Dealer-Free Dynamic Secret Sharing Schemes with Unconditional Security

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 article proposes dealer-free dynamic secret sharing schemes where parameters of the protocol, such as the threshold and the secret, can be changed after the initialization. Our motivation is to tackle the following two problems: (a) 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 and/or change the secret. (b) On the other hand, another common problem with almost all secret sharing schemes is that they are one-time meaning that after secret recovery both the secret and shares are known to everyone. This problem might be resolved if the dealer shares several secrets at the beginning but a better solution for players is to dynamically generate new secrets in the absence of the dealer. As our contribution, we first propose a new scheme, called *incremental multilevel secret sharing*, in order to motivate secret changeability. We then formally analyze the well-known re-sharing technique for threshold changeability and show its drawbacks. Finally, we present our solutions for dealer-free dynamic schemes in both passive and active adversary models. In our constructions, players do not need to save extra shares beforehand, and both the threshold and the secret can be changed multiple times to arbitrary values. In the proposed schemes, each secret is changed based on the linear combination of previous secrets. As a result, we are able to recover old secrets at any time.

Keywords: threshold changeability, secret changeability

1 Introduction

In a *secret sharing scheme*, a secret is divided into different shares for distribution among several players, and a subset of players 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 $t-1$ parties cannot learn the secret. Our goal is to construct a *dynamic secret sharing scheme* where parameters of the protocol are changed after the initialization. We would like to change the threshold and the secret simultaneously in the absence of the dealer.

We consider various types of adversaries in our constructions. In the *passive adversary* setting, players follow protocols correctly but are curious to learn the

[*] Research supported by NSERC Canada Graduate Scholarship
[**] Research supported by NSERC Discovery Grant 203114-06

secret. On the other hand, in the *active adversary* model, players may deviate from protocols (e.g., prevent the secret recovery or reconstruct an incorrect secret) 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 players at different stages of the protocol's execution. Finally, the entire security model can be *computational*, where the security of the protocols relies on computational assumptions such as the hardness of factoring, or *unconditional*, in which the adversary has unlimited computational power.

1.1 Motivation

Secret sharing is an essential tool used in many cryptographic constructions such as *secure multiparty computation* [30] where various players cooperate to perform a computational task based on the private data they each provide. In this article, our motivation is to tackle the following two major problems:

1. In practice, the ability or the computational power of the adversary might be enhanced over time, for instance, by compromising more participants. In other words, increasing the threshold and/or changing the secret might be required throughout the lifetime of a secret. This problem can be resolved by changing these parameters in the absence of the *dealer* (i.e., the entity who initiates the scheme). In the literature, there exist some techniques to address this issue but these solutions suffer from either of the following drawbacks, they:

   - assume the existence of a trusted authority.
   - have a large storage requirement.
   - are limited to predefined modifications.
   - rely on computational assumptions.

   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.
2. Another well-known problem with almost all secret sharing schemes is that they are one-time meaning that after secret recovery both the secret and shares are known to everyone. To resolve this issue, the concept of *multistage secret sharing* is proposed [10], where many independent secrets are shared by the dealer ahead of time, but a better solution for players is to dynamically generate new secrets in the absence of the dealer.

Our motivation is therefore to apply existing tools and develop new techniques in order to tackle these problems and add more utility to secret sharing schemes. We construct new protocols with the ability of (a) threshold increase as well as (b) secret changeability based on the linear combination of previous secrets.

We motivate our constructions by two applications: *incremental multilevel secret sharing* and *sealed-bid auctions*. It worth mentioning that in the context of the secret sharing, a dynamic scheme refers to a protocol with threshold and/or access structure changeability. To the best of our knowledge, all those constructions update the threshold without changing the secret.

1.2 Our Contributions

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

1. First of all, we motivate secret and threshold changeability by constructing a new scheme, called *incremental multilevel secret sharing.* To further motivate our dynamic scheme, we illustrate a sealed-bid auction protocol in which secrets, i.e., bids, are changed in order to determine the winner in a secure fashion.
2. We then analyze the well-known re-sharing technique, also known as 2-level sharing. We initially show the security and correctness of the re-sharing method under the passive adversary model and then illustrate two major drawbacks of this approach in the active adversary setting. More precisely:
   - We formally 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 parties agree on a set of exactly t good players (t is the initial threshold).
3. Finally, we propose a solution for a dealer-free dynamic secret sharing scheme in both passive and active adversary models. To change the secret, we generate a random symmetric polynomial in the absence of the dealer by a new protocol, called *dealer-free verified polynomial production.* To adjust the threshold, we extend the degree reduction and randomization approach of [9] to the verified symmetric bivariate polynomials.

1.3 Organization

The paper is organized as follows. Section 2 reviews existing protocols for changing the threshold in the secret sharing schemes. Section 3 provides some preliminaries. Section 4 illustrates two applications of secret and threshold changeability. Section 5 formally analyzes the well-known re-sharing approach. Section 6 demonstrates our dealer-free dynamic secret sharing, and Section 7 contains concluding remarks.

2 Previous Work

Martin et al. [14] 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 geometric construction. They make two 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, Maeda et al. [13] 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 protocol, each player receives one full share and extracts the subsequent subshares from it 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 changes, therefore, no share has been pooled.

Steinfeld et al. [24] construct a threshold changeability mechanism for the Shamir secret sharing scheme. The general idea is that players add an appropriate amount of random noise to their shares in order to create subshares that contain incomplete information regarding the primary shares. As a result, t subshares are not sufficient to recover the secret, but by using a relatively large number of subshares, say t', the secret can be reconstructed.

Tartary and Wang [28] 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 along with their real shares to increase the threshold t at the side of the combiner to a new value t'. First, the threshold stays constant among players. Second, their algorithm does not allow any value t' to be chosen.

In addition to the drawbacks we mentioned regarding the existing techniques, there is one common problem with all of these solutions. That is, if an adversary attacks the shareholders (not the combiner) then he can have access to the (a) original shares, (b) shares related to various thresholds, or (c) shares without any noise. Consequently, the secret can be recovered by the attacker.

Other techniques are proposed in the literature for threshold changeability in 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' [7]; 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 [6, 15]; 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 paper [1] also considers schemes 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

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 holds 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 the desired $t-1$. Second, $h(x)$ is reducible as a product of two polynomials, which may not be secure. To overcome these problems, [2] uses 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 resulting 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.

Later on, this method was simplified by Gennaro et al. [9], his approach is illustrated in the second phase of our constructions in Section 6. They combine the randomization and degree reduction stage by a simpler approach.

3.2 Lagrange Interpolation Formula

In this part, we recall the Lagrange method for the polynomial interpolation [25]. Let q be a prime number. Let $x_1, x_2, ..., x_t$ and $f_1, f_2, ..., f_t$ be distinct 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 $1 \leq i \leq t$:

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

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 $1 \leq i \leq t$:

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

4 Security Applications

As we stated earlier, dealer-free threshold increase along with secret changeability (based on the linear combination of previous secrets) is our new approach motivated by the following applications. For the sake of simplicity, we only consider the passive adversary model in this section.

4.1 Incremental Multilevel Secret Sharing

In this new approach, players are able to progressively construct a multilevel secret sharing scheme with several secrets in the absence of the dealer, that is, players change both the access structure and the threshold while generating multiple secrets. Simmons [23] first proposed a *disjunctive multilevel secret sharing*, and then Tassa [29] extended that construction to a *conjunctive multilevel secret sharing.* They both use a single secret to construct those hierarchical threshold secret sharing schemes. We first state the following definitions and then show our new secret sharing scheme.

Definition 1. *An access structure Γ defines different authorized subsets of users and satisfies two conditions: (a) if $A \in \Gamma$ and $A \subseteq B$ then $B \in \Gamma$, i.e., monotonicity, and (b) if $A \in \Gamma$ then $|A| > 0$. In a threshold access structure, $|A| \geq t$ where $t-1$ is the degree of the secret sharing polynomial.*

Definition 2. *An incremental multilevel secret sharing scheme is a construction where several secrets $\alpha_1 \cdots \alpha_i \cdots \alpha_l$ are shared among players with monotonically increasing thresholds $t_1 < \cdots < t_i < \cdots < t_l$. Let $\mathcal{P}$ be a set of n players and assume $\mathcal{P}$ is composed of different levels where $\mathcal{P}_i$-s show disjoint subsets of players.*

$$\mathcal{P} = \bigcup_{i=1}^{l} \mathcal{P}_i \textit{ where } \mathcal{P}_i \cap \mathcal{P}_j = \emptyset \textit{ for all } 1 \leq i \neq j \leq l$$

Then α_k is recovered if players $\bigcup_{i=k}^{l} \mathcal{P}_i$, where $|\mathcal{P}_i| \geq t_i$, cooperate and first recover their secrets sequentially, i.e., from the highest level l to the level k.

Secret Sharing ($\mathcal{S}ha$)

1. Suppose, a dealer distributes shares of an initial secret α_1 with a polynomial of degree $t_0 - 1$ among players $\mathcal{P} = \{P_1, \cdots, P_n\}$, and then leaves the scheme.
2. Subsequently, players perform the following steps for $1 \leq i \leq l-1$ to construct an incremental l-level secret sharing:

   (a) Players $\mathcal{P}$ generate a random polynomial of degree $t_i - 1$ (that is, t_i is the threshold) with an unknown constant term β_i where $t_{i-1} < t_i$.
   (b) They compute shares of $\alpha_{i+1} = \alpha_i \beta_i$ where $|\mathcal{P}| \geq t_0 + t_i - 1$, and adjust its threshold t_k to be $2 \leq t_k \leq t_0 + t_i - 1$. They finally erase shares of α_i.
   (c) A subset of players, say $\mathcal{P}_i \subset \mathcal{P}$ where $|\mathcal{P}_i| \geq t_i$, only keep shares of β_i and the rest of players, i.e., $\mathcal{P} - \mathcal{P}_i$, only keep shares of α_{i+1}.
   (d) Finally, $\mathcal{P} = \mathcal{P} - \mathcal{P}_i$ and $t_0 = t_k$. Players in this new set $\mathcal{P}$ go to (a) to construct another level of secret sharing.

Secret Recovery ($\mathcal{R}ec$)

1. Players first cooperate to recover α_l and $\beta_{l-1} \cdots \beta_i \cdots \beta_1$. They may recover these secrets up until to a specific level i.
2. After that, they solve the following system of linear congruence equations: $\alpha_{i+1} \overset{q}{\equiv} \alpha_i \beta_i$ for $i = l-1$ to $i = 1$. Therefore, $\alpha_l \cdots \alpha_i \cdots \alpha_1$ are recovered.

Since q is a prime number, each congruence equation has a unique solution for α_i. We later show how to generate shares of β_i, compute shares of $\alpha_{i+1} = \alpha_i \beta_i$, and adjust its threshold through a dynamic secret sharing. It is worth mentioning that our approach can also be implemented by the addition operation, i.e., $\alpha_{i+1} = \alpha_i + \beta_i$, where the polynomial encoding β_i has a higher degree. In this case, we need to randomize the coefficients of the polynomial encoding α_{i+1}.

Example 1. Consider the following incremental 3-level secret sharing. If participants cooperate to recover α_3, β_2 and β_1, they can then solve the following system of linear congruence equations: $\alpha_3 \overset{q}{\equiv} \alpha_2 \beta_2, \alpha_2 \overset{q}{\equiv} \alpha_1 \beta_1$. As a result, α_2 and α_1 are recovered.

$$\alpha_1 : \mathcal{P} = \{P_1, \cdots, P_{13}\}^{t_0=2} \text{ and } \beta_1 : \mathcal{P}_1 = \{P_1, P_2, P_3\}^{t_1=3}$$
$$\alpha_2 : \mathcal{P} = \{P_4, \cdots, P_{13}\}^{t_0=3} \text{ and } \beta_2 : \mathcal{P}_2 = \{P_4, P_5, P_6, P_7\}^{t_2=4}$$
$$\alpha_3 : \mathcal{P}_3 = \{P_8, P_9, P_{10}, P_{11}, P_{12}, P_{13}\}^{t_3=6}$$

Players from $\mathcal{P}_1$ are only able to recover the original secret α_1, if the larger group of players from $\mathcal{P}_3$ and consequently from $\mathcal{P}_2$ first cooperate to reveal α_3 and β_2, i.e., balance of power. You can imagine the president and vice president, ministers, and senators accordingly as the realization of this hierarchical authority. Any decision on troops by the president and vise president is subject to the confirmations of ministers and senators. On the other hand, even by having those confirmations, it is the president office that makes the final decision.

To explain how our *incremental multilevel secret sharing* differs from existing protocols, consider the above hierarchy along with those thresholds, as shown in Example 1. In the case of the *disjunctive multilevel secret sharing*, players from $\mathcal{P}_1$ are enough to recover the secret without the contributions of other players, i.e., cooperations of all levels are not required. In the case of the *conjunctive multilevel secret sharing*, all players from $\mathcal{P}_1$, and one player from $\mathcal{P}_2$, and two players from $\mathcal{P}_3$ are enough to recover the secret. In both cases, we only have a single secret.

4.2 Sealed-Bid Auctions

We demonstrate another application of secret changeability by a simple example, in which we can also increase the threshold if it is being required. Suppose in a secure auction protocol, two bidders distribute shares of their bids (α_1 and α_2) among auctioneers. Assume auctioneers first receive shares of a random polynomial with an unknown constant term β (known as the multiplicative factor) from a trusted initializer, and then receive shares of two random polynomials with unknown constant terms δ_1 and δ_2 (known as additive factors), where $1 \leq \delta_1 \neq \delta_2 < \beta$.

Subsequently, they compute shares of $\alpha'_1 = \alpha_1\beta + \delta_1$ and $\alpha'_2 = \alpha_2\beta + \delta_2$, and recover α'_1 and α'_2 in the absence of the bidders. As a result, they can define the winner who proposed the higher valuation without revealing the actual bids or the relative difference between them. (To prevent the modular reduction, assume $q > \beta(\kappa + 1)$, where κ denotes the maximum possible price and q shows the size of the finite filed.) This method are used in the proposed constructions of [20].

There exist other protocols in the literature using a similar masking approach. For instance, the authors in [18, 27] only use the additive factor in their masking method. In the first paper, the authors apply a bitwise technique in the passive adversary model to generate shares of a mask. In the second article, the authors rely on the mask publishers (i.e., trusted parties) to generate shares of the masks in a secure combinatorial auction protocol; in this construction, the secret is encoded as a degree of a secret sharing polynomial.

5 Formal Analysis of an Existing Approach

In this section, we review a well-known technique for threshold changeability in the absence of the dealer (as we mentioned earlier, in all the existing dynamic schemes, only the threshold is changed and the secret remains the same). 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 [7]. In the passive adversary model, there are only private channels between each pair of players, but in the active adversary setting, a synchronous broadcast channel are also considered. All computations are performed in the finite field $\mathbb{Z}_q$.

If the decision is to keep the same secret (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 [14], and even in proactive secret sharing schemes [21, 11]. Otherwise, the secret itself must be changed (this issue is discussed in Section 6).

5.1 Passive Adversary Model

In the initial setting, we present a secret sharing scheme with threshold changeability under the passive adversary model [7].

Secret Sharing ($\mathcal{S}ha$). Suppose, the dealer initiates a secret sharing protocol and then leaves the scheme. That is, he randomly generates a polynomial $f(x) \in \mathbb{Z}_q[x]$ 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 ($\mathcal{P}re$). 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 randomly 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. 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 ($\boldsymbol{\mathcal{R}ec}$). Now, if at least t' players P_j cooperate, where $j \in \Delta'$ and $|\Delta'| \geq t'$, they can recover α by using the Lagrange interpolation method:

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

Theorem 1. *The re-sharing protocol $\mathcal{P}re$ for 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:

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

□

Example 2. Suppose the dealer distributes shares of $f(x) = 3+2x+x^2 \in \mathbb{Z}_{19}$, where $t = 3$, among four players as follows: $f(1) = 6, f(2) = 11, f(3) = 18, f(4) = 8$. The re-sharing phase has four steps as follows:

1. Players re-share their shares with new polynomials of degree three, i.e., $t' = 4$.

$$f_1(x) = \mathbf{6} + x + x^2 + 2x^3 \qquad f_3(x) = \mathbf{18} + 3x + 2x^2 + x^3$$
$$f_2(x) = \mathbf{11} + 2x + x^2 + 3x^3 \qquad f_4(x) = \mathbf{8} + 2x + 2x^2 + 2x^3$$

2. The $\mathcal{E}_{n \times n}$, where each P_i generates a row and receives a column, is as follows:

$$\mathcal{E}_{n \times n} = \begin{pmatrix} 10 & 9 & 15 & 2 \\ 17 & 5 & 12 & 18 \\ 5 & 2 & 15 & 12 \\ 14 & 17 & 10 & 5 \end{pmatrix}$$

3. At this stage, each player P_i has to store four shares or players need to define a set Δ (in the active adversary setting, this set only contains the identifiers of t good participants) in order to convert these shares to a single share. Suppose $\Delta = \{P_1, P_2, P_3\}$, we therefore have:

$$\gamma_1 = \frac{(0-2)(0-3)}{(1-2)(1-3)} = 3 \qquad \gamma_2 = \frac{(0-1)(0-3)}{(2-1)(2-3)} = -3 \qquad \gamma_3 = \frac{(0-1)(0-2)}{(3-1)(3-2)} = 1$$

4. At this step, players convert their shares to a single share based on $\mathit{\Delta}$ and γ_i-s, and erase their old shares, shown in $\mathcal{E}_{n\times n}$:

$$\varphi_1(x) = (3)10 + (-3)17 + (1)5 = -16 \qquad \varphi_3(x) = (3)15 + (-3)12 + (1)15 = 5$$
$$\varphi_2(x) = (3)9 + (-3)5 + (1)2 = 14 \qquad \varphi_4(x) = (3)2 + (-3)18 + (1)12 = -17$$

The original secret α can be reconstructed by the new threshold $t' = 4$ as follows:

$$\alpha = \frac{(0-2)(0-3)(0-4)}{(1-2)(1-3)(1-4)}(-16) + \frac{(0-1)(0-3)(0-4)}{(2-1)(2-3)(2-4)}(14)$$
$$+ \frac{(0-1)(0-2)(0-4)}{(3-1)(3-2)(3-4)}(5) + \frac{(0-1)(0-2)(0-3)}{(4-1)(4-2)(4-3)}(-17) = -16 \overset{19}{\equiv} 3$$

5.2 Active Adversary Model

In this section, a modified version of the previous approach is illustrated which is also secure against an active adversary. The paper [17] presents such a construction and use it in a different context [16], e.g., for creating a proactive secret sharing scheme. We present a similar approach in order to change the threshold, and formally prove that it is not secure in the mobile adversary setting; although this has been stated in [16], the authors have not provided a formal proof for this claim.

Secret Sharing ($\boldsymbol{\mathcal{S}ha}$). Suppose an honest dealer initiates a secret sharing scheme by using a symmetric bivariate polynomial, i.e., he randomly generates a polynomial $f(x,y) \in \mathbb{Z}_q[x,y]$ 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 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 3. *In a VSS, when the dealer applies a symmetric polynomial $f(x,y)$ to generate shares, each pair of players P_i and P_j are able to check the validity of their shares through pairwise channels. The matrix representing those values is called pairwise check matrix.*

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

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 (i.e., top left to bottom right) after executing the $\mathcal{S}ha$ phase.

In the next re-sharing, details of the accusation and defense procedures among players are removed to simplify the scheme; see [5, 19] for details of pairwise checks.

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

1. Each P_i randomly creates a symmetric polynomial $g_i(x, y)$ of degree $t' - 1$ s.t.:
$$g_i(x, 0) = f_i(x) \tag{9}$$
   (a) Generate the symmetric bivariate polynomial $g'_i(x, y)$, where $a_{k0} = a_{0l} = 0$ for $0 \leq k, l \leq t' - 1$.
   (b) Extend $f_i(x)$ to a symmetric polynomial $f'_i(x, y)$ by adding corresponding y-terms.
   (c) Finally, compute $g_i(x, y) = f'_i(x, y) + g'_i(x, y)$, 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 $1 \leq i, j \leq n$, 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:
$$\mathcal{E}_{n\times n} = \begin{pmatrix} g_1(x, \omega^1) & g_1(x, \omega^2) & \cdots & g_1(x, \omega^n) \\ g_2(x, \omega^1) & g_2(x, \omega^2) & \cdots & g_2(x, \omega^n) \\ \vdots & \vdots & \ddots & \vdots \\ g_n(x, \omega^1) & g_n(x, \omega^2) & \cdots & g_n(x, \omega^n) \end{pmatrix}$$

3. Players first perform pairwise checks on $g_i(x, \omega^j)$ for $1 \leq i \leq n$, i.e., n pairwise check matrices. They then perform a single pairwise check, as shown below, on $g_i(0, \omega^j)$ to make sure that constant terms of shares are consistent with original shares distributed by the honest dealer:
$$\begin{pmatrix} - & g_1(0, \omega^2) & \cdots & g_1(0, \omega^n) \\ g_2(0, \omega^1) & - & \cdots & g_2(0, \omega^n) \\ \vdots & \vdots & \ddots & \vdots \\ g_n(0, \omega^1) & g_n(0, \omega^2) & \cdots & - \end{pmatrix} \overset{?}{=} \begin{pmatrix} - & f_1(\omega^2) & \cdots & f_1(\omega^n) \\ f_2(\omega^1) & - & \cdots & f_2(\omega^n) \\ \vdots & \vdots & \ddots & \vdots \\ f_n(\omega^1) & f_n(\omega^2) & \cdots & - \end{pmatrix}$$

4. A set Δ is determined s.t. it contains the identifiers of t good players (defined after pairwise checks), and 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 ($\boldsymbol{\mathcal{R}ec}$). Now, if at least t' players P_j cooperate, where $j \in \Delta'$ and $|\Delta'| \geq t'$, they can recover α by using the Lagrange interpolation method:

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

Example 3. Shares of $f(x, y) = \mathbf{11}+3x+3y+4x^2+4y^2+xy^2+x^2y+7xy+9x^2y^2 \in \mathbb{Z}_{13}$ are initially distributed by the dealer among four players, where $t = 3$ and $\omega = 2$:

$$\begin{aligned} f_1(x) &= f(x, 2^1) = \mathbf{7} + 8x + 3x^2 & \qquad f_3(x) &= f(x, 2^3) = \mathbf{5} + 6x + 3x^2 \\ f_2(x) &= f(x, 2^2) = \mathbf{9} + 8x + 9x^2 & f_4(x) &= f(x, 2^4) = \mathbf{4} + 7x + 10x^2 \end{aligned}$$

1. Players re-share their shares with $g_i(x, y)$. For the sake of simplicity, let $t' = 3$.

$$\begin{aligned} g_1 &= f'_1 + g'_1 = (\mathbf{7} + 8x + 3x^2 + 8y + 3y^2) + (3xy + 7xy^2 + 7x^2y + 5x^2y^2) \\ g_2 &= f'_2 + g'_2 = (\mathbf{9} + 8x + 9x^2 + 8y + 9y^2) + (7xy + 8xy^2 + 8x^2y + 7x^2y^2) \\ g_3 &= f'_3 + g'_3 = (\mathbf{5} + 6x + 3x^2 + 6y + 3y^2) + (8xy + 5xy^2 + 5x^2y + 3x^2y^2) \\ g_4 &= f'_4 + g'_4 = (\mathbf{4} + 7x + 10x^2 + 7y + 10y^2) + (4xy + 9xy^2 + 9x^2y + 2x^2y^2) \end{aligned}$$

2. The $\mathcal{E}_{n \times n}$, where each P_i generates a row and receives a column, is as follows:

$$\begin{pmatrix} 9 + 3x + 11x^2 & 9 + 2x + 7x^2 & 3 + 12x + 2x^2 & 6 + 2x + 4x^2 \\ 9 + 2x + x^2 & 3 + 8x + 10x^2 & 12 + 4x + x^2 & 10 + 10x + 5x^2 \\ 3 + 3x + 12x^2 & 12 + x + 6x^2 & 11 + x^2 & 11 + 10x + 6x^2 \\ 6 + 12x + 10x^2 & 10 + 11x & 11 + 4x + 2x^2 & 11 + 9x + 3x^2 \end{pmatrix}$$

3. They then perform n pairwise checks on the shares that they have received. These matrices must be symmetric with respect to the main diagonal:

$$\begin{pmatrix} 0 & 2 & 9 & 0 \\ 2 & 0 & 5 & 0 \\ 9 & 5 & 0 & 5 \\ 0 & 0 & 5 & 0 \end{pmatrix} \begin{pmatrix} 0 & 7 & 11 & 11 \\ 7 & 0 & 5 & 0 \\ 11 & 5 & 0 & 7 \\ 11 & 0 & 7 & 0 \end{pmatrix} \begin{pmatrix} 0 & 12 & 2 & 3 \\ 12 & 0 & 1 & 4 \\ 2 & 1 & 0 & 7 \\ 3 & 4 & 7 & 0 \end{pmatrix} \begin{pmatrix} 0 & 6 & 1 & 2 \\ 6 & 0 & 7 & 4 \\ 1 & 7 & 0 & 2 \\ 2 & 4 & 2 & 0 \end{pmatrix}$$

4. Now, each P_i has to store four shares or players need to define a set Δ, which contains the identifiers of t good players, in order to convert these shares to a single share. Suppose $\Delta = \{P_1, P_2, P_3\}$, we therefore have (in $\bmod 13$):

$$\gamma_1 = \frac{(0-4)(0-8)}{(2-4)(2-8)} = 7 \qquad \gamma_2 = \frac{(0-2)(0-8)}{(4-2)(4-8)} = 11 \qquad \gamma_3 = \frac{(0-2)(0-4)}{(8-2)(8-4)} = 9$$

5. Players convert their shares to a single share and erase their shares in $\mathcal{E}_{n \times n}$:

$$\begin{aligned} \varphi_1(x) &= \mathbf{7} + 5x + x^2 & \qquad \varphi_3(x) &= \mathbf{5} + 11x + 8x^2 \\ \varphi_2(x) &= \mathbf{9} + 7x + 5x^2 & \varphi_4(x) &= \mathbf{4} + 6x + 7x^2 \end{aligned}$$

Players, say P_2, P_3, P_4, can recover the secret α by the threshold $t' = 3$ as follows:

$$\alpha = \frac{(0-8)(0-16)}{(4-8)(4-16)}(\mathbf{9}) + \frac{(0-4)(0-16)}{(8-4)(8-16)}(\mathbf{5}) + \frac{(0-4)(0-8)}{(16-4)(16-8)}(\mathbf{4}) = \mathbf{11}$$

It is not difficult to show that, in the active adversary model, it suffices to use the constant terms to reconstruct the secret. In other words, if at least t players P_j interpolate $f_j(0)$-s, the secret $\alpha = f(0,0)$ is revealed:

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

Claim. The re-sharing protocol $\mathcal{A}re$ is not secure under the mobile adversary model. This means the constant terms of the players' shares stay the same after re-sharing; whether we change the threshold or not.

Proof. If these constants stay the same, the mobile adversary can incrementally collect players' shares in different time periods in order to recover the secret. We simply prove the equality $\varphi_j(0) = f_j(0)$ for P_j where $j \in \{1 \dots n\}$ (see Example 3).

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

□

Claim. In the re-sharing protocol $\mathcal{A}re$, each player has to store several shares (this may threaten the security of the scheme even more) unless participants agree on a set Δ of good players with a cardinality exactly equal to t.

Proof. First of all, we should mention that if players do not agree on Δ, they each need to keep n shares, as shown in Examples 3: steps $2, 4, 5$. But this claim says, even if players agree on a set of good players, the cardinality of this set must be exactly t. Let assume $|\Delta| \ne t$, we therefore have:

1. Suppose $|\Delta| > t$. As a result, the number of possible combinations of $|\Delta|$ values taken t at a time defines the number of possible sets of constants:

   $$\binom{|\Delta|}{t} = d \text{ then } \Phi = \{\{\gamma_{1,1} \dots \gamma_{1,t}\}, \dots, \{\gamma_{d,1} \dots \gamma_{d,t}\}\} \text{ and } |\Phi| = d$$

   Consequently, each player has to save d possible shares $\varphi_j(x)$-s according to each set of constants in Φ.

2. If $|\Delta| < t$, then players are not able to execute this protocol since we need at least t shares, in the fourth and fifth steps, in order to make a single share.

If $t = |\Delta|$, then players are able to compute a single set of constants $\Phi = \{\{\gamma_1 \dots \gamma_t\}\}$ because $\binom{t}{t} = 1$. As a result, each P_i updates his share to a single share $\varphi_j(x)$. □

6 Increasing the Threshold and Changing the Secret

In this section, we discuss how to change both the threshold t and the secret α after the initialization when the dealer no longer exists. Our goal is to generate a new secret based on the linear combination of the previous secret at each stage, i.e., $\alpha_{i+1} = \alpha_i \beta_i + \delta_i$ where β_i and $\delta_i \in \mathbb{Z}_q$ are unknown. (The case where β_i's and δ_i's are known is a well-known problem, called *secure polynomial evaluation.*)

The first problem for players is to generate verified shares of an unknown secret in the absence of the dealer, named *verified polynomial production* protocol. It is also easy to show that the scheme can select $\beta_i = 1$ in order to increase the threshold without changing the secret at any stage. We should mention that we only present the multiplication case since the addition is much simpler.

The proposed protocols in our constructions consist of n participants $P_1 \dots P_n$. In the passive adversary model (Appendix), there are only private channels between each pair of players, but in the active adversary setting, we also assume the existence of a synchronous broadcast channel. Let $\mathbb{Z}_q$ be a finite field and let ω be a primitive element in this field. All computations are performed in the field $\mathbb{Z}_q$.

6.1 Active Adversary Model

In the active adversary model, the general idea is to multiply 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 an unknown constant term β.

To create $g(x, y)$, we develop a new protocol with a similarity to the initialization method in [26, 5]. In that construction, a dealer initiates a secret sharing scheme under the assumption that $t-1 \leq \left\lfloor \frac{n-1}{4} \right\rfloor$. The reason behind this assumption is that the dishonest dealer may disrupt $\frac{1}{4}$ of the shares in the initialization phase and $\frac{1}{4}$ out of the remaining $\frac{3}{4}$ shares might be disrupted by colluders. Therefore, a suitable error correction technique, such as the Reed-Solomon code [12], can be used to recover the secret correctly. Our scheme is a dealer-free protocol for generating a verified random symmetric polynomial $g(x, y)$ under the assumption that $t-1 \leq \left\lfloor \frac{n-1}{3} \right\rfloor$, where $\xi = t-1$ is the number of colluders that the scheme can tolerate. The reason for this assumption is the fact that our scheme is dealer-free, therefore, $\frac{1}{3}$ of the entire shares can be corrupted. Suppose $f(x, y)$ is a symmetric polynomial and $f(x, \omega^i)$ is the share of each player P_i in the initial setting.

`Phase-1: Dealer-Free Verified Polynomial Production`

1. To construct $g(x, y)$, t players P_i are chosen based on a group agreement or a random selection. Suppose the first t players are selected, i.e., $1 \leq i \leq t$. Each P_i 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:

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

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

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

   In fact, shares $g_k(x)$ associated with players' identifiers generate 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 create shares on $g(x,y)$ for the other $n-t$ players P_j, where $t+1 \leq j \leq n$, the following sub-protocol is repeated $n-t$ times:

   (a) Each 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, 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. Subsequently, if P_i finds that the above equality does not hold, he then broadcasts (i,j), that is, P_i is accusing P_j.
5. Each P_i computes a subset $\Gamma \subseteq \{1,...,n\}$ s.t. any ordered pair $(i,j) \in \Gamma\times\Gamma$ has not been broadcasted (Γ is a *clique*). The authors in [8] construct this clique by the *maximal matching* problem which has a polynomial time solution.
6. If $|\Gamma| \geq n-\xi$, 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 and players proceed to the next phase. Otherwise, another set of t players is chosen to create $g(x,y)$.

At the end of the phase-1, all good players belonging to Γ have consistent shares with respect to an unknown secret β. We should mention that this protocol is successfully passed if the first t players act honestly even if some of them are malicious; this is not unlikely since sometimes bad players behave honestly in some steps in order to remain in the scheme and act maliciously during the secret recovery. As an alternative solution, we provide another protocol which guarantees a correct solution but tolerates less colluders, i.e., $t-1 \leq \lfloor \frac{n-1}{4} \rfloor$, and needs $t+1$ executions of a $\mathcal{VSS}$.

1. Initially, $t+1$ players P_i are selected at random in order to act as independent dealers; they each might be honest or malicious.
2. Each P_i shares a secret, say β_i, by the $\mathcal{VSS}$ of [5] where the degree is $t-1$. Sharing is accepted if all good players have consistent shares with respect to β_i.
3. Each player locally adds shares of secrets β_i-s together. Now, each player has a share on a symmetric polynomial of degree $t-1$ with a constant term $\beta = \sum \beta_i$.

If a party is disqualified in the second step, another player P_i can be selected to share a new β_i. It also worth mentioning that since we have $t+1$ dealers, β remains secret even if $t-1$ colluders in our scheme reveal their β_i-s.

`Phase-2: Secure Multiplication of Two Secrets`

In this step, each player P_i simply multiplies his two shares $f(x,\omega^i)$ and $g(x,\omega^i)$ together, and keeps the result, which is a point on the symmetric polynomial $h(x,y) = f(x,y) \times g(x,y)$ of degree $2t-2$ with a constant term $\alpha\beta$. Honest players also erase all the other values.

An obvious solution for decreasing the threshold is to reveal some shares, but this approach forces players to save extra information along with their personal shares, which might be a new threat to the security of the entire scheme. Therefore, to adjust the threshold, we extend the degree reduction and randomization method in [9] (the simplified version of [2]) to the case with bivariate polynomials.

`Phase-3: Verified Degree Reduction and Randomization`

1. Each 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 $1 \le j \le n$:

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

2. Parties compute the first row of a publicly known matrix $\mathcal{V}^{-1}_{n\times n}$ to adjust the threshold ($\mathcal{V}_{n\times n}$ is the Vandermonde matrix for $[\omega^1, \omega^2, \cdots, \omega^n]$; $\mathcal{V}_{i,j} = (\omega^i)^{(j-1)}$ 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, $\widetilde{h}(x,y)$ 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 $\alpha\beta$:

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

Theorem 2. *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 (the assumption is $t-1 \leq \left\lfloor \frac{n-1}{4} \right\rfloor$ if the alternative solution is used in the first phase).*

Proof. The security of this protocol is similar to the proofs in [26, 5], therefore, we just provide a further clarification. Since our polynomials remain symmetric during all three phases, players can perform pairwise checks through secure channels at any time during the execution of the protocol in order to detect malicious players who deviate (similar to the first phase: steps $4, 5, 6$). This is even simpler than the approach in Appendix B of [9]. For the sake of simplicity, suppose $3|(n-1)$, we therefore have the following assumption for the least possible threshold t (which might be increased to t' later on):

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

This means 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 [12], we can correct all $2\xi/2 = \xi$ faulty shares in our scheme and interpolate a unique polynomial that encodes our secret.

If we use the alternative solution in the first phase, each P_i (as an independent dealer) may also disrupt at most $t-1$ shares when he is sharing an unknown β_i (otherwise he is disqualified). As a result, the protocol works under the assumption that $t-1 \leq \left\lfloor \frac{n-1}{4} \right\rfloor$. □

7 Conclusion

We constructed a new dealer-free dynamic scheme in the unconditionally secure setting by applying existing techniques as well as developing new protocols. In our constructions, participants do not need to save extra shares ahead of time, and both the threshold and the secret (based on the linear combination of previous secrets) can be changed to arbitrary values multiple times, which is usable in many applications as we mentioned.

Our main construction is dealer-free, unconditional, and also is secure in the active adversary model. In fact, it is quite challenging to design a protocol in this setting. In other words, if one relaxes any of these assumptions, he can therefore decrease the computation and communication complexities, for instance, by using a trusted authority, or relying on computational assumptions such as the hardness of factoring, or considering the simple passive adversary model.

Acknowledgments

We would like to thank Ryan Henry, Professor Ronald Cramer, and the anonymous reviewers for their helpful and constructive comments.

References

1. Barwick, S.G., Jackson, W.A., Martin, K.M.: Updating the parameters of a threshold scheme by minimal broadcast. IEEE Transactions on Information Theory 51(2), 620–633 (2005)
2. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: 20th Annual ACM Symposium on Theory of Computing, STOC. pp. 1–10 (1988)
3. Blakley, G.R.: Safeguarding cryptographic keys. In: National Computer Conference, New York. AFIPS Conference proceedings, vol. 48, pp. 313–317. AFIPS Press (1979)
4. Blundo, C., Cresti, A., Santis, A.D., Vaccaro, U.: Fully dynamic secret sharing schemes. Theoretical Computer Science 165(2), 407–440 (1996)
5. D'Arco, P., Stinson, D.R.: On unconditionally secure robust distributed key distribution centers. In: Proceedings of the 8th Int. Conf. on the Theory and Application of Cryptology and Info. Security, ASIACRYPT. pp. 346–363. LNCS, Springer (2002)
6. Desmedt, Y., Jajodia, S.: Redistributing secret shares to new access structures and its applications. In: Technical Report ISSE TR-97-01. George Mason University (1997)
7. Frankel, Y., Gemmell, P., MacKenzie, P.D., Yung, M.: Optimal-resilience proactive public-key cryptosystems. In: 38th Annual Symposium on Foundations of Computer Science, FOCS. pp. 384–393. IEEE Computer Society (1997)
8. Gennaro, R., Ishai, Y., Kushilevitz, E., Rabin, T.: The round complexity of verifiable secret sharing and secure multicast. In: 33th Annual ACM Symposium on Theory of Computing, STOC. pp. 580–589 (2001)
9. Gennaro, R., Rabin, M.O., Rabin, T.: Simplified vss and fast-track multiparty computations with applications to threshold cryptography. In: 17th annual ACM symposium on Principles of Distributed Computing, PODC. pp. 101–111 (1998)
10. He, J., Dawson, E.: Multistage secret sharing based on one-way function. Electronics Letters 30(19), 1591–1592 (1994)
11. Herzberg, A., Jarecki, S., Krawczyk, H., Yung, M.: Proactive secret sharing or: How to cope with perpetual leakage. In: 15th Annual International Cryptology Conference, CRYPTO. LNCS, vol. 963, pp. 339–352. Springer (1995)
12. MacWilliams, F., Sloane, N.: The theory of error-correcting codes. North-Holland Amsterdam (1978)
13. Maeda, A., Miyaji, A., Tada, M.: Efficient and unconditionally secure verifiable threshold changeable scheme. In: 6th Australasian Conference Information Security and Privacy, ACISP. pp. 403–416. LNCS, Springer (2001)
14. Martin, K.M., Pieprzyk, J., Safavi-Naini, R., Wang, H.: Changing thresholds in the absence of secure channels. In: Proceedings of the 4th Australasian Conference Information Security and Privacy, ACISP. LNCS, vol. 1587, pp. 177–191. Springer (1999)
15. Martin, K.M., Safavi-Naini, R., Wang, H.: Bounds and techniques for efficient redistribution of secret shares to new access structures. The Computer Journal 42(8), 638–649 (1999)
16. Nikov, V., Nikova, S.: On proactive secret sharing schemes. In: 11th International Workshop on Selected Areas in Cryptography, SAC. LNCS, vol. 3357, pp. 308–325. Springer (2004)
17. Nikov, V., Nikova, S., Preneel, B.: Multi-party computation from any linear secret sharing scheme unconditionally secure against adaptive adversary: The zero-error case. In: 1st Int. Applied Crypto and Network Sec, ACNS. LNCS, vol. 2846, pp. 1–15. Springer (2003)
18. Nishide, T., Ohta, K.: Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. In: 10th Int. Conf. on Practice and Theory in Public-Key Cryptography, PKC. pp. 343–360. LNCS, Springer (2007)

19. Nojoumian, M., Stinson, D., Grainger, M.: Unconditionally secure social secret sharing scheme. IET Information Security, Special Issue on Multi-Agent and Distributed Information Security 4(4), 202–211 (2010)
20. Nojoumian, M., Stinson, D.R.: Unconditionally secure sealed-bid auction protocols using verifiable secret sharing. In: Submitted (2011)
21. Ostrovsky, R., Yung, M.: How to withstand mobile virus attacks. In: 10th Annual ACM Symposium on Principles of Distributed Computing, PODC. pp. 51–59 (1991)
22. Shamir, A.: How to share a secret. Communications of the ACM 22(11), 612–613 (1979)
23. Simmons, G.J.: How to (really) share a secret. In: 8th Annual International Cryptology Conference, CRYPTO. LNCS, vol. 403, pp. 390–448. Springer (1988)
24. Steinfeld, R., Wang, H., Pieprzyk, J.: Lattice-based threshold-changeability for standard shamir secret-sharing schemes. In: 10th Int. Conf. on the Theory and Application of Crypto and Info Sec, ASIACRYPT. LNCS, vol. 3329, pp. 170–186. Springer (2004)
25. Stinson, D.R.: Cryptography: Theory and Practice,Third Edition. CRC Press (2005)
26. Stinson, D.R., Wei, R.: Unconditionally secure proactive secret sharing scheme with combinatorial structures. In: Proceedings of the 6th Annual Int. Workshop on Selected Areas in Cryptography, SAC. LNCS, vol. 1758, pp. 200–214. Springer (1999)
27. Suzuki, K., Yokoo, M.: Secure combinatorial auctions by dynamic programming with polynomial secret sharing. In: 6th International Conference on Financial Cryptography, FC. LNCS, vol. 2357, pp. 44–56. Springer (2002)
28. Tartary, C., Wang, H.: Dynamic threshold and cheater resistance for shamir secret sharing scheme. In: 2nd SKLOIS Conference on Information Security and Cryptology, Inscrypt. LNCS, vol. 4318, pp. 103–117. Springer (2006)
29. Tassa, T.: Hierarchical threshold secret sharing. In: 1st Theory of Cryptography Conference, TCC. LNCS, vol. 2951, pp. 473–490. Springer (2004)
30. Yao, A.C.: Protocols for secure computations. In: 23rd Annual Symposium on Foundations of Computer Science, FOCS. pp. 160–164. IEEE Computer Society (1982)

Appendix: Passive Adversary Model

The general idea is the same as the active adversary construction. Suppose $f(i)$ is the share of the player P_i from the original secret sharing polynomial.

`Threshold and/or Secret Changeability Protocol`

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 implicitly 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 the other $n-t$ participants, the *enrollment* protocol in [19] can be used to create $g(k)$ for the relevant players P_k. Since $t+1 \leq k \leq n$, the following protocol is repeated $n-t$ times:

   (a) Each P_i for $1 \leq i \leq t$ computes his corresponding Lagrange interpolation constant: $\gamma_i = \prod_{1 \leq j \leq t, i \neq j} (k-j)/(i-j)$, where i, j, k are players' *ids*.

   (b) Subsequently, each participant P_i multiplies his share φ_i by his Lagrange interpolation constant, and randomly splits the result into t portions, i.e., $g(i) \times \gamma_i = \partial_{1i} + \partial_{2i} + \cdots + \partial_{ti}$ for $1 \leq i \leq t$.

(c) Players exchange ∂_{ji}'s accordingly through pairwise channels. Therefore, each P_j holds t values. P_j adds them together and sends the result to P_k, that is, $\sigma_j = \sum_{i=1}^{t} \partial_{ji}$.

(d) Finally, P_k adds these values σ_j for $1 \leq j \leq t$ together to compute his share $g(k) = \sum_{j=1}^{t} \sigma_j$.

3. At this stage, each participant P_i simply multiplies his two shares $f(i)$ and $g(i)$ together, and keeps the result, which 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.
4. Each 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 $1 \leq j \leq n$, as a result, each player receives a vector of shares, i.e., a column in the share-exchange matrix:

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

5. 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 $1 \leq i, j \leq n$. Suppose this vector is $\mathcal{V}^{-1}_{1\times n} = (v_1 \quad v_2 \quad \cdots \quad v_n)$.
6. 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 $\alpha\beta$:

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