Proactive Verifiable Linear Integer Secret Sharing Scheme

Chuangui Ma and Xiaofei Ding

Zhengzhou Information Science and Technology Institute,
Zhengzhou, 450002, China
chuanguima@yahoo.com

Abstract. The commonly used technique for cheating detection in verifiable secret sharing (VSS) require public key systems. Based on linear integer secret sharing (LISS) scheme, this paper presents a private verifiable protocol over arbitrary access structure without public key systems, which can avoid cheating both from participants and dealers. For further consideration of share refreshing and renewal, this paper shows the proactive property of our scheme with new method. Furthermore, this paper applies combinatorial structure into the proactive scheme to reduce the time of the computation.

Keywords: LISS; refresh; renewal; proactive property; combinatorial structure.

1 Introduction

One important topic in cryptography is how to securely share a secret among a group of people. In a secret sharing scheme, a *dealer* distributes the secret to a number of shareholders, such that only qualified sets can reconstruct the secret, while other subsets have no information about it. It is a fundamental building block for many cryptographic protocols and is often used in the general composition of secure multiparty computations. The collection of consisting of qualified sets is called the access structure.

Blackley, G.R.[1] and Shamir, A. [2] independently introduce the first Secret Sharing in 1979, which store critical information such that we get at the same time protection of privacy and security against loosing the information. Later, secret sharing has proved extremely useful, not just as a passive storage mechanism, but also as a tool in interactive protocols. So it's a important and useful tool to make a good secret sharing scheme.

Linear integer secret sharing (LISS) was introduced by Damgard, I. and Thorbek, R. [3]. In LISS scheme, the secret was an integer chosen from a (publicly known) interval, and each share was computed as an integer linear combination of the secret and some random numbers chosen by the dealer. Reconstruction of the secret was also by computing a linear combination with integer coefficients of the shares in a qualified set.

S. Qing, C.J. Mitchell, and G. Wang (Eds.): ICICS 2009, LNCS 5927, pp. 439–448, 2009.

Based on the concept of an integer span program (ISP) introduced by Cramer et al. shown that any ISP could be used to build a private secure LISS scheme. The details could be find in [4,5]. Private security and perfect security were different concepts in some cases and they were shown in [6,7] that perfect secret sharing and private computation over countably infinite domains (like the integers) were not possible. However, this didn't rule out schemes of this type since the secrets were chosen from a publicly known interval, and protocols were proved to be statistical security rather than perfect privacy.

Another security aspect in secret sharing schemes is cheating prevention. There are two ways to do this. One method uses longer shares as in [21]. The other requires extra information to verify the shares of the shareholders. There are many papers investigate such schemes[10,11]. The verifiable secret sharing schemes depended on some cryptographic assumptions. In Pedersen's scheme, the privacy of the secret was unconditionally secure, but the correctness of the shares based on a computational assumption.

The third security consideration is share refreshing and renewal. In some occasions, a secret value (for example a cryptographic master keys, data files and legal documents) should to be stored for a long time. In this case, an adversary attacked the locations one by one and eventually got the secret or destroyed it. To resist such attack, proactive secret sharing schemes were proposed. Proactive security for secret sharing was first suggested by Ostrovsky, R. and Yung, M. [8]. Their paper presented a proactive polynomial secret sharing scheme. Proactive security refers to security and availability in the presence of a mobile adversary. Herzberg, A. et al. [9] specialized this notion to robust secret sharing schemes and gave a efficient proactive secret sharing scheme.

Our contribution are three fold: the first is verifiable, the second is proactive and the third optimize. In this paper a new proactive secret sharing scheme is proposed. Shares are periodically renewed without changing the secret. Every participant is able to verify the share which he receives and those other participants show. This scheme can prevent adversaries from getting the secret or sharing and the participants cheating from each other efficiently.

Moreover, we introduce some combinatorial structures [12] in the scheme so that the scheme will be more efficient. With uses of combinatorial structures, we can obtain a predetermined arrangement of the servers which permits the possibility of reducing the computation of the scheme. Our scheme is more efficient in the situation when the number of the possible corrupted servers are much smaller as compared to the total number of the servers in the system.

The remainder of this paper is organized as follows. In Section 2 we give some preliminaries and recall the LISS scheme. Section 3 describes our Verifiable LISS scheme. In Section 4 we provide a Proactive secret sharing. Section 5 gives an security analysis. Section 6 introduces Verifiable LISS scheme with combinatorial structure and analyzes the efficiency of our scheme briefly. Finally, conclusions is presented in Section 7.

2 Secret Sharing Scheme

2.1 Preliminaries

First we describe the definition of the access structures from [5].

Definition 1. *[5] A monotone access structure on* $\{1,\ldots,n\}$ *is a non-empty collection* Γ *of sets* $A \subseteq \{1,\ldots,n\}$ *such that* $\emptyset \notin \Gamma$ *and such that for all* $A \in \Gamma$ *and for all sets* B *with* $A \subseteq B \subseteq \{1,\ldots,n\}$*, it holds that* $B \in \Gamma$*.*

In the following we define the notion of an Integer Span Program (ISP, introduced in [5]) and show how any ISP can be used to build a correct and private LISS scheme in [13].

Definition 2. *[5]* $\mathscr{M} = (M, \psi, \varepsilon)$ *is called an Integer Span Program (ISP), if* $M \in Z^{d\times e}$ *and the* d *rows of* M *are labelled by a surjective function* $\psi : \{1,\ldots,d\} \to \{1,\ldots,n\}$*. Finally,* $\varepsilon = (1,0,\ldots,0)^T \in Z^e$ *is called the target vector. We define size* $(\mathscr{M}) = d$*, where* d *is the number of rows of* M*.*

Definition 3. *[13] Let* Γ *be a monotone access structure and let* $\mathscr{M} = (M, \psi, \varepsilon)$ *be a integer span program. Then* $\mathscr{M}$ *is an ISP for* Γ*, if for all* $A \subseteq \{1,\ldots,n\}$ *the following holds.*
1. If $A \in \Gamma$*, then there is a vector* $\lambda \in Z^d$ *such that* $M_A^T \lambda = \varepsilon$*.*
2. If $A \notin \Gamma$ *, then there exists* $\kappa = (\kappa_1,\ldots,\kappa_e)^T \in Z^e$ *such that* $M_A \kappa = 0 \in Z^d$ *with* $\kappa_1 = 1$*, which is called the sweeping vector for* A*.*

Then we review the definition of Verifiable Secret Sharing protocol in [14].

Definition 4. *[14] A player execute protocol honestly is called a good player.*

Definition 5. *[14] The protocol* π *is an private Verifiable Secret Sharing protocol if the following properties are hold:*

1.If a good player P_i *outputs* $ver_i = 0$ *at the end of Share then every good player outputs* $ver_i = 0$ *;*

2.If the dealer is good, then $ver_i = 1$ *for every good* P_i *;*

3.If at least $n - b$ *players* P_i *output* $ver_i = 1$ *at the end of Share, then there exists an* $s' \in S$ *such that the event that all good* P_i *outputs* s' *at the end of Reconstruct is fixed at the end of Share and* $s' = s$ *if the dealer is good ;*

4.For any two secrets s, s'*, any forbidden set* A *of shareholders. Set vector* $r_i = M_i\rho$ *for* $i = 1,\ldots,t-1$*,* $r_i \in [0\ldots 2^l]$ *we can find symmetrical matrix* ρ' *, set* $r_i = M_i\rho'$ *, where* r, r' *are statistically indistinguishable. More precisely, the statistical distance between the two distribution is negligible in* k*.*

2.2 Linear Integer Secret Sharing

The materials of this subsection come from [13].

Let $P = \{1,\ldots,n\}$ denote the n shareholders (or players) and D as the dealer. The dealer D wants to share a secret s from the publically known interval

$[0 \ldots 2^l]$ to the shareholders P over Γ. such that every set of shareholders $A \in \Gamma$ can reconstruct s, but such that a set of shareholders $A \notin \Gamma$ gets no or little information on s.

There is an adversary which can corrupt at most b servers at most during any time period. Corrupting a server means learning the secret information of the server, modifying its data, sending out wrong message, changing the intended behavior of the server, disconnecting it, and so on. Since the server can be rebooted , the adversary is a mobile one.

A secret value $s \in GF(q)$ will be shared by the servers through the scheme. The value of s needs to be maintained for a long period of time. The life time is divided into time periods which are determined by the global clock. At the beginning of each time period the servers engage in an interactive update protocol. The update protocol will not reveal the value of s. At the end of the period the servers hold new shares of s. The mobile adversary who corrupts b servers in a time period cannot get any information about the secret value s. The system can reproduce s in the presence of the mobile adversary at any time.

Share

We use a distribution matrix $M \in Z^{d \times e}$ and a distribution vector $\rho = (s, \rho_2, \ldots, \rho_e)^T$, where s is the secret, and the ρ_i^s are uniformly random chosen integers in $[0 \ldots 2^{l_0+\kappa}]$ for $2 \leq i \leq e$, where κ is a security parameter and l_0 is a constant that is part of the description of the scheme. The dealer D calculates shares by

$$M \cdot \rho = (s_1, \ldots, s_d)^T$$

where we denote each s_i as a share unit for $1 \leq i \leq d$. Let $\psi : \{1, \ldots, d\} \to P$ be a surjective function. The i'th share unit is then given to the i'th shareholder, we say that $\psi(i)$ owns the i'th row in M. If $A \subseteq P$ is a set of shareholders, then M_A denotes the restriction of M to rows jointly owned by A. We denote d_A as the number of rows in M_A. Similarly, for $s \in Z^d$ let $s_A \in Z^{d_A}$ denote the restriction of s to the coordinates jointly owned by A.

Reconstruct

For a qualified set A, there is $\lambda_A \in Z^{d_A}$ which gives $M_A \kappa = 0 \in Z^{d_A}$,

$$s_A^T \lambda_A = (M_A \rho)^T \cdot \lambda_A = \rho^T \cdot (M_A^T \cdot \lambda_A) = \rho^T \cdot \varepsilon = s$$

From [3] we know that the LISS scheme is correct and private. But secret sharing have two problems: one is the security of initialization, although we think *Dealer* as a trusty center in many instances, there are many unpredictable factors practicality, which is difficult to guarantee transmission correctly in network and malicious attack dealer. Another aspect is efficiency of share units, i.e. how to ensure the correctness of share units when transmitted from each other. The next section will show the method to solve these problems.

3 Verifiable Secret Sharing

To prevent cheating behaviors among secret sharing and recovering. we present a verifiable LISS scheme with a different method from traditional verifiable schemes.

Before reconstructing secret, combiner reconstructor first validate share units from other participators. The following is the detail protocol.

1. *Dealer* random construct a symmetrical matrix $\rho = (\rho_0, \ldots, \rho_{e-1}) = (\rho_{ij})_{e\times e}$,where $\rho_{00} = s$, M is matrix of ISP. Let $M\rho_{00} = (s_{01}, \ldots s_{0d})$ be share units and $M\rho_i$, for $i = 1, \ldots e-1$ as public verifiable vectors.
2. P_i sends $s_{\psi(i)} = M_{\psi(i)}\rho_0$ to P_j,
3. After receiving $M_{\psi(i)}\rho_0$, P_j checks whether $M_{\psi(i)}\rho M_{\psi(j)}$=$M_{\psi(j)}\rho M_{\psi(i)}$, if P_j finds that $M_{\psi(i)}\rho M_{\psi(j)} \neq M_{\psi(j)}\rho\ M_{\psi(i)}$, P_j broadcasts(i, j).
4. Each P_i computes the maximum subset $G \subseteq \{1, \ldots, n\}$ such that any ordered pair $(i, j) \in G \times G$ is not broadcasted, If $|G| \geq n - b$, then P_i outputs $ver_i = 1$; otherwise, P_i outputs $ver_i = 0$, and requires *Dealer* repeat share secret.

It is obvious that every good player computes the same subset G in the end of *share*. The reconstruct phase that is the same as stated above.

4 Proactive Secret Sharing

It is dangerous for long live periodic secret. The most efficient method is processing. This section introduces share renewal in period to protect this kind of secret. So in this situation, we can divide life time into time periods: mark the length of each time period as t, time of share renewal at beginning and end phase,and keep secret changeless after share renewal.

4.1 Share Renewal Protocol

Each P_i for $1 \leq i \leq n$ random choose vector $\rho_i = (0, \rho_{i2}, \ldots, \rho_{ie})$, calculates shares by $M\rho_i = (s_{i1}, \ldots, s_{id})$, where s_{ij} is given to $p_{\psi(j)}$, i.e. $p_{\psi(j)}$ receive $(s_{1j}, \ldots, s_{nj})$, renewal share as $s_j^t = s_j^{t-1} + s_{1j} + \ldots + s_{nj}$.

Reconstruct secret: For a qualified subset A we have that

$$(s_A^t)^T\lambda_A = (s_1^t, \ldots, s_{d_A}^t)^T\lambda_A = (s_1^{t-1} + s_{11} + \ldots + s_{n1}, \ldots, s_{d_A}^{t-1} + s_{1d_A} + \ldots + s_{nd_A})^T\lambda_A = (s_1^{t-1}, \ldots, s_{d_A}^{t-1})\lambda_A + (s_{11} + \ldots + s_{n1}, \ldots, s + 1d_A + \ldots + s_{nd_A})\lambda_A = s$$

4.2 Detection of Corrupted Shares

In the proactive secret sharing system, users must be able to ensure that shares of other users have not been corrupted or lost, and be able to restore the correct shares if necessary. Otherwise, an adversary could cause the loss of the secret by destroying shares. This subsection presents a mechanism for detection of corrupted shares.

The idea is to save some fingerprint for each share that is common to all the shareholders, so that periodically, shareholders can compare shares (using secure broadcast). In order to implement the distributed verifiability of shares, a basic feature is added to the previous protocol. In each time period, each user stores the encryptions of all the shares he/she received from the other users.

Definition 6. *1. P_i sends $M_{\psi(i)}\rho$ to P_j, $j = 1, \ldots, n, j \neq i$;*

2. P_j checks whether $M_{\psi(i)}\rho M_{\psi(j)} = M_{\psi(j)}\rho M_{\psi(i)}$, then P_j broadcasts an accusation $list_j$ which contains those i such that $M_{\psi(i)}\rho M_{\psi(j)} \neq M_{\psi(j)}\rho M_{\psi(i)}$ or $M_{\psi(i)}\rho$ was not received.

3. Each good player updates the list $\mathcal{L}$ so that it contains those i accused by at least $b+1$ players of the system.

4.3 Recovery of Lost/Corrupted Shares

This is a fundamental phase in the proactive scheme, because without it, this scheme would not be secure against adversaries who disable some users from performing the required protocol.

After running *detection*, the system will recover the shares for all players P_l, where $l \in \mathcal{L}$. The recovery protocol is as follows.

1. For each $l \in \mathcal{L}$, every good players P_i sends $M_{\psi(i)}\rho_0^k$ to P_l;

2. Upon receiving the data, P_l computes $M_{\psi(l)}\rho^k M_{\psi(i)} = M_{\psi(i)}\rho^k M_{\psi(l)}$, P_l sets $M_{\psi(l)}\rho_0^k + M_{\psi(l)}\rho_0$ as its shares.

5 Security Analysis

The private and verifiable of our scheme can be illuminated from the follow theorem.

Theorem 1. *The above LISS scheme is private and Verifiable.*

Proof. We prove that the above scheme satisfies the conditions of the definition 5 as follows:

1. If a good player P_i outputs $ver_i = 0$, then the size of the maximum subset G is at most $n - b - 1$. Thus every good player will output 0;

2. If the dealer is good, then good player receives $M_{\psi(i)}\rho^0$ .Since ρ is a symmetric matrix, $M_{\psi(i)}\rho M_{\psi(j)} = M_{\psi(j)}\rho M_{\psi(i)}$ for all good players P_j, Thus all good players are in the subset G . Therefore $ver_i = 1$ for every good P_i ;

3. Suppose at least $n - b$ players output "1" at the end of the Share. Then there is a subset G of size $n - b$ such that no one in the subset complained the others. Since we assume that there at most b bad players , there are at least $n - 2b$ good players in G, in which who all of them have consistent shares. Thus there is a qualified set A, $\lambda_A \in Z^{d_A}$, st:

$$s' = s_A^T \lambda_A = (M_A \rho)^T \cdot \lambda_A = \rho^T \cdot (M_A^T \cdot \lambda_A) = \rho^T \cdot \varepsilon = s$$

4. For arbitrary s We have chose $\rho = (\rho_0, \rho_1, \ldots, \rho_{e-1})$, with $\rho_{00} = s$, $\rho_{ij} \in [0 \ldots 2^{l_0+\kappa}]$ as uniformly random numbers, and secret $s \in [0 \ldots 2^l]$.

Let $s' \in [0 \ldots 2^l]$ be arbitrary, We first observe that $s_A = M_A \rho$ is the subset of shares that belongs to A. If A is a forbidden set, there exists a sweeping vector κ , such that $M_A \kappa = 0 \in Z^{d_A}$.

Define $\rho' = (\rho_0 + (s' - s)\kappa, \rho_1, \ldots, \rho_{e-1})$, then we have $M\rho = M\rho'$, that is the shareholders in A see the same shares, but the secret s' was shared instead of s. Define ρ is good if $\rho' = (\rho_0 + (s' - s)\kappa, \rho_1, \ldots, \rho_{e-1})$ has entries in the specified range, as mean to $\rho' \in [0 \ldots 2^{l_0+\kappa}]$. That request each ρ_{0i} for $i = 1, \ldots . e-1$ satisfied :

$$|\rho_i| + 2^l \cdot \kappa_{max} \leq 2^{l_0+\kappa}$$

where $k_{max} = \max\{|a| : \text{a is an entry in some sweeping vector}\}$

So the probability that a ρ_{0i} is not good is :

$$1 - \frac{2^{l_0+\kappa} - 2^l \cdot \kappa_{max}}{2^{l_0+\kappa}} = \frac{2^l \cdot \kappa_{max}}{2^{l_0+\kappa}}$$

It follows that the statistical distance between the distribution of A's shares of s and s' is at most twice the probability that ρ is not good. Which we can estimate by the union bound as $e-1$ times the probability that a single entry is out of range. So $|s' - s| \leq 2^l$. the distance is at most

$$2 \cdot \frac{2^l k_{max}(e-1)}{2^{l_0+k}} \leq 2^{-k}$$

Now the Theorem 1 holds.

6 Optimization of Our Scheme

In this section, we will introduce combinatorial structure [12,14] into our scheme. The combinatorial structure provides a predetermined arrangement of the servers which permits the possibility of reducing the computation of the scheme. This method optimizes our scheme apparently.

6.1 Set Systems

A set system is a pair $(X, \mathcal{B})$, where X is a set of n points and $\mathcal{B}$ is a collection of subsets of X called blocks.

We will use a set system with the following properties:

1. $|B| \geq t$ for any $B \in \mathcal{B}$;
2. For any subset $F \subseteq X$ with $|F| \leq b$, there exists a $B \in \mathcal{B}$ such that $F \bigcap B = \emptyset$. where $t \leq \frac{n}{4} - 1$

It is easy to see that such a set system exists.

Definition 6. A collection $\mathcal{T}$ of k-subsets of $\{1, \ldots, n\}$ (called blocks) is an (n, k, b)-covering if every b-subset of $\{1, \ldots, n\}$ is contained in at least one block.

Theorem 2. A set system $\mathcal{T}$ satisfies above properties:1,2 if and only if the set system

$$\{1,\ldots,n\}\backslash T : T \in \mathcal{T}$$

is an $(n, n-t, b)$

It is easy to see that if $(X, \mathcal{B})$ is an $(n, n-k, b)$-covering, then the set system

$$\{1,\ldots,n\}\backslash T : T \in \mathcal{T}$$

is a set system satisfying our purpose.

6.2 Applying Set System to the Verifiable LISS

The idea of using the set system is to reduce the computations for the share renewal and recover protocol. In the Section 4, share renewal and recover used the data from all the participants. However, these operations can be carried out using the data from good players. So there are redundant computations. On the other hand, we should be very careful when the good players are selected, since the adversary is mobile. The good player could turn to bad at any time. Thus in the scheme of this section, we will actually select correct information instead of good servers, although we will use "good player" for convenience.

Now let us use the set system to improve our LISS scheme. Suppose $(X, \mathcal{B})$ is a set system satisfying the condition of subsection 4.1, where $X = \{1,\ldots,n\}$, and $\mathcal{B} = \{B_1,\ldots,B_s\}$. The set system is published so that each participant can consult it.

Note that in our scheme, in any phase there is a list $\mathcal{L}$ containing all the bad players. By the property of the set system, there is a block B which contains only good players. If the system can determine one of the "good" blocks, then the system can renew the shares or recover the shares only using the data from these players. We will call these players as the members of an *executive committee*.

For a list $\mathcal{L}$ of bad players, the system can decide following list of blocks: $B_{i1},\ldots,B_{ie}$, such that $B_{ij} \cap \mathcal{L} = \emptyset, j = 1,\ldots,e$. These blocks are called *executive committee candidates*. Note that the adversary is mobile, therefore we cannot guarantee that these candidates contain only good servers in the next time period.

The verifiable LISS scheme with combinatorial structure works. In each time period the system dose the following:

1. Run the *Detection* to obtain the list $\mathcal{L}$ of bad players and the executive committee candidates: $B_{i1},\ldots,B_{ie}$;

2. If an *executive committee* has not been found, then for next *executive committee candidates* B, each $P_k \in B$ dose the following:

(1) Each P_k selects a random symmetric matrix $\rho^k = (\rho_0^k,\ldots,\rho_{e-1}^k)$, where $\rho_{ij}^k = \rho_{ji}^k$, and $\rho_{00}^k = 0$, M is matrix of ISP. We send $M\rho_0^k$ as shares, $M\rho_i^k$, $i = 1,\ldots,e-1$ as public verifiable vectors;

(2) P_k sends $M_{\psi(k)}\rho_0^k$ to P_m, $m = 1,\ldots,n$. P_m checks whether $M_{\psi(k)}\rho^k M_{\psi(m)} = M_{\psi(m)}\rho^k M_{\psi(k)}$, If the conditions are not satisfied, P_m broadcasts an accusation of P_k.

(3) A member in B is accused by at least $b+1$ players is bad. If a member in B is accused by at most b players, then it can defend itself. If no member in B is bad, then B is found to be the *executive committee*.

3. The system runs the *recovery* protocol to recover the shares for the players in $\mathcal{L}$;

4. Each player $P_k \in B$ updates its shares:

$$M_{\psi(k)}\rho_0^k + M_{\psi(k)}\rho_0$$

The reconstruction protocol is the same as in the Section 3.

6.3 Performance Evaluation

In this subsection, we analyze the efficiency of our scheme more clearly and explicitly. We claim the scheme is more efficient, according to the two reasons as follows:

First reason, the traditional Verifiable secret sharing scheme used exponentiation with pubic key, the cost of computation is tremendous. This paper introduces a proactive verifiable LISS protocol, whose computation is only the level of integral multiplication. So this method improves efficiency in the practical application.

The second reason is that we apply combinatorial structure into our scheme. The combinatorial structure provides a predetermined arrangement of the servers, which reduce the computations for the share renewal and share recover protocol. From comparing the difference and advantage between our proposal and previous scheme, our scheme is more efficient.

7 Conclusion

In this paper, we propose a proactive verifiable Linear Integer Secret Sharing protocol which improves efficiency in the practical application. Then we describe the process of shares renewal and recovery carefully and prove it correct, private and verifiable. We also scheme out a verifiable LISS scheme with combinatorial structure which makes the scheme more efficient. Finally, we give the performance evaluation of this scheme.

Acknowledgement

This work is supported by a grant from the National High Technology Research and Development Program of China (No. 2007AA01Z431). The authors would like to thanks the anonymous referees for their helpful comments.

References

1. Blackley, G.R.: Safeguarding cryptographic keys. In: AFIPS 1979, pp. 313–317 (1979)
2. Shamir, A.: How to Share a Secret. Commun. ACM 22(11), 612–613 (1979)
3. Damgard, I., Thorbek, R.: Linear Integer Secret Sharing and Distributed Exponentiation. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T.G. (eds.) PKC 2006. LNCS, vol. 3958, pp. 75–90. Springer, Heidelberg (2006)

4. Benaloh, J.C., Leichter, J.: Generalizad Secret Sharing and Monotone Functions. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 27–35. Springer, Heidelberg (1990)
5. Cramer, R., Fehr, S.: Optimal Black-Box Secret Sharing over Arbitrary Abelian Groups. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 272–287. Springer, Heidelberg (2002)
6. Chor, B., Kushilevitz, E.: Secret Sharing Over Infinite Domains. In: Cryptology 1993, vol. 6(2), pp. 87–95 (1993)
7. Chor, B., Mihály Geréb, G., Kushilevitz, E.: Private Computations over the Intergers. SIAM J. Comput. 24(2), 376–386 (1995)
8. Ostrovsky, R., Yung, M.: How to withstand mobile virus attacks. In: ACM Symposium on principles of distributed computing 1991, pp. 51–59 (1991)
9. Herzberg, A., Jarecki, S., Krawczyk, H., Yung, M.: Proactive secret sharing or: How to cope with perpetual leakage. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 339–352. Springer, Heidelberg (1995)
10. Feldman, P.: A Practical Scheme of Non-Interactive Verifiable Secret sharing. In: 28th Annual Symp. on the Foundations of Computing Science 1987, pp. 427–437 (1987)
11. Pedersen, T.P.: Non-interactive and information-theoretic secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)
12. Rees, R.S., Stinson, D.R., Wei, R., Rees, G.H.J.V.: An application of covering designs:determining the maximum consistent set of shares in a threshold scheme. In: Ars Combinatoria 1999. LNCS, vol. 531, pp. 225–237. Springer, Heidelberg (1999)
13. Damgard, I., Thorbek, R.: Linear Integer Secret Sharing and Distributed Exponentiation(full version). The Eprint archive, `http://www.iacr.org`
14. Stinson, D.R., Wei, R.: Unconditionally Secure Proactive Secret Sharing Scheme with Combinatorial Structures. In: Heys, H.M., Adams, C.M. (eds.) SAC 1999. LNCS, vol. 1758, pp. 200–214. Springer, Heidelberg (2000)
15. Alon, N., Gail, Z., Yung, M.: Efficient dynamic-resharing "verfiable secret sharing" against mobile adversary. In: Spirakis, P.G. (ed.) ESA 1995. LNCS, vol. 979, pp. 523–537. Springer, Heidelberg (1995)
16. Boppana, R.B.: Amplification of Probabilistic Boolean Formulas. In: Advances in Computing Research 1989, pp. 27–45 (1989)
17. Santis, A.D., Desmedt, Y., Frankel, Y., Yung, M.: How to share a function securely. In: STOC 1994, pp. 522–533 (1994)
18. Shoup, V.: Practical Threshold Signatures. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 207–220. Springer, Heidelberg (2000)
19. Gordon, D.M., Kuperberg, G., Patashnik, O.: New constructions for covering design. J. Combin. Designs 3, 269–284 (1995)
20. Stinson, D.R.: Cryptography Theory and Practice. CRC Press, Inc., Boca Raton (1995)
21. Ghodosi, H., Pieprzyk, J.: Cheating Prenvention in Secret Sharing. In: Clark, A., Boyd, C., Dawson, E.P. (eds.) ACISP 2000. LNCS, vol. 1841, pp. 328–341. Springer, Heidelberg (2000)