several times, say N, but only to the values determined in advance. In this construction, each player receives one full share and extracts the subsequent subshares from that by N public functions released by the dealer at the time of the initialization. The dealer also has to distribute N polynomials ahead of time. The authors assume that the secret is not recovered before the threshold changeability.

R. Steinfeld et al. [23] construct a threshold changeable mechanism for the standard Shamir secret sharing scheme. The general idea is that players add an appropriate amount of random noise to their shares in order to generate subshares which contain incomplete information regarding the primary shares. Consequently, t subshares are not sufficient to recover the secret; but, by a relatively large number of subshares, say t', the secret can be reconstructed.

C. Tartary and H. Wang [26] propose a dealer-free threshold changeable scheme in which the problem of secret recovery is reduced to the polynomial reconstruction problem. In this construction, players send some fake shares based on a mathematical assumption along with their real shares to increase the threshold t at the side of the combiner. First, the threshold stays constant among players. Second, their algorithm does not allow any value t' to be chosen. Third, the original threshold must be no larger than $\frac{n}{2}$. Finally, the scheme increases the asymptotic time complexity at the combiner side.

In addition to the problems we stated regarding the earlier techniques, there exists one common problem with all of these solutions. In fact, if an adversary attacks the shareholders (not the combiner) then he can have access to the original shares, shares related to various thresholds, or shares without any noise. Consequently, the secret can be recovered by the attacker.

Other techniques are proposed in the literature which can lead to the threshold changeability of a secret sharing scheme, for instance: re-sharing existing shares of a (t, n)-threshold scheme by a set of new polynomials of degree t' [8]; redistribution of secret shares to new access structures in which participants of a scheme send information to a new set of players in such a way that the old secret is shared among a new access structure [7, 17]; dynamic secret sharing schemes where the dealer triggers a specific access structure out of a given set, or enables the players to recover various secrets in different times by sending them the same broadcast message [4]. The authors in [1] also consider the scheme with changeable parameters, e.g., the threshold and the number of players, in order to minimize both the storage costs (size of shares) and the size of broadcast messages.

3 Preliminaries

In this section, we quickly review a secure multiparty computation technique and the Lagrange method for the polynomial interpolation.

3.1 Secure Multiplication of Secrets

Ben-Or et al. [2] proposed a method for the secure multiplication of two secrets. Suppose secrets α and β are encoded by two polynomials $f(x)$ and $g(x)$ of degree $t-1$, and each player P_i is holding one share on each of these polynomials, $f(i)$ and $g(i)$ respectively. The product of these two secrets $\alpha\beta$ is the constant term of the polynomial $h(x) = f(x) \times g(x)$.

If each player multiplies his shares together, the resulting value is a point on $h(x)$. There are two problems with this approach. First, the degree of $h(x)$ is $2t-2$ instead of $t-1$. Second, $h(x)$ is reducible as a product of two polynomials, which may not be secure. To overcome these problems, the authors in [2] first use a degree reduction protocol in which the polynomial $h(x)$ is truncated in the middle to decrease its degree to $t-1$. Let $k(x)$ be the truncation of $h(x)$. Subsequently, they apply a simple procedure to randomize the coefficients of $k(x)$, except the constant term which is the product of the two secrets.

Suppose $n \geq 2t-1$, where n is the number of players, $\mathcal{H}$ is the coefficient vector of $h(x)$, $\mathcal{S}$ is an evaluation vector of $h(x)$ at $2t-1$ points (i.e., players' shares on $h(x)$), $\mathcal{K}$ is the coefficient vector of $k(x)$, $\mathcal{R}$ is an evaluation vector of $k(x)$ at t points (i.e., players' shares on $k(x)$), $\mathcal{V'}_{n\times n}$ is the transpose of a Vandermonde matrix, and $\mathcal{P}_{n\times n}$ is a projection matrix (i.e., if $i=j$ and $i,j \leq t$ then $\mathcal{P}_{ij} = 1$, otherwise $\mathcal{P}_{ij} = 0$).

Before illustrating the degree reduction procedure, the coefficients of $h(x)$ are randomized as follows: each player P_i for $1 \leq i \leq n$ randomly selects a polynomial $q_i(x)$ of degree $2t-2$ with a zero constant term. After that, they each send $q_i(j)$ to P_j for $1 \leq j \leq n$. Finally, players add these values to their shares on $h(x)$:

$$h(x) = h(x) + \sum_{i=1}^{n}\sum_{j=1}^{n} q_i(j) \text{ and } h(0) = \alpha$$

$$\begin{array}{lrl}
 & \mathcal{H}\cdot\mathcal{V}'\cdot\mathcal{V}'^{-1} = \mathcal{H} & \text{(a)} \\
 & \mathcal{H}\cdot\mathcal{P} = \mathcal{K} & \text{(b)} \\
 & \mathcal{K}\cdot\mathcal{V}' = \mathcal{R} & \text{(c)} \\
 & \mathcal{H}\cdot\mathcal{V}' = \mathcal{S} & \text{(d)} \\
\text{(a),(b)} \Rightarrow & \mathcal{H}\cdot\mathcal{V}'\cdot\mathcal{V}'^{-1}\cdot\mathcal{P} = \mathcal{K} & \text{(e)} \\
\text{(c),(e)} \Rightarrow & \mathcal{H}\cdot\mathcal{V}'\cdot\mathcal{V}'^{-1}\cdot\mathcal{P}\cdot\mathcal{V}' = \mathcal{R} & \text{(f)} \\
\text{(d),(f)} \Rightarrow & \mathcal{S}\cdot\underbrace{\mathcal{V}'^{-1}\cdot\mathcal{P}\cdot\mathcal{V}'}_{\text{publicly known}} = \mathcal{R} &
\end{array}$$

The above computation shows that if we multiply the evaluation vector of a polynomial $h(x)$ in a publicly known matrix, we get the evaluation vector of $h(x)$'s truncation, denoted by $k(x)$, depending on the number of ones in the projection matrix's diagonal. Later on, this approach was simplified by R. Gennaro et al. [10], his approach is illustrated in the second phase of our constructions in Section 5. They combine the randomization and degree reduction stage by a simpler approach.

3.2 Lagrange Interpolation Formula

In this part, we illustrate the Lagrange method for the polynomial interpolation [24]. Suppose q is a prime number and $x_1, x_2, ..., x_t$ are distinct elements in $\mathbb{Z}_q$. In addition, suppose $f_1, f_2, ..., f_t$ are elements in $\mathbb{Z}_q$. Then, there is a unique polynomial $f(x) \in \mathbb{Z}_q[x]$ of degree at most $t-1$ such that $f(x_i) = f_i$ for $1 \leq i \leq t$:

$$f(x) = \sum_{i=1}^{t}\left(\prod_{1\leq j\leq t, i\neq j} \frac{x - x_j}{x_i - x_j} \times f_i\right) \tag{1}$$

In the case of bivariate polynomials, $y_1, y_2, ..., y_t$ are distinct elements in $\mathbb{Z}_q$ and $f_1(x), f_2(x), ..., f_t(x)$ are polynomials of degree at most $t-1$ in $\mathbb{Z}_q[x]$. Consequently, there is a unique polynomial $f(x, y) \in \mathbb{Z}_q[x, y]$ of degree at most $t-1$ such that $f(x, y_i) = f_i(x)$ for $1 \leq i \leq t$:

$$f(x,y) = \sum_{i=1}^{t} \left(\prod_{1 \leq j \leq t, i \neq j} \frac{y - y_j}{y_i - y_j} \times f_i(x) \right) \tag{2}$$

4 Changing the Threshold While Keeping the Same Secret

In this section, we review techniques for threshold changeability in the absence of a dealer. The general idea is to re-share existing shares of a (t, n)-threshold scheme by a set of polynomials of a higher degree t', i.e., converting a (t, n)-threshold scheme into a (t', n)-threshold scheme [8]. In the first model, there are only private channels between each pair of players, but in the second one, we also assume the existence of a synchronous broadcast channel. All computations are performed in a finite field $GF(q)$.

If the decision is to keep the same secret value (a desirable property in the case where the secret is difficult or expensive to change), then at least $n-t+1$ participants have to erase their old shares honestly. Erasing old shares is an inevitable assumption in a threshold changeable scheme with a constant secret [16], and even in proactive secret sharing schemes [21, 13]. Otherwise, the secret value itself must be changed (this issue is discussed in Section 5).

4.1 Passive Adversary Model

In the initial setting, we present a secret sharing scheme with threshold changeability under the passive adversary model.

Secret Sharing ($\mathcal{S}\boldsymbol{ha}$). Suppose, the dealer triggers a secret sharing protocol and then leaves the scheme. That is, he generates a univariate polynomial $f(x) \in \mathbb{Z}_p[x]$ of degree $t-1$ in which its constant term is the secret $f(0) = \alpha$, and then sends share $f(i)$ to player P_i for $1 \leq i \leq n$ [22].

Re-sharing Shares ($\mathcal{P}\boldsymbol{re}$). Now, suppose the participants decide to switch to a new threshold of $t' > t$ in the absence of the dealer.

1. Each player P_i generates a polynomial $g_i(x)$ of degree $t'-1$ such that its constant term is the player's share on $f(x)$:
$$g_i(0) = f(i) \tag{3}$$
2. Each player P_i sends $g_i(j)$ to player P_j for $1 \leq i, j \leq n$, i.e., re-sharing the original shares by auxiliary shares. The share-exchange matrix $\mathcal{E}_{n \times n}$, where each player generates a row and receives a column, is as follows:
$$\mathcal{E}_{n\times n} = \begin{pmatrix} g_1(1) & g_1(2) & \cdots & g_1(n) \\ g_2(1) & g_2(2) & \cdots & g_2(n) \\ \vdots & \vdots & \ddots & \vdots \\ g_n(1) & g_n(2) & \cdots & g_n(n) \end{pmatrix}$$

3. At this step, a set Δ is determined such that it contains the identifiers of t elected players, i.e., $|\Delta| = t$. Consequently, the following constants are publicly computed:

$$\gamma_i = \prod_{j \in \Delta, i \neq j} \frac{0-j}{i-j} \quad \text{where } 1 \leq i, j \leq n \text{ represent players' } ids \tag{4}$$

4. Each player P_j erases his old shares, and then combines the auxiliary shares he has received from other players to compute his new share as follows:

$$\varphi_j = \sum_{i \in \Delta} \Big(\gamma_i \times g_i(j) \Big) \tag{5}$$

Secret Recovery ($\mathcal{R}$*ec*). At this point, if at least t' players P_j cooperate, where $j \in \Delta'$ and $|\Delta'| \geq t'$, they can recover the secret by using the Lagrange interpolation method:

$$\alpha = \sum_{j \in \Delta'} \Big(\prod_{i \in \Delta', i \neq j} \frac{0-i}{j-i} \times \varphi_j \Big) \tag{6}$$

Theorem 1. *The re-sharing protocol $\mathcal{P}re$ for the threshold changeability is secure under the passive adversary model, and correctly computes secret α.*

Proof. Initially a set ∇ of colluders, where $|\nabla| = t-1$, are not able to recover the secret. In the next stage, players re-share their shares with $g_i(x)'s$ of degree $t'-1$ where $t' > t$. Consequently, colluders cannot reconstruct any of those re-sharing polynomials in order to reveal the good players' shares. Finally, all players erase their old shares to compute the new ones. Therefore, the protocol is secure under the passive (honest-but-curious) adversary model. Now, we show its correctness:

$$\begin{aligned}
\alpha &= \sum_{j \in \Delta'} \Big(\prod_{i \in \Delta', i \neq j} \frac{0-i}{j-i} \times \varphi_j \Big) && \text{by (6)} \\
&= \sum_{j \in \Delta'} \Big(\prod_{i \in \Delta', i \neq j} \frac{0-i}{j-i} \times \sum_{i \in \Delta} \Big(\gamma_i \times g_i(j) \Big) \Big) && \text{by (5)} \\
&= \sum_{j \in \Delta'} \Big(\prod_{i \in \Delta', i \neq j} \frac{0-i}{j-i} \times \sum_{i \in \Delta} \Big(\prod_{j \in \Delta, i \neq j} \frac{0-j}{i-j} \times g_i(j) \Big) \Big) && \text{by (4)} \\
&= \sum_{i \in \Delta} \Big(\prod_{j \in \Delta, i \neq j} \frac{0-j}{i-j} \times g_i(0) \Big) && \text{by (1)} \\
&= \sum_{i \in \Delta} \Big(\prod_{j \in \Delta, i \neq j} \frac{0-j}{i-j} \times f(i) \Big) && \text{by (3)} \\
&= f(0) && \text{by (1)}
\end{aligned}$$

□

4.2 Active Adversary Model

In this section, a modified version of the previous approach is illustrated which is also secure against an active adversary model. The authors in [19] present such a construction and use it in a different context [18], e.g., for creating a proactive secret sharing scheme. They also mention that the usual re-sharing technique is not secure against a mobile adversary. However, they do not provide any formal proof for this claim. We present a similar approach in order to change the threshold, and formally prove why it is not secure in the mobile adversary setting. For the sake of simplicity, details of accusation and defense procedures among players are removed.

Secret Sharing ($\mathcal{S}\boldsymbol{ha}$). Suppose, an honest dealer initiates a secret sharing scheme by using a symmetric bivariate polynomial. That is, he generates a polynomial $f(x, y) \in \mathbb{Z}_p[x, y]$ of degree $t-1$ in which its constant term is the secret:

$$f(x,y) = \sum_{i=0}^{t-1} \sum_{j=0}^{t-1} a_{ij} x^i y^j \text{ where } a_{00} = \alpha \text{ and } \forall i, j : a_{ij} = a_{ji}$$

The dealer then sends shares of players P_i for $1 \le i \le n$ accordingly, and leaves the scheme:

$$f_i(x) = f(x, \omega^i) \text{ where } \omega \text{ is a primitive element} \tag{7}$$

Definition 2. *In a secret sharing scheme, when the dealer uses a symmetric polynomial $f(x, y)$ to generate shares, players P_i are able to check the validity of their shares $f_i(x)$ by pairwise checks. The matrix representing those values is called pairwise check matrix, and is defined as follows:*

$$\mathcal{C}_{n\times n} = \begin{pmatrix} null & f_1(\omega^2) & \cdots & f_1(\omega^n) \\ f_2(\omega^1) & null & \cdots & f_2(\omega^n) \\ \vdots & \vdots & \ddots & \vdots \\ f_n(\omega^1) & f_n(\omega^2) & \cdots & null \end{pmatrix} \textit{ where } f_i(\omega^j) = f_j(\omega^i) \tag{8}$$

Since our focus is to construct a dealer-free protocol, we assume the dealer who initiates the scheme is honest because we would like all the distributed data to be verified shares up until the existence of the dealer. As a result of this assumption, the pairwise check matrix must be symmetric with respect to the main diagonal (top left to bottom right).

Re-sharing Shares ($\mathcal{A}\boldsymbol{re}$). Now, suppose the players decide to switch to a new threshold of $t' > t$ in the absence of the dealer.

1. Each player P_i generates a symmetric polynomial $g_i(x, y)$ of degree $t' - 1$ such that:

$$g_i(x, 0) = f_i(x) \tag{9}$$

   (a) Generate the symmetric bivariate polynomial $g'(x, y)$, where $a_{i0} = a_{0j} = 0$ for $0 \le i, j \le t'$.
   (b) Extend $f_i(x)$ to a symmetric bivariate polynomial $f'_i(x, y)$ by adding corresponding y-terms.
   (c) Finally, compute $g_i(x, y) = g'(x, y) + f'_i(x, y)$, which satisfies the condition $g_i(x, 0) = f_i(x)$.

2. Each player P_i sends $g_i(x, \omega^j)$ to player P_j for $1 \le i, j \le n$, i.e., re-sharing the original shares by auxiliary shares. The share-exchange matrix $\mathcal{E}_{n\times n}$, where each player generates a row and receives a column, is as follows:

$$\mathcal{E}_{n\times n} = \begin{pmatrix} g_1(x,\omega^1) & g_1(x,\omega^2) & \cdots & g_1(x,\omega^n) \\ g_2(x,\omega^1) & g_2(x,\omega^2) & \cdots & g_2(x,\omega^n) \\ \vdots & \vdots & \ddots & \vdots \\ g_n(x,\omega^1) & g_n(x,\omega^2) & \cdots & g_n(x,\omega^n) \end{pmatrix}$$

3. Players first perform n pairwise checks on $g_i(x, \omega^j)$ for $1 \leq i \leq n$, and then a single pairwise check on $g_i(0, \omega^j)$, to make sure that the constant terms of shares are consistent with the original shares distributed by the honest dealer:

$$\begin{pmatrix} null & g_1(0,\omega^2) & \cdots & g_1(0,\omega^n) \\ g_2(0,\omega^1) & null & \cdots & g_2(0,\omega^n) \\ \vdots & \vdots & \ddots & \vdots \\ g_n(0,\omega^1) & g_n(0,\omega^2) & \cdots & null \end{pmatrix} \overset{?}{=} \begin{pmatrix} null & f_1(\omega^2) & \cdots & f_1(\omega^n) \\ f_2(\omega^1) & null & \cdots & f_2(\omega^n) \\ \vdots & \vdots & \ddots & \vdots \\ f_n(\omega^1) & f_n(\omega^2) & \cdots & null \end{pmatrix}$$

4. At this step, a set Δ is determined such that it contains the identifiers of t good players, i.e., $|\Delta| = t$. Consequently, the following constants are publicly computed:

$$\gamma_i = \prod_{j \in \Delta, i \neq j} \frac{0 - \omega^j}{\omega^i - \omega^j} \text{ where } 1 \leq i, j \leq n \text{ represent players' } ids \tag{10}$$

5. Each player P_j erases his old shares, and then combines the auxiliary shares he has received from other players to compute his new share as follows:

$$\varphi_j(x) = \sum_{i \in \Delta} \Big(\gamma_i \times g_i(x, \omega^j) \Big) \tag{11}$$

Secret Recovery ($\boldsymbol{\mathcal{R}ec}$). At this point, if at least t' players P_j cooperate, where $j \in \Delta'$ and $|\Delta'| \geq t'$, they can recover the secret by using the Lagrange interpolation method:

$$\alpha = \sum_{j \in \Delta'} \left(\prod_{i \in \Delta', i \neq j} \frac{0 - \omega^i}{\omega^j - \omega^i} \times \varphi_j(x) \right) \tag{12}$$

Lemma 3. *In the active adversary model, after the initialization protocol $\mathcal{S}ha$, the secret α is reconstructed only by the constant terms of players' shares.*

Proof. We show that if at least t players P_j interpolate $f_j(0)'s$, the secret $\alpha = f(0,0)$ is revealed:

$$\begin{aligned} \alpha &= \sum_{j=1}^{t} \Big(\prod_{1 \leq i \leq t, i \neq j} \frac{0 - \omega^i}{\omega^j - \omega^i} \times f_j(0) \Big) && \text{by (1)} \\ &= \sum_{j=1}^{t} \Big(\prod_{1 \leq i \leq t, i \neq j} \frac{0 - \omega^i}{\omega^j - \omega^i} \times f(0, \omega^j) \Big) && \text{by (7)} \\ &= f(0,0) && \text{by (2)} \end{aligned}$$

□

Theorem 4. *The re-sharing protocol $\mathcal{A}re$ is not secure under the mobile adversary model, meaning that, the constant terms of players' shares $f_j(0)'s$ stay the same after the re-sharing protocol.*

Proof. If these constant terms stay the same, the mobile adversary can incrementally collect players' shares in different time periods in order to recover the secret. Therefore, we simply prove the following equality for all players $P_j; \forall j \in [1..n] \Rightarrow \varphi_j(0) = f_j(0)$:

$$\begin{aligned}
\varphi_j(0) &= \sum_{i\in\Delta}\left(\gamma_i \times g_i(0,\omega^j)\right) && \text{by (11)}\\
&= \sum_{i\in\Delta}\left(\prod_{j\in\Delta, i\neq j}\frac{0-\omega^j}{\omega^i-\omega^j}\times g_i(0,\omega^j)\right) && \text{by (10)}\\
&= \sum_{i\in\Delta}\left(\prod_{j\in\Delta, i\neq j}\frac{0-\omega^j}{\omega^i-\omega^j}\times g_i(\omega^j,0)\right) && \text{symmetry}\\
&= \sum_{i\in\Delta}\left(\prod_{j\in\Delta, i\neq j}\frac{0-\omega^j}{\omega^i-\omega^j}\times f_i(\omega^j)\right) && \text{by (9)}\\
&= \sum_{i\in\Delta}\left(\prod_{j\in\Delta, i\neq j}\frac{0-\omega^j}{\omega^i-\omega^j}\times f_j(\omega^i)\right) && \text{by (8)}\\
&= f_j(0) && \text{by (1)}
\end{aligned}$$

□

Theorem 5. *In the re-sharing protocols $\mathcal{A}re$ for the threshold changeability, each player has to store several shares unless participants predetermine (i.e., agree on) a set Δ of good players with $|\Delta| = t$.*

Proof. In the first case, we analyze the number of shares that each player has to store. Suppose $t < |\Delta| = m \leq n$, as shown in the protocol, i and j depend on the set Δ only in the last two steps. Therefore, the number of possible combinations of m values taken t at a time defines the number of possible sets of constants:

$$\binom{m}{t} = d \text{ then } \Phi = \{\{\gamma_{1,1}, \gamma_{1,2}, \ldots, \gamma_{1,t}\}, \ldots, \{\gamma_{d,1}, \gamma_{d,2}, \ldots, \gamma_{d,t}\}\} \text{ and } |\Phi| = d$$

Consequently, each player has to save d possible shares $\varphi_j(x)$'s according to each set of constants in Φ (storing extra shares may threaten the security of the scheme even more). Second, assume $t = |\Delta|$, then players are able to compute a single set of constants $\Phi = \{\{\gamma_1, \gamma_2, \ldots, \gamma_t\}\}$ because $\binom{t}{t} = |\Phi| = 1$. As a result, each player P_i updates his share to a single $\varphi_j(x)$. Finally, if $t > |\Delta|$, then players are not able to execute this protocol since it is the condition of the fourth step. It can be simply seen that determining the set Δ is crucial in the case of an active adversary model since $\forall i \in \Delta; P_i$ must be a good player. □

5 Changing Both the Threshold and the Secret

The goal of this section is to change both the threshold t and the secret α after the initialization, when the dealer no longer exists, i.e., to immunize the scheme against colluders by changing the main parameters of the scheme.

In our first solution, players initially generate a random polynomial $g(x)$ of degree $t-1$ with an arbitrary constant term β known to only t participants. After that, the original secret sharing polynomial $f(x)$ of degree $t-1$ is securely multiplied by $g(x)$. As a consequence, the new secret value of the scheme is $\alpha\beta$ with $2t-1$ as a new threshold. This requires the contribution of at least $2t-1$ participants. Finally, by using the degree reduction and randomization approach in [10] (the simplified version of [2]), the final threshold t' will be adjusted to something between $t < t' < 2t-1$.

The proposed constructions consist of n participants, $P_1, P_2, \dots, P_n$. In the first model, there are only private channels between each pair of players, but in the second one, we also assume the existence of a synchronous broadcast channel. Let $GF(q)$ be a finite field and let ω be a primitive element in this field. All computations are performed in the field $GF(q)$.

5.1 Passive Adversary Model

In this construction, first a new polynomial is generated and then the secret is changed by a secure multiplication. Finally, the threshold is adjusted while coefficients of the new secret sharing polynomial are randomized. Suppose the original secret sharing polynomial is $f(x)$ of degree $t-1$ with a constant term α, therefore, $f(i)$ is the share of the player P_i.

`Phase-1: Polynomial Production`

1. Based on a group agreement or a random selection, t players are chosen so that each generates a private random number for himself. In fact, these t random values associated with players' identifiers i form a polynomial $g(x)$ of degree $t-1$ with an arbitrary constant term β. Suppose the first t players are selected to generate $g(x)$, i.e., $1 \leq i \leq t$.
2. To generate shares on $g(x)$ for other $n-t$ participants, the *enrollment* protocol in [20] is used to create $g(k)$ for players P_k. Since $t+1 \leq k \leq n$, the following protocol is repeated $n-t$ times.

   (a) Each player P_i for $1 \leq i \leq t$ computes his corresponding Lagrange interpolation constant:

   $$\gamma_i = \prod_{1 \leq j \leq t, i \neq j} \frac{k-j}{i-j} \quad \text{where } i, j, k \text{ represent players' } ids \tag{13}$$

   (b) After that, each participant P_i multiplies his share by his Lagrange interpolation constant, and randomly splits the result into t portions, i.e., a row in a *share-exchange matrix*:

   $$g(i) \times \gamma_i = \partial_{1i} + \partial_{2i} + \cdots + \partial_{ti} \quad \text{for } 1 \leq i \leq t \tag{14}$$

   (c) Players exchange ∂_{ji}'s accordingly by pairwise channels. Then, each P_j holds t values, i.e., a column in a share-exchange matrix. P_j adds them together and sends the result to P_k:

   $$\sigma_j = \sum_{i=1}^{t} \partial_{ji} \quad \text{where } \partial_{ji} \text{ is the } j^{th} \text{ share-portion of the } i^{th} \text{ participant} \tag{15}$$

   (d) Finally, player P_k adds these values σ_j for $1 \leq j \leq t$ together to compute his share $g(k)$:

   $$g(k) = \sum_{j=1}^{t} \sigma_j \tag{16}$$

`Phase-2: Secure Multiplication`

At this stage, each participant P_i simply multiplies his two shares $f(i)$ and $g(i)$ together, and keeps the result, that is, a point on $h(x) = f(x) \times g(x)$ of degree $2t-2$ with $\alpha\beta$ as a new secret value. Players also erase all of the other values. An obvious solution for decreasing the threshold is to reveal some shares (for instance, players can call the enrollment protocol and run it at some arbitrary points $l \notin [1, n]$), but revealing shares forces players to save extra information along with their personal shares. This may threaten the security of the entire scheme even more. Therefore, a better solution is to reduce the degree of the secret sharing polynomial $h(x)$ while randomizing its coefficients [10].

`Phase-3: Threshold Adjustment`

1. Each player P_i generates a random polynomial $r_i(x)$ of degree $t' - 1$ with a constant term equal to his share, i.e., $r_i(0) = h(i)$, where t' is the new threshold based on the players' consensus. Then P_i gives $r_i(j)$ to P_j for $1 \le j \le n$, as a result, each player receives a vector of shares, i.e., a column in the share-exchange matrix:

$$\mathcal{E}_{n\times n} = \begin{pmatrix} r_1(1) & r_1(2) & \cdots & r_1(n) \\ r_2(1) & r_2(2) & \cdots & r_2(n) \\ \vdots & \vdots & \ddots & \vdots \\ r_n(1) & r_n(2) & \cdots & r_n(n) \end{pmatrix}$$

2. Participants then compute the first row of a publicly known matrix $\mathcal{V}^{-1}_{n\times n} \pmod q$ to adjust the threshold, where $\mathcal{V}_{n\times n}$ is the Vandermonde matrix, i.e., $\mathcal{V}_{i,j} = i^{(j-1)}$ for $1 \le i, j \le n$. Suppose this vector is $\mathcal{V}^{-1}_{1\times n} = (v_1 \quad v_2 \quad \cdots \quad v_n)$.
3. Eventually, each player P_j computes his final share by multiplying $\mathcal{V}^{-1}_{1\times n}$ by his vector of shares. In fact, $\widetilde{h}(x)$ is a polynomial of degree $t' - 1$ with the constant term $\alpha\beta$, and randomized coefficients compared to $h(x)$:

$$\widetilde{h}(j) = \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix} \cdot \begin{pmatrix} r_1(j) \\ r_2(j) \\ \vdots \\ r_n(j) \end{pmatrix}$$

To recover the secret, t' participants P_j have to collaborate in order to construct a polynomial of degree $t' - 1$, where its constant term is the new secret $\alpha\beta$:

$$\widetilde{h}(x) = \sum_{j=1}^{t'} \Bigg(\prod_{1\le i\le t', i\ne j} \frac{x-i}{j-i} \times \widetilde{h}(j) \Bigg) \quad \Rightarrow \quad \widetilde{h}(0) = \alpha\beta$$

Theorem 6. *The proposed protocol for the secret and threshold changeability is secure under the passive adversary model.*

Proof. The security of the enrollment protocol is illustrated in [20], and the security of the simplified multiplication is presented in [10].

5.2 Active Adversary Model

In the case of the active adversary model, the general idea is the same as the earlier protocol, i.e., multiplying the original polynomial $f(x, y)$ of degree $t-1$ with a constant term α, by a random polynomial $g(x, y)$ of degree $t-1$ with a constant term equal to β.

To generate $g(x, y)$, we develop a new protocol with some similarity to the initialization method in [25, 6]. In their construction, a dealer initiates a secret sharing scheme under the assumption that $t-1 \leq \lfloor \frac{n-1}{4} \rfloor$. Our construction is a dealer-free protocol for generating a verified random symmetric polynomial $g(x, y)$ under the assumption that $t-1 \leq \lfloor \frac{n-1}{3} \rfloor$, where $\xi = t-1$ is the number of colluders that the scheme can tolerate by using an error correction technique such as the Reed-Solomon code [14]. Suppose $f(x, y)$ is symmetric and $f(x, \omega^i)$ is the share of each player P_i in the initial setting.

`Phase-1: Polynomial Production`

1. To initiate $g(x, y)$, t players P_i are chosen based on a group agreement or a random selection. For the sake of simplicity, suppose the first t players are selected, i.e., $1 \leq i \leq t$. Each player generates a private random number g_{ii} for himself. Subsequently, each pair of players P_i and P_j agree on a common value $g_{ij} = g_{ji}$ through private channels:

$$\mathcal{C}_{t\times t} = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1t} \\ g_{21} & g_{22} & \cdots & g_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ g_{t1} & g_{t2} & \cdots & g_{tt} \end{pmatrix}$$

2. Each player P_k for $1 \leq k \leq t$ computes his new share $g_k(x)$ by the Lagrange method as follows:

$$g_k(x) = \sum_{i=1}^{t} \Big(\prod_{1\leq j\leq t, i\neq j} \frac{x-\omega^j}{\omega^i-\omega^j} \times g_{ki} \Big)$$

   In fact, shares $g_k(x)$'s associated with players' identifiers form a symmetric bivariate polynomial $g(x, y)$ of degree $t-1$ with an arbitrary constant term β:

$$g(x, y) = \sum_{k=1}^{t} \Big(\prod_{1\leq j\leq t, k\neq j} \frac{y-\omega^j}{\omega^k-\omega^j} \times g_k(x) \Big)$$

3. To generate shares on $g(x, y)$ for other $n-t$ players P_j, where $t+1 \leq j \leq n$, the following sub-protocol is repeated $n-t$ times:

   (a) Each player P_i for $1 \leq i \leq t$ sends $g_i(\omega^j)$ to P_j to help him create his share $g_j(x)$.
   (b) After that, player P_j computes $g_j(x)$ through the interpolation of pairs $(\omega^i, g_i(\omega^j))$:

$$g_j(x) = \sum_{i=1}^{t} \Big(\prod_{1\leq j\leq t, i\neq j} \frac{x-\omega^j}{\omega^i-\omega^j} \times g_i(\omega^j) \Big)$$

4. Each pair of players P_i and P_j perform the pairwise checks $g_i(\omega^j) \overset{?}{=} g_j(\omega^i)$ through secure channels. Consequently, if a player P_i finds that the above equation does not hold while checking it with P_j, he then broadcasts (i, j), meaning that P_i is accusing P_j.
5. Each player P_i computes a subset $\mathit{\Gamma} \subseteq \{1, ..., n\}$ such that any ordered pair $(i, j) \in \mathit{\Gamma} \times \mathit{\Gamma}$ has not been broadcasted. In fact, $\mathit{\Gamma}$ is a *clique*, and its construction is NP-complete. The authors in [9] resolve this issue by the *maximal matching* problem which has a polynomial time solution.
6. If $|\mathit{\Gamma}| \geq n-\xi$, then P_i outputs $ver_i = 1$, otherwise, P_i outputs $ver_i = 0$. Consequently, if at least $n - \xi$ players output $ver_i = 1$, $g(x, y)$ is accepted as a new symmetric polynomial and players proceed to the next step. Otherwise, another set of t players is selected to initiate $g(x, y)$.

At the end of the phase-1, all good players belonging to $\mathit{\Gamma}$ have consistent shares with respect to an arbitrary secret value β.

`Phase-2: Secure Multiplication`

In this step, each participant P_i simply multiplies his two shares $f(x, \omega^i)$ and $g(x, \omega^i)$ together, and keeps the result, that is, a point on the symmetric $h(x, y) = f(x, y) \times g(x, y)$ of degree $2t - 2$ with $\alpha\beta$ as a new secret value. Honest players also erase all the other values. At this point, the secret is encoded by a polynomial of degree $2t - 2$. To adjust the threshold, we use the extended version of the degree reduction and randomization method in [10].

`Phase-3: Threshold Adjustment`

1. Each player P_i generates a random symmetric polynomial $r_i(x, y)$ of degree $t' - 1$ (the new threshold based on the players' consensus) such that $r_i(x, 0) = h'(x, \omega^i)$. This is similar to the first step of the protocol $\mathcal{A}re$, where $h'(x, \omega^i)$ is the truncation of $h(x, \omega^i)$, that is, terms with the degree of less than or equal to t'. Then, player P_i sends $r_i(x, \omega^j)$ to P_j for $1 \leq j \leq n$:

$$\mathcal{E}_{n\times n} = \begin{pmatrix} r_1(x,\omega^1) & r_1(x,\omega^2) & \cdots & r_1(x,\omega^n) \\ r_2(x,\omega^1) & r_2(x,\omega^2) & \cdots & r_2(x,\omega^n) \\ \vdots & \vdots & \ddots & \vdots \\ r_n(x,\omega^1) & r_n(x,\omega^2) & \cdots & r_n(x,\omega^n) \end{pmatrix}$$

2. Participants then compute the first row of a publicly known matrix $\mathcal{V}^{-1}_{n\times n} \pmod{q}$ to adjust the threshold, where $\mathcal{V}_{n\times n}$ is the Vandermonde matrix for $[\omega^1, \omega^2, \cdots, \omega^n]$, i.e., $\mathcal{V}_{i,j} = (\omega^i)^{(j-1)}$ for $1 \leq i, j \leq n$. Suppose this vector is $\mathcal{V}^{-1}_{1\times n} = (v_1 \quad v_2 \quad \cdots \quad v_n)$.
3. Eventually, each player P_j computes his final share by multiplying $\mathcal{V}^{-1}_{1\times n}$ by his vector of shares. In fact, $\widetilde{h}(x, y)$ is a symmetric polynomial of degree $t' - 1$ with the constant term $\alpha\beta$, and randomized coefficients compared to $h(x, y)$:

$$\widetilde{h}(x,\omega^j) = \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix} \cdot \begin{pmatrix} r_1(x,\omega^j) \\ r_2(x,\omega^j) \\ \vdots \\ r_n(x,\omega^j) \end{pmatrix}$$

To recover the secret, t' players P_j have to collaborate in order to construct a bivariate polynomial of degree $t'-1$, where its constant term is the secret $\alpha\beta$:

$$\widetilde{h}(x,y) = \sum_{j=1}^{t'} \Bigg(\prod_{1 \leq i \leq t', i \neq j} \frac{y - \omega^i}{\omega^j - \omega^i} \times \widetilde{h}(x, \omega^j) \Bigg) \quad \Rightarrow \quad \widetilde{h}(0,0) = \alpha\beta$$

Theorem 7. *The proposed protocol for the secret and threshold changeability is secure under the active adversary model, where $t-1 \leq \left\lfloor \frac{n-1}{3} \right\rfloor$ and $\xi = t-1$ is the degree of the secret sharing polynomials.*

Proof. Since all polynomials remain symmetric during different stages, players can perform the pairwise checks through secure channels at any time in order to detect malicious players who deviate from protocols (similar to the first phase). This is even simpler than the approach in Appendix B of [10]. For the sake of simplicity, suppose $3|(n-1)$, then we have:

$$\begin{aligned} t-1 &\leq (n-1)/3 \\ 3(t-1) &\leq n-1 \\ 3\xi &\leq n-1 \\ 3\xi+1 &\leq n \\ 3\xi &< n \end{aligned}$$

There exist n players for ξ possible faulty shares where $3\xi < n$. That is, 2ξ redundancy in the codewords. Therefore, by using the Reed-Solomon error correction technique [14], we can correct $2\xi/2 = \xi$ faulty shares and interpolate a unique polynomial of degree $t-1$ at any stages. □

6 Conclusion

We constructed a new dealer-free dynamic scheme in the unconditionally secure setting by using existing techniques and developing new protocols. In our protocols, participants do not need to save extra shares ahead of time, and both the secret and the threshold can be modified to any arbitrary values multiple times.

Our constructions are dealer-free, unconditional, and are secure in the active adversary models. In fact, it is quite challenging to design protocols in this setting. In other words, if one relaxes any of these assumptions, then he can decrease the computation and communication complexities. For instance, by using a trusted authority, or constructing the proposed scheme by relying on computational assumptions, or considering the simple passive adversary model.

References

[1] S. G. Barwick, W. A. Jackson, and K. M. Martin, *Updating the parameters of a threshold scheme by minimal broadcast*, IEEE Transactions on Information Theory, 51 (2005), pp. 620–633.

[2] M. Ben-Or, S. Goldwasser, and A. Wigderson, *Completeness theorems for non-cryptographic fault-tolerant distributed computation*, in STOC, ACM, 1988, pp. 1–10.

[3] G. R. Blakley, *Safeguarding cryptographic keys*, in National Computer Conference, New York, R. E. Merwin, J. T. Zanca, and M. Smith, eds., vol. 48 of AFIPS Conference proceedings, Montvale, NJ, USA, 1979, AFIPS Press, pp. 313–317.

[4] C. Blundo, A. Cresti, A. D. Santis, and U. Vaccaro, *Fully dynamic secret sharing schemes*, Theoretical Computer Science, 165 (1996), pp. 407–440.

[5] D. Boneh and M. Franklin, *Efficient generation of shared rsa keys*, Journal of ACM, 48 (2001), pp. 702–722.

[6] P. D'Arco and D. R. Stinson, *On unconditionally secure robust distributed key distribution centers*, in Advances in Cryptology, Proceedings of ASIACRYPT '02, Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 346–363.

[7] Y. Desmedt and S. Jajodia, *Redistributing secret shares to new access structures and its applications*, in Technical Report ISSE TR-97-01, George Mason University, 1997.

[8] Y. Frankel, P. Gemmell, P. D. MacKenzie, and M. Yung, *Optimal-resilience proactive public-key cryptosystems*, in FOCS '97: Proceedings of the 38th Annual Symposium on Foundations of Computer Science, Washington, DC, USA, 1997, IEEE Computer Society, p. 384.

[9] R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin, *The round complexity of verifiable secret sharing and secure multicast*, in STOC, 2001, pp. 580–589.

[10] R. Gennaro, M. O. Rabin, and T. Rabin, *Simplified vss and fast-track multiparty computations with applications to threshold cryptography*, in PODC '98: Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing, New York, NY, USA, 1998, ACM, pp. 101–111.

[11] S. Goldwasser, *Multi party computations: past and present*, in PODC '97: Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing, ACM, 1997, pp. 1–6.

[12] M. Harkavy, J. D. Tygar, and H. Kikuchi, *Electronic auctions with private bids*, in WOEC'98: Proceedings of the 3rd Conference on USENIX Workshop on Electronic Commerce, USENIX Association, 1998, pp. 6–6.

[13] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, *Proactive secret sharing or: How to cope with perpetual leakage*, in CRYPTO, D. Coppersmith, ed., vol. 963 of LNCS, Springer-Verlag, 1995, pp. 339–352.

[14] F. MacWilliams and N. Sloane, *The theory of error-correcting codes*, North-Holland Amsterdam, 1978.

[15] A. Maeda, A. Miyaji, and M. Tada, *Efficient and unconditionally secure verifiable threshold changeable scheme*, in ACISP '01: Proceedings of the 6th Australasian Conference on Information Security and Privacy, London, UK, 2001, Springer-Verlag, pp. 403–416.

[16] K. M. Martin, J. Pieprzyk, R. Safavi-Naini, and H. Wang, *Changing thresholds in the absence of secure channels*, in ACISP, J. Pieprzyk, R. Safavi-Naini, and J. Seberry, eds., vol. 1587 of LNCS, Springer, 1999, pp. 177–191.

[17] K. M. Martin, R. Safavi-Naini, and H. Wang, *Bounds and techniques for efficient redistribution of secret shares to new access structures*, The Computer Journal, 42 (1999), pp. 638–649.

[18] V. Nikov and S. Nikova, *On proactive secret sharing schemes*, in Selected Areas in Cryptography, H. Handschuh and M. A. Hasan, eds., vol. 3357 of Lecture Notes in Computer Science, Springer, 2004, pp. 308–325.

[19] V. Nikov, S. Nikova, and B. Preneel, *Multi-party computation from any linear secret sharing scheme unconditionally secure against adaptive adversary: The zero-error case*, in ACNS, J. Zhou, M. Yung, and Y. Han, eds., vol. 2846 of Lecture Notes in Computer Science, Springer, 2003, pp. 1–15.

[20] M. Nojoumian, D. R. Stinson, and M. Grainger, *An unconditionally secure social secret sharing scheme.* Cryptology ePrint Archive, Report 2009/207, 2009. http://eprint.iacr.org/.

[21] R. Ostrovsky and M. Yung, *How to withstand mobile virus attacks*, in PODC '91: Proceedings of the 10th Annual ACM Symposium on Principles of Distributed Computing, ACM, 1991, pp. 51–59.

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

[23] R. Steinfeld, H. Wang, and J. Pieprzyk, *Lattice-based threshold-changeability for standard shamir secret-sharing schemes*, in ASIACRYPT, P. J. Lee, ed., vol. 3329 of LNCS, Springer, 2004, pp. 170–186.

[24] D. R. Stinson, *Cryptography: Theory and Practice,Third Edition*, CRC Press, 2005.

[25] D. R. Stinson and R. Wei, *Unconditionally secure proactive secret sharing scheme with combinatorial structures*, in Selected Areas in Cryptography, H. M. Heys and C. M. Adams, eds., vol. 1758 of Lecture Notes in Computer Science, Springer, 1999, pp. 200–214.

[26] C. Tartary and H. Wang, *Dynamic threshold and cheater resistance for shamir secret sharing scheme*, in Inscrypt, H. Lipmaa, M. Yung, and D. Lin, eds., vol. 4318 of LNCS, Springer, 2006, pp. 103–117.

[27] A. C. Yao, *Protocols for secure computations*, in SFCS '82: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, Washington, DC, USA, 1982, IEEE Computer Society, pp. 160–164.