Dealer-Free Threshold Changeability in 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 paper proposes a dealer-free threshold changeable construction for secret sharing schemes. In practice, the adversary's ability might be enhanced over time, for instance by compromising more players. This problem can be resolved only by increasing the threshold. In the literature, there exist some techniques to address this issue. These solutions either have a large storage requirement or are limited to a predefined threshold modification. In addition, they increase the threshold at the side of the combiner with some mathematical assumptions. We apply two secure multiparty computation techniques [2, 16] to tackle these problems. In our constructions, participants do not need to save any information or extra shares ahead of time, and the 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: secure multiparty computations and threshold secret sharing schemes

1 Introduction

In *secret sharing schemes*, a secret divided into different shares for distribution among several participants, and a subset of participants then collaborate to recover the secret [19, 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.

The secret sharing scheme is an essential tool used in *secure multiparty computation* [22] 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 [10], shared RSA keys [5], and electronic auctions with sealed-bid [11]. In practice, the adversary's ability or computation power might be enhanced over time, for instance by compromising more players. In other words, changing the threshold is more required as the lifetime of a secret increases. This problem can be tackled only by increasing the threshold in the absence of the *dealer*, i.e., the entity who initializes the scheme.

In the literature, there exist some solutions to address this issue. They either have a large storage requirement or are limited to a predefined threshold modification. In addition, they increase the threshold at the side of the *combiner*, i.e., the entity who recovers the secret, and consider some strong mathematical assumptions for secret recovery in the existence of noise. First of all, these constructions are not immune against any coalition of malicious participants. Second, the secret can be reconstructed if an adversary attacks the participants rather than the combiner. Our motivation is to apply secure multiparty computation methods in order to tackle these problems and add more utility to the threshold changeable schemes.

1.1 Our Contributions

The contribution of this paper is to construct a threshold changeable scheme with a variety of desirable properties as follows:

1. Our constructions take place in a *dealer-free* framework, meaning that participants change the threshold based on a group agreement after the initialization.
2. They are *unconditionally* secure in the sense that they do not rely on any computational assumptions such as the hardness of factoring or discrete logarithms.
3. They have the minimum *storage cost* since players do not need to save any information or extra shares in advance in order to change the threshold subsequently.
4. They are *flexible* constructions since the threshold can be changed to any arbitrary values multiple times with the presence of enough participants.
5. The univariate construction is an *ideal* threshold changeable scheme since the shares of players are taken from the same domain as that of the secret [6].

We also extend the proposed approach for the secure multiplication of two secrets [2], that is the coefficient randomization and the polynomial truncation, to the case where bivariate polynomials are used.

1.2 Organization

This paper is organized as follows. Section 2 reviews existing constructions for changing the threshold in secret sharing schemes. Section 3 provides the required background for this paper. Section 5 illustrates the proposed dealer-free threshold changeable scheme. Section 6 discusses the security of the construction. Finally, Section 7 outlines concluding remarks and future directions.

2 Related Work

K. M. 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 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 threshold. Second, the proposed construction assumes that shareholders behave honestly in the sense that they only use the subshares that is relevant to the threshold in current use.

By using the prior approach, A. Maeda et al. [13] illustrate an unconditionally secure verifiable scheme in which the threshold can be changed several times, say N, 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. Authors assume that the secret is not recovered before the threshold changeability, therefore, no share has been pooled.

R. Steinfeld et al. [20] construct a threshold changeable mechanism for the standard Shamir secret sharing schemes. 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 [21] 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 these solutions. In fact, if an adversary attacks the shareholders (not the combiner) then he can have access to the original shares, i.e., full 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, 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 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 two secure multiparty computation techniques which is used in our construction in Section 5, all the computations are performed in the field $GF(q)$.

3.1 Secure Multiplication of Secrets

Ben-Or et al. [2] proposed a method for the secure multiplication of two secrets which then simplified by R. Gennaro et al. [9]. Suppose secrets a_1 and a_2 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, $a_1 \times a_2$, 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, they 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 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, H is the coefficient vector of $h(x)$, S is an evaluation vector of $h(x)$ at $2t-1$ points, i.e., players' shares on $h(x)$, K is the coefficient vector of $k(x)$, R is an evaluation vector of $k(x)$ at t points, i.e., players' shares on $k(x)$, $B_{n\times n}$ is the transpose of a Vandermonde matrix, and $P_{n\times n}$ is a projection matrix, i.e., if $i = j$ and $i, j \leq t$ then $P_{ij} = 1$, otherwise $P_{ij} = 0$. Then, we have:

$$H = H \cdot B \cdot B^{-1} \tag{1}$$

$$H \cdot P = K \tag{2}$$

$$K \cdot B = R \tag{3}$$

$$H \cdot B = S \tag{4}$$

$$(1),(2) \Rightarrow \; H \cdot B \cdot B^{-1} \cdot P = K \tag{5}$$

$$(3),(5) \Rightarrow \; H \cdot B \cdot B^{-1} \cdot P \cdot B = R \tag{6}$$

$$(4),(6) \Rightarrow \; S \cdot \underbrace{B^{-1} \cdot P \cdot B}_{\text{publicly known}} = R \tag{7}$$

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 on the projection matrix's diagonal. To randomize the coefficients of $k(x)$ in the above computation, each player P_i randomly selects a polynomial $q_i(x)$ of degree $2t-2$ with a zero constant term. Then we can use $\widetilde{h}(x)$ instead of $h(x)$, as shown in the following equation, in the degree reduction protocol, which will be explained in detail afterward.

$$\widetilde{h}(x) = h(x) + \sum_{i=1}^{n} q_i(x), \text{ satisfying } \widetilde{h}(0) = h(0)$$

3.2 Enrollment of a New Participant

In this section, we review the proposed protocols for enrolling new players in a threshold secret sharing scheme. Suppose a dealer initiates a (t, n)-threshold scheme based on a polynomial $f(x)$ of degree $t-1$ by distributing n shares among participants, and then leaves the scheme, i.e., the dealer is not accessible anymore. The question is how players can securely collaborate to generate corresponding shares on $f(x)$ for new participants without revealing their own shares. This problem can be resolved if any subset of t participants, where t is the threshold, cooperate together. Herzberg et al. [12] propose the first solution for this problem, called *share recovery*, but their solution is not efficient because of its random shuffling procedure. Subsequently, Saxena et al. [18] provide a non-interactive technique by using bivariate polynomials, called *bivariate admission control*, but this protocol is secure under the discrete logarithm assumption. The latest solution is presented by Nojoumian et al. [16], which is an efficient protocol with unconditional security and verifiability, this protocol is as follows:

1. First, each participant P_i computes its corresponding Lagrange interpolation constant as follows:

$$C_i = \prod_{1 \le j \le t}^{j \ne i} \frac{x - x_j}{x_i - x_j}, \text{ where } x \text{ is the new player's } id, \; x_i \text{ and } x_j \text{ represent other players' } ids$$

2. Second, participants multiply their shares in related Lagrange interpolation constants. After that, each participant randomly splits the result into t portions:

$$S_1 \times C_1 = s_{1,1} + s_{2,1} + \cdots + s_{t,1}, S_2 \times C_2 = s_{1,2} + s_{2,2} + \cdots + s_{t,2}, \cdots, S_t \times C_t = s_{1,t} + s_{2,t} + \cdots + s_{t,t}$$

3. Third, each participant keeps one share-portion for himself, and exchanges the rest of them with other $t-1$ players. As a result, each player P_j holds t values; so, he adds those values together and sends the result to the new participant:

$$v_j = \sum_{i=1}^{t} s_{j,i}, \text{ where } s_{j,i} \text{ is the } j^{th} \text{ share-portion of the } i^{th} \text{ participant}$$

4. Finally, the new participant adds all these values together to construct his share: $S = \sum_{j=1}^{t} v_j$.

Example 1: Assume $t = 3$ and a dealer generates corresponding shares for three participants with $ids = 1, 2, 3$ based on $f(x) = 3 + 2x + x^2$, i.e., $f(1) = 6$, $f(2) = 11$, and $f(3) = 18$. After some period of time, suppose it is desired to construct a share for a new player, $id = 4$, without having access to the dealer. First each player P_i computes $S_i \times C_i$ as follows: $S_1 \times C_1 = 6 \times \frac{(4-2)(4-3)}{(1-2)(1-3)} = 6$, $S_2 \times C_2 = 11 \times \frac{(4-1)(4-3)}{(2-1)(2-3)} = -33$, and $S_3 \times C_3 = 18 \times \frac{(4-1)(4-2)}{(3-1)(3-2)} = 54$. Then, they cooperate to create a share for the new player, as shown in Figure 1.

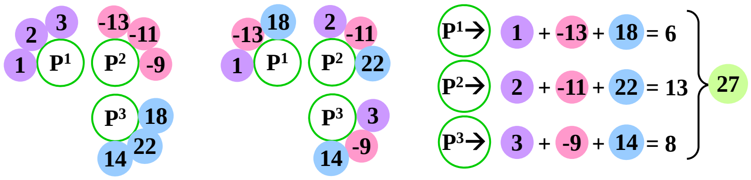

Fig. 1. Enrollment of a New Participant [16]

This protocol can be extended to the situation in which the share generator is a bivariate polynomial, i.e., each share is a polynomial $\in \mathbb{Z}_p[x]$. Using a symmetric bivariate polynomial, the scheme can be verifiable in the sense that newcomers verify their shares with existing players. The security proof of this protocol has been given in [16].

4 Simple Construction by Re-sharing Shares

In this section, we review a simple technique for threshold changeability in secret sharing schemes. The general idea is to re-share existing shares of a (t, n)-threshold scheme by a set of polynomials of degree t', i.e., converting a (t, n)-threshold scheme into a (t', n)-threshold scheme [8]. The re-sharing protocol is as follows:

1. Initially, the dealer generates a polynomial $f(x)$ of degree $t-1$ in which its constant term is the secret, i.e., $f(0) = s$, and then sends shares of players P_i accordingly, that is $f(i)$ for $1 \leq i \leq n$.
2. Now, participants want to switch to a new threshold of t'. Each player P_i generates a polynomial $g_i(x)$ of degree $t'-1$ such that $g_i(0) = f(i)$, i.e., its constant term is the player's share.
3. 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.

4. At this step, a set $\mathit{\Delta}$ is determined such that it contains the identifiers of at least t good players. For the sake of simplicity suppose $|\mathit{\Delta}| = t$. The following constants are computed:

$$C_i = \prod_{i,j \in \mathit{\Delta}}^{i \neq j} (0 - j) \times (i - j)^{-1}, \text{ where } i \text{ and } j \text{ represent players' } \textit{ids}$$

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

$$S_i = \sum_{j \in \mathit{\Delta}} (C_j \times g_j(i))$$

6. At this point, if at least t' players P_i cooperate, where $i \in \mathit{\Gamma}$ and $|\mathit{\Gamma}| \geq t'$, they can recover the secret by using the Lagrange interpolation:

$$s = \sum_{i,j \in \mathit{\Gamma}} \left(S_i \times \prod_{i \neq j} (0 - j) \times (i - j)^{-1} \right)$$

Theorem: *In the re-sharing protocol for threshold changeability, each player has to store several shares unless a set $\mathit{\Delta}$ of good players with $|\mathit{\Delta}| = t$ is predetermined.*

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

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

In this case, each player has to save either d possible S_i's or t' shares he has originally received from other participants. In the case of $|\mathit{\Delta}| = t$, players are able to compute a single set of constants $\mathit{\Phi} = \{\{C_{1,1}, C_{1,2}, \ldots, C_{1,t}\}\}$, i.e., $\binom{t}{t} = |\mathit{\Phi}| = 1$. As a consequence, each player P_i updates his share to a single value S_i. Since the cardinality of $\mathit{\Phi}$ is 1, all players belonging to $\mathit{\Delta}$ must be honest in order to compute s correctly. Finally, if $|\mathit{\Delta}| < t$, then players are not able to execute this protocol since it is the condition of the forth step. To conclude, it is not practical to predetermine a set of good players ahead of time, and storing extra shares will threaten the security of the scheme more.

5 Our Constructions

5.1 Setting

The goal is to change the threshold from t to t' after the initialization phase, when the dealer no longer exists. The proposed model consists of n participants, $P_1, P_2, \ldots, P_n$, with private channels between each pair of them, and an authenticated public broadcast channel, on which information is transmitted instantly and accurately to all players. If the decision is to keep the secret value constant, 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 [17, 12]; otherwise, the secret value itself must be changed. Let $GF(q)$ be a finite field and let ω be a primitive element in this field; all the computations are performed in the field $GF(q)$.

5.2 Dealer-Free Threshold Changeability

In this section, we first provide a secure protocol for changing the threshold in a secret sharing scheme with constant secret; this protocol is a dealer-free scheme with a univariate polynomial as the share generator. We then extend the proposed approach to the situation in which the share generator is a bivariate polynomial. Finally, we clarify the scenario where players decide to change the secret value themselves. The proposed approach in [2], also formulated by [23], is applied in the following constructions.

Using Univariate Polynomials with a Constant Secret. The general idea is to multiply the original polynomial $f(x)$ of degree $t-1$, encoding a secret a_1, by a random polynomial $g(x) = 1 + xg'(x)$ of degree t with constant term equal to 1; the degree of $g'(x)$ is $t-1$. As a consequence, the same secret value will be kept even after changing the threshold. This is a desirable property in the case where the secret is difficult or expensive to be changed.

Since the degree of $f(x) \times g(x)$ is $2t-1$, we need the contribution of at least $2t$ participants in order to be able to use the secure multiplication protocol illustrated in Section 3.1. This condition can be satisfied in the weighted secret sharing context if the total weight of contributing participants is equal or bigger than the twice the primary threshold $\sum w_{P_i} \geq 2t$.

1. Initially each player has a share on $f(x)$. To satisfy the $n \geq 2t$ condition, execute the enrollment protocol presented in Section 3.2, in order to enroll new players to the scheme or increase the weights of some participants.
2. Based on a group agreement or a random selection, t players will be chosen so that each generates a private random number for himself. In fact, these t random values associated with players' identifiers form a polynomial of degree $t-1$, known as $g'(x)$.
3. Selected players apply the enrollment protocol one more time to generate corresponding shares on $g'(x)$ for other participants. Now, each player has a share on $f(x)$ as well as $g'(x)$.
4. To calculate shares of $g(x) = 1+xg'(x)$, each player multiplies his share on $g'(x)$ by his identifier, and then adds one to the result. Consequently, each player also has a share on $g(x)$.
5. In this step, each participant simply multiplies his two shares on $f(x)$ and $g(x)$ together, and keeps the new value, i.e., a point on $h(x) = f(x) \times g(x)$.
6. To randomize coefficients of $h(x)$, each participant P_i generates a random polynomial $q_i(x)$ of degree $2t-1$ with a zero constant term, and gives $q_i(j)$ to P_j for $1 \leq j \leq n$.
7. Each participant P_j adds his share on $h(x)$ and all values he received from other participants together, and erases all the other values.

$$S_j = f(j) \times g(j) + \sum_{i=1}^{n} q_i(j)$$

Now, the secret is encoded by a polynomial of degree $2t-1$ with random coefficients, i.e., the threshold is $2t$ at this point. To adjust the threshold, we use the degree reduction protocol shown in Section 3.1; therefore, players have to exchange their shares securely for the next stage.

8. Each player P_i generates a random polynomial $r_i(x)$ of degree $t'-1$, where t' is the new threshold value based on players' consensus, with a constant term equal to his share, i.e., $r_i(0) = S_i$, and gives $r_i(j)$ to P_j for $1 \leq j \leq n$.
9. Participants then construct a publicly known matrix, $A_{n\times n} = B^{-1}\cdot P\cdot B$, to adjust the threshold, where $B_{n\times n}$ is the transpose of the Vandermonde matrix, and $P_{n\times n}$ is a projection matrix, i.e. if $i = j$ and $i, j \leq t'$ then $p_{ij} = 1$, otherwise $p_{ij} = 0$.
10. Each participant P_j multiplies the vector $[r_1(j), \cdots, r_n(j)]$ by the matrix $A_{n\times n}$. Suppose the resulting vector is $[v_{j1}, \cdots, v_{jn}]$. Consequently, P_j sends v_{ji} to P_i for $1 \leq i \leq n$.
11. Finally, each player P_i interpolates $(1, v_{1i}), \cdots, (n, v_{ni})$ and constructs a new polynomial, where its constant term is the new share of participant P_i with respect to the new threshold t'.

To recover the secret, t' participants have to collaborate in order to construct a polynomial of degree $t'-1$, where its constant term is the secret a_1. The example of this protocol is demonstrated in the next part.

Example 2: Suppose we have three players, P_1, P_2, and P_3, sharing the secret value $a_1 = 3$ with $f(x) = 3 + 7x$ over finite field $GF(13)$, i.e., the threshold is $t = 2$ and players' shares are $S_1 = 10$, $S_2 = 4$, and $S_3 = 11$ respectively. Players decide to increase the threshold to $t' = 3$ while keeping the same secret value. The procedure would be as follows:

1. To satisfy the $n \geq 4$ condition, players sign up a new participant P_4 with the enrollment protocol, see Example 1. As a result, new player's share is $S_4 = 5$.
2. By an agreement or a random selection, $t = 2$ players, say P_2 and P_3, generate two private random numbers $S'_2 = 10$ and $S'_3 = 6$ accordingly.
3. By having $S'_2 = 10$ and $S'_3 = 6$, i.e. $g'(x) = 5 + 9x$, players P_2 and P_3 can use the enrollment protocol to generate $S'_1 = 1$ and $S'_4 = 2$ as new shares on $g'(x)$ for P_1 and P_4.
4. Players compute $g(x) = 1 + xg'(x)$ as follows: $S''_1 = 1 + (1 \times 1) = 2$, $S''_2 = 1 + (2 \times 10) = 8$, $S''_3 = 1 + (3 \times 6) = 6$, and $S''_4 = 1 + (4 \times 2) = 9$, i.e. $g(x) = 1 + 5x + 9x^2$.
5. Now, each player P_i multiplies his shares on $f(x)$ and $g(x)$ together, $S_i = S_i \times S''_i$. Consequently we have $S_1 = 7$, $S_2 = 6$, $S_3 = 1$, and $S_4 = 6$, i.e. $h(x) = 3 + 9x + 10x^2 + 11x^3$.
6. To randomize coefficients of $h(x)$, players generate the following random polynomials of degree $2t-1 = 3$ with zero constant term respectively: $q_1(x) = 11x+11x^2+12x^3$, $q_2(x) = 2x+6x^2+2x^3$, $q_3(x) = 8x+5x^2+7x^3$, and $q_4(x) = 6x+12x^2+6x^3$. Then each P_i gives $q_i(j)$ to P_j for $1 \leq j \leq n$. Here is the matrix presentation of shares exchange in which P_i generates i^{th} row for other players and receives i^{th} column from others.

$$SharesExchange : \begin{pmatrix} 8 & 6 & 1 & 0 \\ 10 & 5 & 10 & 11 \\ 7 & 1 & 11 & 1 \\ 11 & 4 & 2 & 2 \end{pmatrix}$$

7. Players compute $S_j = f(j)\times g(j)+\sum_{i=1}^{n} q_i(j)$ to update their shares: $S_1 = 7+[8+10+7+11] = 4$, $S_2 = 6 + [6 + 5 + 1 + 4] = 9$, $S_3 = 1 + [1 + 10 + 11 + 2] = 12$, and $S_4 = 6 + [0 + 11 + 1 + 2] = 7$, i.e. $\widetilde{h}(x) = 3 + 10x + 5x^2 + 12x^3$. Now players erase all the other values they are holding.
8. To truncate $\widetilde{h}(x)$, each player P_i generates a random polynomial of degree $t'-1 = 2$ with constant term equal to his share: $r_1(x) = 4 + 6x + 6x^2$, $r_2(x) = 9 + 7x + x^2$, $r_3(x) = 12 + 2x + 3x^2$,

and $r_4(x) = 7 + 6x + x^2$. Then each P_i gives $r_i(j)$ to P_j for $1 \leq j \leq n$. Here is another matrix presentation of shares exchange.

$$SharesExchange : \begin{pmatrix} 3 & 1 & 11 & 7 \\ 4 & 1 & 0 & 1 \\ 4 & 2 & 6 & 3 \\ 1 & 10 & 8 & 8 \end{pmatrix}$$

9. Participants then construct a publicly known matrix, $A_{n\times n} = B^{-1} \cdot P \cdot B$, to adjust the threshold to the new value $t' = 3$ as follows:

$$A_{n\times n} = \begin{pmatrix} 4 & 0 & 8 & 2 \\ 7 & 3 & 9 & 7 \\ 4 & 6 & 10 & 6 \\ 12 & 4 & 12 & 11 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 3 \\ 1 & 8 & 1 & 12 \end{pmatrix} = \begin{pmatrix} 12 & 10 & 11 & 2 \\ 6 & 10 & 6 & 7 \\ 7 & 4 & 8 & 6 \\ 2 & 3 & 2 & 12 \end{pmatrix}$$

10. Each participant P_j then multiplies the vector $[r_1(j), \cdots, r_n(j)]$ by the matrix $A_{n\times n}$.

$$\begin{gathered} P_1 : [3, 4, 4, 1] \cdot A_{n\times n} = [12, 11, 0, 5] \\ P_2 : [1, 1, 2, 10] \cdot A_{n\times n} = [0, 6, 1, 11] \\ P_3 : [11, 0, 6, 8] \cdot A_{n\times n} = [8, 2, 3, 11] \\ P_4 : [7, 1, 3, 8] \cdot A_{n\times n} = [10, 12, 6, 5] \end{gathered}$$

Afterward, they exchange new values. As a consequence, each player is holding four ordered values. $P_1 : \{12, 0, 8, 10\}$, $P_2 : \{11, 6, 2, 12\}$, $P_3 : \{0, 1, 3, 6\}$, and $P_4 : \{5, 11, 11, 5\}$.

11. Finally, each player interpolates those values in order to construct a polynomial whose constant term is the player's new share.

$$\begin{gathered} P_1 : \big\{(1, 12), (2, 0), (3, 8), (4, 10) \longmapsto 5 + 10x + 10x^2\big\} \rightarrow S_1 = 5 \\ P_2 : \big\{(1, 11), (2, 6), (3, 2), (4, 12) \longmapsto 4 + 7x^2\big\} \rightarrow S_2 = 4 \\ P_3 : \big\{(1, 0), (2, 1), (3, 3), (4, 6) \longmapsto 6x + 7x^2\big\} \rightarrow S_3 = 0 \\ P_4 : \big\{(1, 5), (2, 11), (3, 11), (4, 5) \longmapsto 6 + 2x + 10x^2\big\} \rightarrow S_4 = 6 \end{gathered}$$

If any $t' = 3$ players collaborate, they can first construct the polynomial $\widehat{h}(x) = 3 + 10x + 5x^2$, i.e., truncation of $\widetilde{h}(x)$, and then recover the secret $a_1 = 3$.

Using Bivariate Polynomials with a Constant Secret. In this case, the general idea is the same as the earlier protocol, i.e., multiplying the original polynomial $f(x, y)$ of degree $t-1$, encoding a secret a_1, by a random polynomial $g(x, y)$ of degree t with constant term equal to 1. Consequently, the same secret value will be kept even after changing the threshold. Applying bivariate polynomials is important since then the secret sharing protocol can be extended to a verifiable scheme.

1. Initially, each player has a share on $f(x, y)$, i.e., $f(x, \omega^{id})$. To satisfy the $n \geq 2t$ condition, call the bivariate version of the enrollment protocol presented in Section 3.2, in order to enroll new players to the scheme or increase the weights of most deserving participants.
2. Based on a group agreement or a random selection, t players will be chosen so that each generates a private random polynomial $\in \mathbb{Z}_p[x]$ of degree at most $t-1$ for himself. In fact, these t random polynomials (second shares) associated with players' identifiers, i.e., $y = \omega^{id}$, form a bivariate polynomial $\in \mathbb{Z}_p[x, y]$ of degree at most $t-1$, known as $g'(x, y)$.

3. Selected players apply the bivariate version of the enrollment protocol another time to generate corresponding shares on $g'(x, y)$ for other participants. Now, each player has a share on $f(x, y)$ as well as $g'(x, y)$.
4. To calculate shares of $g(x, y)$ of degree t with constant term equal to 1, the following equation must be computed:

$$g(x, y) = 1 + x \times g'(x, y) + c \times m(y) + y^t \times n(x)$$

   where $g'(x, y)$ is the player's second share, c is a random constant number, $m(y) \in \mathbb{Z}_p[y]$ is a random polynomial of degree t with zero constant term, and $n(x) \in \mathbb{Z}_p[x]$ is a random polynomial of degree t with zero constant term; c, $m(y)$, and $n(x)$ are publicly generated by players' agreement. Then, each player replaces y with ω^{id}, consequently, each player also has a share on $g(x, y)$.
5. In this step, each participant simply multiplies his two shares on $f(x, y)$ and $g(x, y)$ together, and keeps the new value, i.e., a point on $h(x, y) = f(x, y) \times g(x, y)$ of degree $2t - 1$.
6. To randomize coefficients of $h(x, y)$, each participant P_i generates a random polynomial $q_i(x, y)$ of degree $2t - 1$ with a zero constant term, and gives $q_{i,j}(x, \omega^j)$ to P_j for $1 \leq j \leq n$.
7. Each player P_j adds his share on $h(x, y)$ and all values he received from other players together, and erases all the other values.

$$S_j = f(x, \omega^j) \times g(x, \omega^j) + \sum_{i=1}^{n} q_{i,j}(x, \omega^j)$$

   Now, the secret is encoded by a polynomial of degree $2t - 1$ with random coefficients, i.e. the threshold is $2t$ at this point. To adjust the threshold, we use the degree reduction protocol shown in Section 3.1; therefore, players have to exchange their shares, which is a polynomial in x of degree $2t - 1$, securely for the next stage.

8. Each player P_i generates a random polynomial $r_i(x, y)$ of degree $t' - 1$, where t' is the new threshold value based on players' consensus; this polynomial does not have any constant and x-alone terms, i.e., terms with $x^k y^l$ for arbitrary k and l are fine but terms consisting of only x^k are not acceptable. Then, each player adds the truncation of his share to $r_i(x, y)$; only terms with the degree of less than or equal to t', i.e., $r_i(x, 0) =$ truncation of S_i, and gives $r_{i,j}(x, \omega^j)$ to P_j for $1 \leq j \leq n$.
9. Participants then construct a publicly known matrix, $A_{n\times n} = B^{-1} \cdot P \cdot B$, to adjust the threshold, where $B_{n\times n}$ is the transpose of the Vandermonde matrix for $[\omega^1, \omega^2, \cdots, \omega^n]$, and $P_{n\times n}$ is a projection matrix, i.e. if $i = j$ and $i, j \leq t'$ then $p_{ij} = 1$, otherwise $p_{ij} = 0$.
10. Each player P_j multiplies the vector $[r_{1,j}(x, \omega^j), \cdots, r_{n,j}(x, \omega^j)]$ by the matrix $A_{n\times n}$. Suppose the resulting vector is $[v_{j,1}(x), \cdots, v_{j,n}(x)]$. Consequently, P_j sends $v_{j,i}(x)$ to P_i for $1 \leq i \leq n$.
11. Finally, each player P_i interpolates $(\omega^1, v_{1,i}(x)), \cdots, (\omega^n, v_{n,i}(x))$ by a bivariate interpolation and constructs a new polynomial $\in \mathbb{Z}_p[x, y]$, where its constant and x-alone terms are the new share of participant P_i with respect to the new threshold t'.

To recover the secret, t' players have to collaborate in order to construct a bivariate polynomial of degree $t' - 1$, where its constant term is the secret a_1. This scheme can be verifiable in the case of using, for instance, symmetric bivariate polynomial. However, we will address this issue in the future work. An example of the presented protocol is demonstrated in the Appendix.

Extending the Protocols for Variable Secrets. In the case of changing the secret value, i.e., a situation in which majority of players decide to immunize the scheme against malicious participants, the constant term of $g(x)$ or $g(x, y)$ can be an arbitrary secret value a_2 known to only t' participants, where $t' > t$. As a consequence, the new secret value of the scheme would be $a_1 \times a_2$.

To fulfill this task, more than t players, say t', must be chosen in the second step of the proposed protocols to directly construct $g(x)$ or $g(x, y)$ without using $g'(x)$ or $g'(x, y)$. In this case, the degree of the resulting polynomial $h(x)$ or $h(x, y)$ is $t + t' - 2$. Accordingly, the initial condition for the secure multiplication of secrets will be changed to the contribution of at least $t+t'-1$ participants.

6 Security Analysis

The security of the first two constructions can be illustrated by the fact that the secret remains constant and it cannot be computed at the end of the protocols (i.e., t is changed to t' where $t' > t$) by an adversary attacking $t' - 1$ players or a new coalition of $t' - 1$ malicious participants.

First of all, the constructed polynomial $g(x)$ or $g(x, y)$ has a constant value equal to 1, as a consequence, the resulting multiplication keeps the secret value the same. In the next stage, while the coefficients of the polynomial $h(x)$ or $h(x, y)$ are being randomized, each participant generates a random polynomial $q(x)$ or $q(x, y)$ with a zero constant term. At the end, players truncate the polynomial $h(x)$ or $h(x, y)$ such that the first t' terms will stay constant. Therefore, the secret value does not change.

Second, during the first seven steps, any coalition of t players can definitely recover the secret, therefore, the system is resistant to a coalition of $t - 1$ malicious players, but immediately after erasing old shares by at least $n - t + 1$ players (an inevitable assumption as mentioned earlier), the threshold will be $2t$ which then is decreased to t'. Consequently, an adversary or a new coalition of bad players needs at least t' shares to recover the secret.

Finally, we should clarify that the security of some phases of our constructions relies on the security of two multiparty computation techniques used in the proposed protocols; the security proofs of these techniques are provided in [2, 16]. Moreover, as we stated previously, the second bivariate protocol can be extended to a verifiable scheme. This verifiability can further strengthen the proposed construction by helping honest participants to immunize the scheme (i.e., changing the secret value form a_1 to $a_1 \times a_2$) in the case of observing malicious behaviors.

7 Conclusions and Future Work

We created a new dealer-free threshold changeable scheme by using secure multiparty computation techniques proposed for multiplication of secrets [2] and new player enrollments in threshold schemes [16]. In our construction, participants do not need to save any information or extra shares ahead of time, and the threshold can be increased or decreased multiple times (by adjusting the number of ones in the projection matrix) to any arbitrary values. It is an unconditionally secure threshold changeable scheme in the sense that it does not rely on any computational assumptions. In addition, the presented constructions realize the proactive model since the players' shares are updated at the time of the threshold changeability.

In future work we intend to develop a dealer-free verifiable threshold changeable scheme by using the bivariate construction, extend the coefficient randomization and polynomial truncation to multivariate polynomials, and evaluate the communication and time complexity of the constructed protocols.

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] R. M. Capocelli, A. D. Santis, L. Gargano, and U. Vaccaro, *On the size of shares for secret sharing schemes*, J. Cryptology, 6 (1993), pp. 157–167.

[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, 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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/.

[17] 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.

[18] N. Saxena, G. Tsudik, and J. H. Yi, *Efficient node admission for short-lived mobile ad hoc networks*, in Proceedings of the 13th IEEE International Conference on Network Protocols, IEEE Computer Society, 2005, pp. 269–278.

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

[20] 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.

[21] 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.

[22] 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.

[23] M. Yung, *Secure distributed computing: Theory and practice*, in WDAG '94: Proceedings of the 8th International Workshop on Distributed Algorithms, London, UK, 1994, Springer-Verlag, pp. 53–73.

Appendix

Example 3: Suppose we have three players, P_1, P_2, and P_3, sharing the secret value $a_1 = 7$ with $f(x) = 7 + 5x + 5y + 9xy$ over finite field $GF(13)$ and $\omega = 2$, i.e., the threshold is $t = 2$ and players' shares are $S_1 = 4 + 10x$, $S_2 = 1 + 2x$, and $S_3 = 8 + 12x$ respectively. Players decide to increase the threshold to $t' = 3$ while keeping the same secret value. The procedure would be as follows:

1. To satisfy the $n \geq 4$ condition, players sign up a new participant P_4 with the enrollment protocol, see Example 1. As a result, new player's share is $S_4 = 9 + 6x$.
2. By an agreement or a random selection, $t = 2$ players, say P_2 and P_3, generate two private random polynomials $S'_2 = 4 + 5x$ and $S'_3 = 3 + 7x$ accordingly.
3. By having S'_2 and S'_3, i.e. $g'(x, y) = 5 + 3x + 3y + 7xy$, players P_2 and P_3 can use the enrollment protocol to generate $S'_1 = 11 + 4x$ and $S'_4 = 1 + 11x$ as new shares on $g'(x, y)$ for P_1 and P_4.
4. Players compute $g(x, y) = 1 + xg'(x, y) + 3(7y + 3y^2) + y^2(5x + 3x^2)$ as follows: $S''_1 = 1 + 5x + 3x^2$, $S''_2 = 8 + 6x + x^2$, $S''_3 = 4 + 11x + 4x^2$, and $S''_4 = 2 + 7x + 12x^2$, that is:
$$g(x, y) = 1 + 5x + 3x^2 + 3xy + 7x^2y + 21y + 9y^2 + 5y^2x + 3y^2x^2$$
5. Now, each participant P_i multiplies his shares on $f(x, y)$ and $g(x, y)$ together, $S_i = S_i \times S''_i$. Consequently we have $S_1 = 4 + 4x + 10x^2 + 4x^3$, $S_2 = 8 + 9x + 2x^3$, $S_3 = 6 + 6x + 8x^2 + 9x^3$, and $S_4 = 5 + 10x + 7x^2 + 7x^3$, that is:
$$h(x, y) = 7+9y+x+12y^2+4xy+7x^2+6y^3+11xy^2+7x^2y+2x^3+2xy^3+4x^2y^2+10x^3y+8x^2y^3+x^3y^3$$
6. To randomize coefficients of $h(x, y)$, players generate the following random polynomials of degree $2t - 1 = 3$ with zero constant term respectively:
$$\begin{aligned} q_1(x, y) &= 3x^3 + x^2y + 11xy^2 + 3y^3 + 10xy + 12y^2 + 8x + 5y \\ q_2(x, y) &= 10x^3 + 5x^2y + 11xy^2 + 8y^3 + 4x^2 + 3xy + 3y^2 + 3x + 4y \\ q_3(x, y) &= 10x^3 + 10x^2y + 10xy^2 + 8y^3 + x^2 + 12xy + 10x + 9y \\ q_4(x, y) &= 12x^3 + 12x^2y + 7xy^2 + 2y^3 + x^2 + 2xy + 4y^2 + 8x + 6y \end{aligned}$$
Then each P_i gives $q_{i,j}(x, \omega^j)$ to P_j for $1 \leq j \leq n$. Here is the matrix presentation of shares exchange in which P_i generates i^{th} row for other players and receives i^{th} column from others.
$$\begin{pmatrix} 3x^3 + 2x^2 + 7x + 4 & 3x^3 + 4x^2 + 3x + 1 & 3x^3 + 8x^2 + 12x + 4 & 3x^3 + 3x^2 + 7x + 9 \\ 10x^3 + x^2 + x + 6 & 10x^3 + 11x^2 + 9x + 4 & 10x^3 + 5x^2 + 3x + 4 & 10x^3 + 6x^2 + 7x + 8 \\ 10x^3 + 8x^2 + 9x + 4 & 10x^3 + 2x^2 + 10x + 2 & 10x^3 + 3x^2 + 5x + 8 & 10x^3 + 5x^2 + 6x + 9 \\ 12x^3 + 12x^2 + x + 5 & 12x^3 + 10x^2 + 11x + 8 & 12x^3 + 6x^2 + 4x + 2 & 12x^3 + 11x^2 + 12x + 4 \end{pmatrix}$$
7. Participants compute $S_j = f(x, \omega^j) \times g(x, \omega^j) + \sum_{i=1}^{n} q_{i,j}(x, \omega^j)$ in order to update their shares: $S_1 = 10+9x+7x^2$, $S_2 = 10+3x+x^2+11x^3$, $S_3 = 11+4x+4x^2+5x^3$, and $S_4 = 9+3x+6x^2+3x^3$.
$$\widetilde{h}(x, y) = 7+7y+4x+5y^2+5xy+y^3+11xy^2+9x^2y+11x^3+2xy^3+4x^2y^2+10x^3y+8x^2y^3+x^3y^3$$
now players erase all the other values they are holding.
8. To truncate $\widetilde{h}(x, y)$, each player P_i generates a random polynomial $r_i(x, y)$ of degree $t' - 1 = 2$ without any constant and x-alone terms, and adds the truncation of his share to $r_i(x, y)$, i.e., only terms with the degree of less than or equal to t'.

$$\begin{gathered}
r_1(x,y) = 5x^2y^2 + 7x^2y + 9xy^2 + 3xy + 5y^2 + 2y + (10 + 9x + 7x^2) \\
r_2(x,y) = 7x^2y^2 + 8x^2y + 4xy^2 + 7xy + 12y^2 + 11y + (10 + 3x + x^2) \\
r_3(x,y) = 5x^2y^2 + 7x^2y + 6xy^2 + 8xy + 3y^2 + 2y + (11 + 4x + 4x^2) \\
r_4(x,y) = 2x^2y^2 + 2x^2y + 5xy^2 + 4xy + y^2 + 9y + (9 + 3x + 6x^2)
\end{gathered}$$

Then, each P_i gives $r_{i,j}(x, \omega^j)$ to P_j for $1 \leq j \leq n$. Here is the matrix of the shares exchange.

$$\begin{pmatrix}
2x^2 + 12x + 8 & 11x^2 + 9x + 7 & 6x^2 + 11x + 8 & 8x^2 + 8x + 9 \\
6x^2 + 7x + 2 & 2x^2 + 4x + 12 & 6x^2 + 3x + 8 & 10x^2 + 8x + 8 \\
12x^2 + 5x + 1 & 8x^2 + 2x + 2 & 3x^2 + 10x + 11 & 5x^2 + 4x + 5 \\
5x^2 + 5x + 5 & 7x^2 + 8x + 9 & 7x^2 + 4x + 2 & 4x^2 + 8x + 6
\end{pmatrix}$$

9. Players now construct a publicly known matrix, $A_{n\times n} = B^{-1} \cdot P \cdot B$, to adjust the threshold to the new value $t' = 3$, where B is the transpose of the Vandermonde matrix for $[2, 4, 8, 16]$.

$$A_{n\times n} = \begin{pmatrix} 8 & 3 & 11 & 1 \\ 6 & 4 & 0 & 8 \\ 5 & 0 & 10 & 9 \\ 8 & 6 & 5 & 8 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 4 & 8 & 3 \\ 4 & 3 & 12 & 9 \\ 8 & 12 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 6 & 1 & 8 & 12 \\ 1 & 9 & 12 & 5 \\ 6 & 9 & 8 & 4 \\ 1 & 8 & 12 & 6 \end{pmatrix}$$

10. Each participant P_j then multiplies the vector $[r_{1,j}(x, \omega^j), \cdots, r_{n,j}(x, \omega^j)]$ by the matrix $A_{n\times n}$.

$$\begin{gathered}
[2x^2{+}12x{+}8 \quad 6x^2{+}7x{+}2 \quad 12x^2{+}5x{+}1 \quad 5x^2{+}5x{+}5] \Rightarrow [4x^2{+}10x{+}9 \quad 9x^2{+}4x{+}10 \quad 10x^2{+}7x \quad 2x^2{+}8x{+}10] \\
[11x^2{+}9x{+}7 \quad 2x^2{+}4x{+}12 \quad 8x^2{+}2x{+}2 \quad 7x^2{+}8x{+}9] \Rightarrow [6x^2{+}10 \quad x^2{+}10x{+}10 \quad 11x{+}12 \quad 8x^2{+}2x{+}11] \\
[6x^2 + 11x + 8 \quad 6x^2 + 3x + 8 \quad 3x^2 + 10x + 11 \quad 7x^2 + 4x + 2] \Rightarrow [2x^2 + 3x + 7 \quad 4x \quad 7x^2 + 5x + 12 \quad 3x + 10] \\
[8x^2{+}8x{+}9 \quad 10x^2{+}8x{+}8 \quad 5x^2{+}4x{+}5 \quad 4x^2{+}8x{+}6] \Rightarrow [x^2{+}10x{+}7 \quad 6x^2{+}11x{+}5 \quad 12x^2{+}2x{+}7 \quad 8x^2{+}5x{+}9]
\end{gathered}$$

Afterward, they exchange elements of the resulting vectors. As a consequence, each player is holding four ordered polynomials of degree t'.

11. Finally, each player interpolates those polynomials in order to construct a bivariate polynomial whose constant and x-alone terms are the player's new share.

$$\begin{gathered}
(2\ ,\ 4x^2 + 10x + 9), (4\ ,\ 6x^2 + 10), (8\ ,\ 2x^2 + 3x + 7), (16\ ,\ x^2 + 10x + 7) \\
2 + 6x + 5y + 12xy + 8x^2 + 9y^2 + 3x^2y + 8xy^2 + 4x^2y^2 \rightarrow S_1 = 2 + 6x + 8x^2
\end{gathered}$$

$$\begin{gathered}
(2\ ,\ 9x^2 + 4x + 10), (4\ ,\ x^2 + 10x + 10), (8\ ,\ 4x), (16\ ,\ 6x^2 + 11x + 5) \\
11 + 5x + 9y + xy + 9x^2 + 5y^2 + 2x^2y + 9xy^2 + 12x^2y^2 \rightarrow S_2 = 11 + 5x + 9x^2
\end{gathered}$$

$$\begin{gathered}
(2\ ,\ 10x^2 + 7x), (4\ ,\ 11x + 12), (8\ ,\ 7x^2 + 5x + 12), (16\ ,\ 12x^2 + 2x + 7) \\
6 + 7x + 12y + 12xy + 3x^2 + 12y^2 + 11x^2y + 7xy^2 + 6x^2y^2 \rightarrow S_3 = 6 + 7x + 3x^2
\end{gathered}$$

$$\begin{gathered}
(2\ ,\ 2x^2 + 8x + 10), (4\ ,\ 8x^2 + 2x + 11), (8\ ,\ 3x + 10), (16\ ,\ 8x^2 + 5x + 9) \\
8 + x + 11y + 10xy + 11x^2 + 8y^2 + 8x^2y + 10x^2y^2 \rightarrow S_4 = 8 + x + 11x^2
\end{gathered}$$

If any $t' = 3$ participants collaborate, they can first construct the truncation of $\widetilde{h}(x)$, i.e., $\widehat{h}(x) = 7 + 4x + 7y + 5xy + 5y^2 + 9x^2y + 11xy^2 + 4x^2y^2$, and then recover the secret $a_1 = 7$.