Unconditionally Secure Proactive Verifiable Secret Sharing Using New Detection and Recovery Techniques

Mehrdad Nojoumian
Department of Electrical & Computer Engineering and Computer Science
Florida Atlantic University, Boca Raton, Florida, USA
mnojoumian@fau.edu

***Abstract*—In this paper, a new *proactive verifiable secret sharing* (PVSS) scheme is proposed in an active mobile adversary setting. To the best of our knowledge, the only unconditionally secure PVSS is proposed in ASIACRYPT'02 by D'Arco and Stinson. In this protocol, the authors assume the existence of private channels as well as an authenticated broadcast channel. This scheme uses symmetric bivariate polynomials for secret sharing, i.e., each share is a polynomial over a finite field rather than a single field element, under the assumption that $t < \frac{n}{4}$ where t is the threshold. We propose a new PVSS scheme with prominent properties using a simple *detection* method along with a novel *recovery-and-renewal* technique. First of all, we only consider private channels in our setting. In addition, our scheme can tolerate $t < \frac{n}{3}$ corrupted players. Although we utilize a VSS scheme as a subprotocol in our scheme, shares of the secret are single field elements after the initialization by the dealer.**

Keywords: cryptographic methods, unconditional security, secret sharing schemes, and active mobile adversary.

I. Introduction

In *secret sharing*, a secret is divided into different shares to be distributed among several players. Subsequently, a subset of parties cooperate to reveal the secret [1], [2]. In a threshold (t,n)-*secret sharing scheme*, where $t < n$, the secret is divided into n shares such that any t players can collaborate to reveal the secret, but any subset of $t-1$ parties cannot learn anything about the secret. For instance, in Shamir secret sharing [1], the dealer initially selects a random polynomial $f(x) \in \mathbb{Z}_q[x]$ of degree $t-1$ such that $f(0)$ is the secret. He then distributes shares $f(i)$ among players P_i for $1 \leq i \leq n$. As a result, any set of t or more players can recover the secret using the Lagrange interpolation method whereas any set of size less than t cannot gain any information about the secret.

In *verifiable secret sharing* VSS [3], players are able to verify the consistency of their shares in the scheme's initialization and also the correctness of the shares in other phases. The authors in [4] and [5] provide the first unconditionally secure VSS where $t < \frac{n}{3}$ with a zero probability of error. They only assume the existence of secure private channels between each pair of players.

The protocols in [6] and [7] consider both private channels and an authenticated broadcast channel to construct a new VSS scheme where $t < \frac{n}{2}$. This protocol has a negligible probability of error. To simplify these constructions, the authors in [8] and [9] propose VSS schemes based on symmetric bivariate polynomials in an unconditionally secure setting. These protocols also utilize private channels along with an authenticated broadcast channel under the assumption that $t < \frac{n}{4}$.

Subsequently, the notion of *proactive secret sharing* PSS is introduced in [10], where shares of the players are updated without changing the secret. The share renewal is accomplished by adding some random polynomials with zero constant terms to the original secret sharing polynomial. In fact, PSS is proposed to deal with a *mobile adversary* [11] who collects the shares of an increasing number of players over time in order to finally recover the secret.

A. Our Security Model

To construct a new secret sharing scheme, first the security model must be defined. In a *passive adversary* setting, players follow protocols correctly but are curious to learn the secret. On the other hand, in an *active adversary* model, players may deviate from protocols while trying to learn the secret.

The passive or active adversary might be *static* or *mobile*. The former refers to an adversary who corrupts players ahead of time before the protocol starts, whereas in the latter case, the adversary may corrupt various players while the protocol is executing. In addition, the security model might be *computational*, meaning that the security of the protocol relies on a computational assumption such as the hardness of factoring or discrete logarithm, or *unconditional*, meaning that the adversary has unlimited computational power.

In our model, we consider an *active mobile adversary* who can corrupt at most t players between two consecutive periods during the protocol's execution. He can collect shares, disrupt players, and show any arbitrary behavior. This is the most challenging adversarial setting.

B. Our Motivation and Contribution

Our motivation is to improve the proposed construction of [12], where the authors propose a proactive verifiable secret sharing scheme in an unconditionally secure setting with $t < \frac{n}{4}$. They use a symmetric bivariate polynomial for secret sharing, i.e., each share is a polynomial in $\mathbb{Z}_q[x]$. This construction assumes the existence of both private channels and a broadcast

channel. The problem of tolerating $t < \frac{n}{3}$ corrupted players remains open in the conclusion of this paper.

We therefore provide a new protocol for unconditionally secure proactive verifiable secret sharing using a simple *detection* method along with a novel *recovery-and-renewal* technique for corrupted shares. Compared to [12], our protocol has the following prominent properties:

- It can tolerate an active mobile adversary who corrupts $t < \frac{n}{3}$ players.
- Only a private channel between each pair of the players is required.
- Shares are single field elements after the initialization by the dealer.

II. Review of the Asiacrypt'02 Paper

We now briefly review the proposed scheme in [12]. For the sake of simplicity, we only present the proactive part of the construction. Suppose a dealer distributes shares of a secret α by a symmetric bivariate polynomial $f(x,y) \in \mathbb{Z}_q[x,y]$ of degree $t-1$, i.e., $f(0,0) = \alpha$ and $f_i(x) = f(x,i) \in \mathbb{Z}_q[x]$ is the share of P_i. As mentioned earlier, in this scheme, private channels and a broadcast channel are considered.

The goal is to tolerate an active mobile adversary where $t > b+1$ and b denotes the number of corrupted parties. The proactivity consists of three steps: *(a) renewal*: the servers/players update their shares, as shown in Figure 1, *(b) detection*: the corrupted servers are detected and rebooted, and *(c) recovery*: new shares are generated for the rebooted servers, as shown in Figure 2.

After the renewal phase, each uncorrupted player has an updated share with respect to the original secret α. It is clear that all the corrupted players may not be detected during this stage, i.e., some corrupted parties may act honestly during this update phase. Therefore, the players perform pairwise checks through point-to-point secure channels to detect the corrupted shares. Finally, new shares are generated for the corrupted players/servers once these servers are detected and rebooted.

III. Our Construction

Our proposed model consists of n players $P_1 \dots P_n$. Let assume VSS of [4] is used to initiate a secret sharing scheme in which the secret is α. This VSS only assumes the existence of private channels and works under the assumption that $t < \frac{n}{3}$. All computations are done in a finite field $\mathbb{Z}_q$ where q is a large prime number.

We should stress that, in this construction, the share of each player P_i is a polynomial $f_i(x) \in \mathbb{Z}_q[x]$. It is not difficult to show that only the constant terms of the shares, that is, $f_i(0)$, are enough to recover the secret. Therefore, once the dealer initializes the scheme, we assume that each player P_i only keeps a single filed element $\alpha_i = f_i(0) \in \mathbb{Z}_q$ as his share of the secret and he discards the other part, which was used to qualify/disqualify the dealer. We also assume that the *active mobile adversary* can corrupt at most t players between each two consecutive periods. We first provide a simple protocol to generate shares of a random finite field element "r", as illustrated in Figure 3.

Renewal

1) Each P_l acts as a dealer and selects a random symmetric polynomial of degree $t-2$. He sends $g_i^l(x) = g^l(x,i)$ to P_i for $1 \le i \ne l \le n$ through a secure channel.
$$g^l(x,y) = \sum_{i=0}^{t-2}\sum_{j=0}^{t-2} g_{ij}x^i y^j \text{ where } g_{ij} = g_{ji} \text{ for all } i,j.$$
2) To verify the shares distributed by P_l, each pair of players P_i and P_j perform pairwise checks $g_i^l(j) = g_j^l(i)$ through private channels.
3) If P_j finds that the equality does not hold for more than b values of i (his share $g_j^l(x)$ is not consistent with other shares), P_j broadcasts an accusation of P_l.
4) If P_l is accused by at most b players (otherwise P_l is definitely a bad player), he defends himself by broadcasting $g_j^l(x)$, which he had sent to the accusers.
5) Other players P_i, excluding the conflicting parties P_l and P_j, check $g_j^l(i) = g_i^l(j)$ and broadcast "yes" or "no". If, for each broadcasted share, at least $n-b-2$ players broadcast "yes", P_l is not guilty and P_j stores the broadcasted $g_j^l(x)$.
6) Finally, each P_i updates the list of good players Γ who have not been found guilty in the previous step, and then, he updates his shares as follows:
$$f_i(x) = f_i(x) + (x+i)\sum_{l\in\Gamma} g_i^l(x).$$

Fig. 1. (a) Proactive VSS of [12] where $t < \frac{n}{4}$

Detection

1) Player P_i computes and sends $f_i(j)$ to player P_j for $1 \le j \ne i \le n$ through a secure channel.
2) P_j checks if $f_i(j) = f_j(i)$. He then broadcasts an accusation list which contains those identifiers such that the above equality does not hold.
3) Each player updates the list of good player Γ such that it does not contain the identifiers of the players accused by at least $b+1$ parties.

Recovery

1) For each $j \notin \Gamma$, each good player P_i computes and sends share $f_i(j)$ to P_j.
2) P_j recovers his share $f_j(x)$ by points $(i, f_i(j))$ and an error correction technique.

Fig. 2. (b, c) Proactive VSS of [12] where $t < \frac{n}{4}$

RanGen

1) Each P_i, for $1 \leq i \leq n$, acts as an independent dealer and shares a secret r_i among the players using a VSS scheme of degree $t-1$. The share-exchange matrix, where each player generates a row and receives a column, is as follows:

$$\mathscr{A}_{n\times n} = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \dots & r_{nn} \end{pmatrix}.$$

If sharing is accepted, all good players will have consistent shares with respect to r_i. Otherwise, P_i is disqualified and i is removed from the set of good players.

2) Qualified P_i, where $i \in \Gamma$, locally add shares of the secrets r_i-s together to calculate their shares, i.e., $s_i = \sum_{j\in\Gamma} r_{ji}$. As a result, each player has a share on a polynomial $R(x)$ of degree $t-1$ with a random constant term $r = \sum_{i\in\Gamma} r_i$.

Fig. 3. Jointly Generating Shares of a Random Field Element "r"

Note that, in the *RanGen* protocol, if sharing is accepted in the first phase, all good players in Γ will have consistent shares with respect to r_i. Otherwise, player P_i is disqualified and i is removed from Γ.

A. Our Proactive Verifiable Secret Sharing Scheme (PVSS)

The main idea of the first phase is to mask shares of the players and then reveal the masked values in order to detect corrupted shares through Berlekamp-Welch algorithm, as shown in Figure 4. After locating corrupted shares/servers, those servers will be rebooted. We then generate new shares for the players who have lost their shares. At the same time, we update shares of good players, Figure 5.

Example-1: In this example, we show that $\vec{\delta} = (\delta_1, \dots, \delta_n)$ has the same exact errors as of the original shares' vector $\vec{\alpha} = (\alpha_1, \dots, \alpha_n)$. Suppose our field is $\mathbb{Z}_7$, the original shares' vector is $\vec{\alpha} = (3,0,6,0,3)$ and the corrupted vector is $\vec{\alpha_1} = (\mathbf{2},0,6,0,3)$, where α_1 has been changed from 3 to 2 in the first location. According to the Berlekamp-Welch equation [13]:

$$g(x) = f(x)e(x)$$

where $deg(g) = t+k-1, deg(f) = t-1, deg(e) = k$, and $e(x) = (x-e_1)(x-e_2)\dots(x-e_k)$ is the error-locator polynomial of degree k with the leading coefficient of 1.

In our example, $deg(g) = 3$, $deg(f) = 2$ and $deg(e) = 1$, i.e., secret sharing polynomial is $f(x)$, threshold is $t = 3$, and $k = 1$ is the maximum number of errors that can be located. The Berlekamp-Welch equation is converted into a parametric format, where $f(x)$ is replaced with α_i:

$$g_0 + g_1 x + g_2 x^2 + g_3 x^3 = \alpha_i(l_0 + x).$$

Detection

1) Each P_i now has a share $\alpha_i \in \mathbb{Z}_q$ of secret α. He utilizes a VSS scheme to re-share it among the other players, that is, each P_j receives α_{ij} from P_i. The share-exchange matrix is as follows:

$$\mathscr{A}_{n\times n} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nn} \end{pmatrix}.$$

2) Players jointly generate shares of a random mask β by executing the "RanGen" protocol, shown in Figure 3. Now, each P_i also has a share $\beta_i \in \mathbb{Z}_q$.

3) Each player P_i uses a VSS scheme to re-share β_i among other players, i.e., each P_j receives β_{ij} from P_i. Here is the share-exchange matrix:

$$\mathscr{B}_{n\times n} = \begin{pmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1n} \\ \beta_{21} & \beta_{22} & \dots & \beta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{n1} & \beta_{n2} & \dots & \beta_{nn} \end{pmatrix}.$$

4) Each player P_i locally computes δ_i and then reveals this value to everyone:

$$\delta_i = \sum_{j=1}^{n} (\alpha_{ji} + \beta_{ji}).$$

5) Players can publicly use any error correction technique to detect the corrupted players $j \in \nabla$ in vector $\vec{\delta} = (\delta_1, \dots, \delta_n)$, e.g., *Berlekamp-Welch algorithm* [13] can be use to define locations of errors by solving a system of linear equations. From the implementation point of view, corrupted servers will be rebooted at this time, that is, they will no longer have any shares with respect to secret α.

Fig. 4. Detecting Corrupted Shares in Our PVSS Scheme Where $t < \frac{n}{3}$

By replacing (x, α_i) with $(1,\mathbf{2}), (2,0), (3,6), (4,0), (5,3)$ in the above equation, we obtain five equations with five unknowns g_0, g_1, g_2, g_3 and l_0. For the sake if simplicity, we can use Cramer's rule and just find $l_0 = 6$. This mean $e(x) = x+6$ or $e(x) = x-1$ because $6 \equiv -1(\mod 7)$. Therefore, $e_1 = 1$ indicates that the first share is not consistent with other shares and it is corrupted. Now, add a random value $\beta = 3$ (i.e., mask) to each share to form vector $\vec{\alpha_2} = (\mathbf{5},3,2,3,6)$. By using Berlekamp-Welch algorithm again, $e_1 = 1$ is obtained.
End of Example-1.

In the next stage, shares of at least t uncorrupted players will be re-shared while adding a set of random polynomials of degree $t-1$ with zero constant terms to these shares.

Recovery-and-Renewal: Here, the recovery protocol is also equivalent to an *enrollment* protocol.

1) Uncorrupted players P_i, where $1 \le i \notin \nabla \le n$, execute the protocol "RanGen" to generate $t-1$ random elements $\vec{r} = (r_1, \ldots, r_{t-1})$. As a result, each P_i has t secret values as follows: $\alpha_i, r_{1i}, r_{2i}, \ldots, r_{(t-1)i}$.
2) Each player P_i, for $1 \le i \notin \nabla \le n$, sends φ_{ik} to player P_k for $1 \le k \le n$. We should stress that the share-exchange matrix may have different dimensions. In other words, uncorrupted players generate these shares for all players in the scheme including the players with corrupted shares. Subsequently, α_i-s will be erased.

$$\Phi_{i\times n} = \begin{pmatrix} \varphi_{11} & \varphi_{12} & \ldots & \varphi_{1n} \\ \varphi_{21} & \varphi_{22} & \ldots & \varphi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{i1} & \varphi_{i2} & \ldots & \varphi_{in} \end{pmatrix} \qquad (1)$$

where

$$\varphi_{ik} = \alpha_i + \sum_{\ell=1}^{t-1} r_{\ell i} k^{\ell}$$

3) Finally, each player P_k locally interpolates values φ_{ik} by Lagrange interpolation formula in order to recover his updated share α'_k.

$$\alpha'_k = \sum_{1\le i\notin\nabla\le n} \left(\left(\prod_{1\le j\notin\nabla\le n, j\ne i} \frac{j}{j-i} \right) \times \varphi_{ik} \right). \qquad (2)$$

Fig. 5. Share Recovery-and-Renewal in Our PVSS Scheme Where $t < \frac{n}{3}$

Indeed, after detecting the locations of errors, corrupted servers are rebooted. Subsequently, uncorrupted players P_i, where $1 \le i \notin \nabla \le n$, generate new shares of the original secret α for all players P_k, for $i \le k \le n$ while updating their own shares of α. This can be also seen as an *enrollment protocol* in the active adversary setting, that is, t players can generate new shares for other players/newcomers in the absence of the dealer. To the best of our knowledge, this is the first construction of the enrollment protocol in the active adversary setting. Its counterpart, in the passive adversary setting, was first introduced in [14] and [15], and utilized in [16] and [17] for threshold changeability.

Example-2: For the sake of simplicity, we don't use any VSS scheme here but all polynomials must be verifiable. Suppose the secret sharing polynomial is $f(x) = 3 + 2x + x^2 \in \mathbb{Z}_{29}$, where secret $\alpha = 3$ and threshold is $t = 3$. As a result, shares of the players P_1, P_2 and P_3 are $\alpha_1 = 6, \alpha_2 = 11$ and $\alpha_3 = 18$ accordingly. Let assume the random vector $\vec{r} = (r_1 = 4, r_2 = 4)$ is generated by two random polynomials $R_1(x) = 4 + 12x + 15x^2$ and $R_2(x) = 4 + 28x$. As a result, we have:

$$\begin{array}{llllll} P_1 & \text{has} & \alpha_1 & = 6 & r_{11} & = 2 \quad r_{21} = 3 \\ P_2 & \text{has} & \alpha_2 & = 11 & r_{12} & = 1 \quad r_{22} = 2 \\ P_3 & \text{has} & \alpha_3 & = 18 & r_{13} & = 1 \quad r_{23} = 1 \end{array}$$

The first three players P_1, P_2, P_3 generate the following φ_{ik}:

$$\begin{aligned} \varphi_{11} &= 6 + 2(1)^1 + 3(1)^2 = 11 \\ \varphi_{12} &= 6 + 2(2)^1 + 3(2)^2 = 22 \\ \varphi_{13} &= 6 + 2(3)^1 + 3(3)^2 = 10 \\ \varphi_{14} &= 6 + 2(4)^1 + 3(4)^2 = 4 \end{aligned}$$

$$\begin{aligned} \varphi_{21} &= 11 + 1(1)^1 + 2(1)^2 = 14 \\ \varphi_{22} &= 11 + 1(2)^1 + 2(2)^2 = 21 \\ \varphi_{23} &= 11 + 1(3)^1 + 2(3)^2 = 3 \\ \varphi_{24} &= 11 + 1(4)^1 + 2(4)^2 = 18 \end{aligned}$$

$$\begin{aligned} \varphi_{31} &= 18 + 1(1)^1 + 1(1)^2 = 20 \\ \varphi_{32} &= 18 + 1(2)^1 + 1(2)^2 = 24 \\ \varphi_{33} &= 18 + 1(3)^1 + 1(3)^2 = 1 \\ \varphi_{34} &= 18 + 1(4)^1 + 1(4)^2 = 9 \end{aligned}$$

As a result, the share-exchange matrix $\Phi_{i\times n}$ would be:

$$\Phi_{3\times 4} = \begin{pmatrix} \varphi_{11} = 11 & \varphi_{12} = 22 & \varphi_{13} = 10 & \varphi_{14} = 4 \\ \varphi_{21} = 14 & \varphi_{22} = 21 & \varphi_{23} = 3 & \varphi_{24} = 18 \\ \varphi_{31} = 20 & \varphi_{32} = 24 & \varphi_{33} = 1 & \varphi_{34} = 9 \end{pmatrix}.$$

The players then update their shares, e.g., P_1 will calculate his share as follows:

$$\begin{aligned} \alpha'_1 = \frac{2}{(2-1)}\frac{3}{(3-1)} \times 11 \quad &+ \quad \frac{1}{(1-2)}\frac{3}{(3-2)} \times 14 \\ &+ \quad \frac{1}{(1-3)}\frac{2}{(2-3)} \times 20 = 11. \end{aligned}$$

With the same Lagrange constants, the rest of the shares will be updated to $\alpha'_2 = 27, \alpha'_3 = 22$ and $\alpha'_4 = 25$. Now, the original secret $\alpha = 3$ can be recovered from any three shares by Lagrange interpolation formula as follows:

$$\begin{aligned} \alpha = \frac{3}{(3-2)}\frac{4}{(4-2)} \times 27 \quad &+ \quad \frac{2}{(2-3)}\frac{4}{(4-3)} \times 22 \\ &+ \quad \frac{2}{(2-4)}\frac{3}{(3-4)} \times 25 = 3. \end{aligned}$$

End of Example-2.

Theorem: *The proposed proactive verifiable secret sharing scheme (PVSS) is secure in an active mobile adversary setting assuming that the adversary is capable of corrupting at most t players between two consecutive periods.*

Proof: It can be simply shown that our PVSS scheme is independent of the verifiable secret sharing scheme that is being used, therefore, it works under the assumptions of the VSS scheme proposed in [4], that is, considering only the private channels and assuming $t < \frac{n}{3}$.

Secrecy: In the first three steps of the *detection* protocol (Figure 4), each player P_i acts as an independent dealer by using an existing VSS scheme. As a result, not only values α_i-s and β_i-s remain secret but also uncorrupted players will have consistent shares with respect to these secrets at the end of step-3. Note that corrupted players/dealers are disqualified during these stages.

In the last two steps of the *detection* protocol (Figure 4), when the players reveal $\vec{\delta}$, this vector has the same exact errors as of the original shares' vector. As a result, the corrupted players/servers will be detected. In addition, values δ_i do not reveal anything about the players' shares because of the mask value β. Therefore, the corrupted shares are detected without revealing any information with respect to the secret. In other words, values δ_i only reveal secret $\alpha+\beta$ (masked secret) but not the original secret α.

Again, in the first step of the *recovery-and-renewal* protocol (Figure 5), the players utilize an existing VSS scheme. Similar to the previous discussion, not only random values r_i remain secret but also the players will have consistent shares $r_{\ell i}$ with respect to r_i-s; for each player, these shares $r_{\ell i}$ define coefficients of a random polynomial of degree $t-1$ with a zero constant term. As a result, these zero-constant-term polynomials remain secret.

In the last two steps of the *recovery-and-renewal* protocol (Figure 5), the players don't use any VSS scheme, however, each player P_i first re-shares his original share α_i by $\varphi_{ik} = \alpha_i + r_{1i}k + r_{2i}k^2 + r_{3i}k^3 + \cdots + r_{\ell i}k^\ell$. Since shares α_i-s and coefficients $r_{\ell i}$-s are generated by VSS schemes (i.e., there exist strong commitments for these values), a corrupted player P_i will be disqualified if he sends more than t inconsistent shares φ_{ik} to other players P_k. Alternatively, if P_i sends at most $t-1$ inconsistent shares φ_{ik} to other players, P_i will not be disqualified, however, each player P_k, with at most $t-1$ corrupted shares, can locally perform error correction on φ_{ik}-s to recover his correct share α'_k.

Correctness: Let first define the Lagrange constants for at least t players:

$$\gamma_k \stackrel{\text{def}}{=} \prod_{1\le j\notin\nabla\le n, j\neq k} \frac{j}{j-k} \text{ where } 1 \le k \notin \nabla \le n \qquad (3)$$

To recover the secret α, at least t players must combine their new shares α'_k, by Equation (2):

$$\begin{aligned}
\sum_{1\le k\notin\nabla\le n}\left(\gamma_k\times\alpha'_k\right) &= \sum_{1\le k\notin\nabla\le n}\left(\gamma_k\times\sum_{1\le i\notin\nabla\le n}\left(\gamma_i\times\varphi_{ik}\right)\right)\\
&= \sum_{1\le i\notin\nabla\le n}\left(\gamma_i\times\sum_{1\le k\notin\nabla\le n}\left(\gamma_k\times\varphi_{ik}\right)\right)\\
&= \sum_{1\le i\notin\nabla\le n}\left(\gamma_i\times\alpha_i\right) \quad \text{by (1)}\\
&= \alpha
\end{aligned}$$

End of Proof.

Remark-1: It is not hard to show that *threshold changeability*, i.e., changing threshold t to t', can been simply achieved by our share renewal approach. Let $[x]$ denotes the sharing of "x" and let $r_1,\ldots,r_{t'}$ be t' random numbers where $t' \neq t$ is the new threshold. Each P_i sends $\alpha_i + r_{1i}j + r_{2i}j^2 + \cdots + r_{t'i}j^{t'}$ to player P_j, where $r_{\ell i}$ are shares of ℓ random numbers. As a result, each player has a new share on a new secret sharing polynomial $[\alpha]+[r_1]x+[r_2]x^2+\cdots+[r_{t'}]x^{t'}$ of degree t' where its constant term is the original secret α.

Remark-2: It is also not hard to show that *enrollment* and *disenrollment*, i.e., changing n as the number of players, can been simply achieved by our share renewal approach. In other words, if at least t players P_i re-share α_i and send φ_{ik} to a newcomer P_k, as illustrated in Figure 5, the new player obtains his share α'_k by using the Lagrange interpolation formula. On the other hand, if an existing P_k doesn't receive shares φ_{ik}-s from at least t players, he will be disenrolled from the secret sharing scheme.

IV. Conclusion

In this paper, we provided a new protocol for *proactive verifiable secret sharing* (PVSS) in an active mobile adversary setting where $t < \frac{n}{3}$. We utilized a simple *detection* method along with a novel *recovery-and-renewal* technique.

We showed that our solution is more efficient than the only known solution proposed in Asiacrypt'02 [12]. Moreover, we concluded that our recovery-and-renewal protocol can be simply used for threshold changeability, enrollment and disenrollment in an active mobile adversary setting where $t < \frac{n}{3}$. In other words, our scheme can be utilized to construct a new dynamic secret sharing scheme.

V. Acknowledgment

We would like to thank Dr. Douglas Stinson for his helpful comments. We also thank the anonymous reviewers for their feedback and suggestions.

REFERENCES

[1] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[2] G. R. Blakley, "Safeguarding cryptographic keys," in *National Computer Conference*, vol. 48. AFIPS Press, 1979, pp. 313–317.

[3] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," in *26th Annual Symposium on Foundations of Computer Science, FOCS*. IEEE, 1985, pp. 383–395.

[4] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, STOC*, 1988, pp. 1–10.

[5] D. Chaum, C. Crépeau, and I. Damgård, "Multiparty unconditionally secure protocols," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, STOC*, 1988, pp. 11–19.

[6] T. Rabin and M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority," in *Proceedings of the 21th Annual ACM Symposium on Theory of Computing, STOC*, 1989, pp. 73–85.

[7] D. Beaver, "Multiparty protocols tolerating half faulty processors," in *9th Annual International Cryptology Conference, CRYPTO*, ser. LNCS, G. Brassard, Ed., vol. 435. Springer, 1989, pp. 560–572.

[8] D. R. Stinson and R. Wei, "Unconditionally secure proactive secret sharing scheme with combinatorial structures," in *6th Annual International Workshop on Selected Areas in Cryptography, SAC*, ser. LNCS, vol. 1758. Springer, 1999, pp. 200–214.

[9] R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin, "The round complexity of verifiable secret sharing and secure multicast," in *Proceedings of the 33th annual ACM symposium on Theory of computing, STOC*, 2001, pp. 580–589.

[10] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: How to cope with perpetual leakage," in *15th Annual Int. Cryptology Conference, CRYPTO*, ser. LNCS, D. Coppersmith, Ed., vol. 963. Springer, 1995, pp. 339–352.

[11] R. Ostrovsky and M. Yung, "How to withstand mobile virus attacks," in *10th Symposium on Principles of Distributed Computing, PODC*. ACM, 1991, pp. 51–59.

[12] P. D'Arco and D. R. Stinson, "On unconditionally secure robust distributed key distribution centers," in *8th Int. Conference on the Theory and Application of Cryptology and Info. Security, ASIACRYPT*, ser. LNCS. Springer, 2002, pp. 346–363.

[13] E. R. Berlekamp, *Algebraic Coding Theory: Revised Edition*. World Scientific, 2015.

[14] M. Nojoumian, D. R. Stinson, and M. Grainger, "Unconditionally secure social secret sharing scheme," *IET Information Security (IFS), Special Issue on Multi-Agent and Distributed Information Security*, vol. 4, no. 4, pp. 202–211, 2010.

[15] M. Nojoumian and D. R. Stinson, "Brief announcement: Secret sharing based on the social behaviors of players," in *29th ACM Symposium on Principles of Distributed Computing (PODC)*, 2010, pp. 239–240.

[16] M. Nojoumian, "Novel secret sharing and commitment schemes for cryptographic applications," Ph.D. dissertation, Department of Computer Science, University of Waterloo, Canada, 2012.

[17] M. Nojoumian and D. Stinson, "On dealer-free dynamic threshold schemes," *Advances in Mathematics of Communications (AMC)*, vol. 7, no. 1, pp. 39–56, 2013.