Chapter 3

Extended Capabilities

3.1 Introduction

In this chapter, we explore the extended capabilities to be considered for the development of secret sharing schemes. Most of the secret sharing constructions assume that the Dealer is a trusted entity and the shares distributed are consistent. But an untrusted Dealer may send invalid shares and the secret reconstructed will be inconsistent. Verifiable Secret Sharing (VSS) address this issue. In this, the participant can verify the validity of the shares. A Publicly Verifiable Secret Sharing (PVSS) allows not only the participant but any one will be able to check the validity of the shares send by the Dealer. There are also dealer free secret sharing scheme to avoid the assumption of a trusted Dealer. Another requirement is that, the scheme must be able to identify the cheaters. The participant may submit wrong shares during the reconstruction phase and all other participants except the cheater will get wrong secret. So secret sharing schemes designed should have the capability to detect and identify the cheaters. Robust secret sharing and cheating immune secret sharing schemes ensures that cheater will not get any advantage in the
reconstruction protocol. When a secret share is compromised, it will affect the security of the secret over the life time of the secret. Proactive methods will update the secret shares periodically so that even if the attacker has a share, it will be invalid after some time. The following sections of the chapter will discuss about the extended capabilities in detail.

3.2 Verifiable Secret Sharing

In a secret sharing scheme the Dealer is assumed to be reliable. However a misbehaving Dealer may send inconsistent shares to the participants. To prevent such malicious behavior of the Dealer, protocols need to be implemented which allows the participant to verify the consistency of the shares. Verifiable Secret Sharing (VSS) is to convince shareholders that their shares are consistent. In Shamir’s \((t,n)\) threshold scheme, the participants can verify that their shares are \(t\)-consistent. This means that every subset of \(t\) shares out of \(n\), if used to interpolate a polynomial will get a unique polynomial of degree \(t - 1\).

**Definition 3.2.1.** Set of \(n\) shares \(S_1, S_2, \ldots, S_n\) is \(t\) consistent, if every subset of \(t\) of the \(n\) shares defines the same secret. The problem of verifiable secret sharing is to convince shareholders that their shares (collectively) are \(t\) consistent.

The concept of Verifiable Secret Sharing (VSS) was first introduced in 1985 by Benny Chor, Shafi Goldwasser, Silvio Micali and Baruch Awerbuch [53]. Application of secret sharing homomorphism to verifiable secret sharing is addressed by Benaloh [17]. There are two versions of verifiable secret sharing protocols, **interactive** proofs and **non interactive** proofs. Chor et al and Benaloh schemes are interactive,
3.2. Verifiable Secret Sharing

which need several rounds of interaction between users and Dealer. Feldman [71] has proposed a non interactive scheme. Both the scheme achieve verifiability in the Shamir’s threshold scheme. Verifiable secret sharing and multi party protocols are addressed by Rabin et al [176]. Pedersen [167] proposed a non-interactive and information theoretic secure verifiable variant of Shamir’s threshold scheme. There have been two different approaches to achieve VSS by a CRT-based secret sharing scheme. Qiong et al [172] proposed VSS scheme based on Asmuth-Bloom secret sharing. Their approach is similar to the VSS of Pedersen [167] based on Shamir’s secret sharing scheme. The second one, proposed by Iftene [104] obtains a VSS scheme from Mignotte’s scheme [146] which is another CRT-based secret sharing scheme similar to Asmuth-Bloom. Both the scheme are not secure against attacks. Kaya et al [118] proposed a more secure scheme based on CRT. They also proposed a Joint Random Secret Sharing (JRSS) protocol, which enable a group of users to jointly generate and share a secret, where a trusted Dealer is not available. A verifiable secret sharing scheme based on Azimuth-Bloom without making a computational assumption is proposed by Harn et al [90].

3.2.1 Interactive Proof-Benaloh

In Shamir’s scheme, the shares $S_1, S_2, \ldots, S_n$ are $t$-consistent if and only if the interpolation of the points $(1, S_1), (2, S_2), \ldots, (n, S_n)$ yields a polynomial of degree at most $d = t - 1$. It is also true that if the sum of two polynomials is of degree at most $d$, then either both are of degree at most $d$ or both are of degree greater than $d$. A polynomial $P$, given by its encrypted values at $n$ distinct points is of degree at most $d$. The following is an outline of the interactive proof

1. Encryption of the values of the points that describe $P$ are released by the prover.
2. Encryption of many (say 100) additional random polynomials again of degree at most \( d \) are also released by the prover.

3. A random subset of the random polynomials is designated by the verifier(s).

4. The polynomials in the chosen subset are decrypted by the prover. They must all be of degree at most \( d \).

5. Each remaining random polynomial is added to \( P \). Each of these sum polynomials is decrypted by the prover. They must also all be degree of at most \( d \).

The encryption of the values of each point must be probabilistic and should satisfy the homomorphism property so that sum of the two values can be developed directly from the encryption of the two values. It is not hard to see that a set of random polynomials of degree at most \( d \) together with a set of sums of \( P \) and other random polynomials of degree at most \( d \) gives no useful information about \( P \) other than its bounded degree \( d \).

There are few drawbacks to interactive proofs

- The interactive proof asserts proof only to the participants of this protocol at the time it is held. The proof have no meaning for the person who is not online and does not participate in the random selections.

- These proofs are not valid to a third party and hence cannot have a legal proof in court.

- Communication complexity is exponential.
3.2.2 Non Interactive Schemes

In the Non-interactive proof scheme, only the Dealer is allowed to send messages. The share holders cannot send any information with each other. The share holders are also not allowed to talk with the Dealer when verifying a share. The basic technique in Non-interactive scheme is that the Dealer sends extra information to each participant during the distribution of shares and each participant can verify whether his share is consistent with this extra information. The additional requirement is that the encryption algorithm $E$ should have the homomorphic property both with respect to addition and multiplication. That is $E(x + y) = E(x) + E(y)$ and $E(x \cdot y) = E(x) \cdot E(y)$. Diffie-Hellman encryption algorithm satisfies this property. This scheme is secure only for computationally bounded adversaries. It leaks some information about the secret.

Feldman’s Scheme

The protocol proposed by Feldman [71] is as follows:

- First a cyclic group $G$ of prime order $p$ along with a generator $g$ of $G$ is chosen publicly as a system parameter. The group $G$ must be chosen such that computing discrete logarithms is hard in this group (Typically one takes a subgroup of $\mathbb{Z}_q^*$, where $q$ is a prime such that $q$ divides $p - 1$).

- The Dealer generates a random polynomial $q(x)$ of degree $t - 1$, $q(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1}$, where $a_0$ is set as secret $S$.

- The Dealer distribute shares to each participant $q(1), q(2), \ldots, q(n)$. In addition the Dealer also publishes the encryption of $t$ coefficients $E(a_0), E(a_1), \ldots, E(a_n)$ to make the shares verifiable.
Chapter 3. Extended Capabilities

\[ c_0 = E(a_0) = g^{a_0} \]
\[ c_1 = E(a_1) = g^{a_1} \]
\[ \vdots \]
\[ c_{t-1} = E(a_{t-1}) = g^{a_{t-1}} \]

• user \( i \) can verify the shares by testing

\[ c_0 c_1 i c_2 i^2 \cdots c_{t-1} i^{t-1} = \prod_{j=0}^{t-1} c_j^{i^j} = \prod_{j=0}^{t-1} g^{a_j i^j} = g^{\sum_{j=0}^{t-1} a_j i^j} = g^{q(i)} \]

Example 3.2.1. Let \( q(x) = 5 + 2x + 1x^2 + 2x^3 \), secret \( S = a_0 = 5, a_1 = 2, a_2 = 1, a_3 = 2, n = 7 \). The shares are \( q(1) = 10, q(2) = 29, \ldots, q(7) = 754 \). The encryption of the coefficients are \( E(a_0) = g^5 \) (mod \( p \)), \( E(a_1) = g^2 \) (mod \( p \)), \( E(a_2) = g^1 \) (mod \( p \)), \( E(a_3) = g^2 \) (mod \( p \)). Suitable \( p \) must be chosen. User 2 verifies the share by checking
\[ E(q(2)) = g^{29} \text{ (mod } p\text{) is equal to} \]
\[ E(a_0 + (a_1 \times 2^1) + (a_2 \times 2^2) + (a_3 \times 2^3)) = g^{5+4+4+16} = g^{29} \text{ (mod } p\text{)} \]

Benaloh’s scheme [17] relied on the existence of mutually trusted entity. In Feldman’s scheme [71] this entity is avoided by letting the Dealer publish probabilistic encryptions of the polynomial used to compute the shares. The homomorphism property of the encryption scheme make verification of the shares possible. This scheme is quite efficient, but after the distribution of the shares with verification capability, the privacy of the secret depends on the computational assumptions such as the intractability of computing discrete logarithms. If \( g \) is the generator of the group then \( g^S \) is known where \( S \) is the secret.
Pedersen [167] in 1992 developed a scheme which is unconditionally secure in which he removes the assumption that $g^S$ is known. However in this scheme the Dealer can succeed in distributing incorrect shares, if he can solve the discrete logarithm problem. The scheme is constructed by combining Shamir’s scheme with a commitment scheme, which is unconditionally secure for the committer and furthermore allows commitment to many bits simultaneously. Pedersen’s scheme is mentioned below.

**Pedersen’s Scheme**

Let $p$ and $q$ denote large primes such that $q$ divides $p - 1$, $G_q$ is the unique subgroup of $\mathbb{Z}_p^*$ of order $q$, and $g$ is the generator of $G_q$. If an element $a \in \mathbb{Z}_p^*$ is in $G_q$ since

$$a \in G_q \iff a^q = 1$$

Any element $b \neq 1$ in $G_q$ generates the group. The discrete logarithm of $a \in G_q$ with respect to the base $b$ is defined and it is denoted $\log_b(a)$.

The commitment scheme proposed is as follows:

Let $g$ and $h$ be elements of $G_q$ such that nobody knows $\log_g(h)$. These elements can either be chosen by a trusted center, when the system is initialized or by some of the participants using coin-flipping protocol. The committer commits himself to an $S \in \mathbb{Z}_q$ by choosing $t \in \mathbb{Z}_q$ at random and computing

$$E(S, t) = g^S h^t$$

$E(S, t)$ reveals no information about $S$ and the committer cannot open the commitment to $S$ as $S' \neq S$ unless he can find $\log_g(h)$.

1. Dealer($D$) publishes a commitment to $S : E_0 = E(S, t)$ for a randomly choosen $t \in \mathbb{Z}_q$. 

71
2. \( D \) chooses \( F \in \mathbb{Z}_q[x] \) of degree at most \( k - 1 \) satisfying \( F(0) = S \) and computes \( S_i = F(i) \), for \( i = 1, \ldots, n \).

Let \( F(x) = S + F_1x + \cdots + F_{k-1}x^{k-1} \). \( D \) chooses \( G_1, \ldots, G_{k-1} \in \mathbb{Z}_q \) at random and uses \( G_i \) when committing to \( F_i \), for \( i = 1, \ldots, k - 1 \).

\( D \) broadcasts \( E_i = E(F_i, G_i) \), for \( i = 1, \ldots, k - 1 \).

3. Let \( G(x) = t + G_1x + \cdots + G_{k-1}x^{k-1} \) and let \( t_i = G(i) \), for \( i = 1, \ldots, n \).

Then \( D \) sends \((S_i, t_i)\) secretly to participants \( P_i \), for \( i = 1, 2, \ldots, n \).

When \( P_i \) has received his share \((S_i, t_i)\), he verifies that

\[
E(S_i, t_i) = \prod_{j=0}^{k-1} E_j^{i_j}
\]

This scheme also have the advantage that it is easy to derive a verifiable sharing for a linear combination of some secrets. For example let \( S' \) and \( S'' \) are the two secrets that have been shared. If \((S'_i, t'_i)\) and \((S''_i, t''_i)\) be \( P_i \)'s share of \( S' \) and \( S'' \) respectively and let \((E'_0, E'_1, \ldots, E'_{k-1})\) and \((E''_0, E''_1, \ldots, E''_{k-1})\) be the broadcasted messages when the two secrets were shared.

Each \( P_i \) can compute \((E_0, E_1, \ldots, E_{k-1})\) corresponds to a verifiable distribution of \( S = S' + S'' \) mod \( q \) as

\[
E_j = E'_j E''_j \quad \text{for } j = 0, 1, \ldots, k - 1
\]

Furthermore \( P_i \)'s secret share \((S_i, t_i)\) of \( S \) is given by

\[
S_i = S'_i + S''_i \pmod{q}
\]

\[
t_i = t'_i + t''_i \pmod{q}
\]

If both \((S'_i, t'_i)\) and \((S''_i, t''_i)\) are correct shares satisfying the equation \( E(S_i, t_i) = \prod_{j=0}^{k-1} E_j^{i_j} \) then \((S_i, t_i)\) is also a correct share of \( S_i \). That is

\[
g^{S_i t_i} = E_0 E_1^i \cdots E_{k-1}^{i_k-1}
\]
3.3. Publicly Verifiable Secret Sharing

If $S$ is computed as $S = aS' \pmod{q}$ for some $a \in \mathbb{Z}_q^*$, then $P_i$ can compute his share $(S_i, t_i)$ and $(E_0, E_1, \ldots, E_{k-1})$ as follows:

\[
E_j = E_j^a, \quad \text{for } j = 0, 1, \ldots, k - 1
\]

\[
S_i = aS'_i \pmod{q}
\]

\[
t_i = at'_i \pmod{q}
\]

It is noted that any $k$ share holders who have accepted their shares of $S'$ and $S''$ can find a pair $(S, t)$ such that

\[
g^S h^t = E_0
\]

Fewer than $k$ persons have no information about $S$, if $S'$ and $S''$ are distributed correctly.

3.3 Publicly Verifiable Secret Sharing

Stadler [203] has introduced the notion of Publicly Verifiable Secret Sharing (PVSS) schemes in 1996. The proposed PVSS schemes can also be used with general (monotone) access structures. Both schemes are based on ElGamal’s cryptosystem [68]. In PVSS scheme not only the participants but everybody can verify that the shares are correctly distributed. Apart from the applications for ordinary VSS, PVSS can be used for new escrow-cryptosystems and for the realization of digital payment systems with revocable anonymity.

A VSS scheme is a secret sharing scheme with an additional interactive algorithm $Verify$ which allows the participants to verify the validity of their shares:

\[
\exists u \forall A \in \mathcal{A} : (\forall i \in A : Verify(S_i) = 1) \implies Recover(\{S_i|i \in A\}) = u
\]

and $u = S$, if the Dealer was honest. This shows that all group of participants recover the same value if their shares are valid and this
unique value is the secret if the Dealer was honest. In the non interactive scheme the algorithm Verify does not require the interaction between the participants. But even with a non-interactive VSS scheme, the participants can verify the validity of only their own shares. But they cannot know whether other participants have also received valid shares.

This problem can be solved with publicly verifiable secret sharing (PVSS). In a PVSS scheme a public encryption function $E_i$ is assigned to each participant $P_i$, such that only he knows the corresponding decryption function $D_i$. The Dealer now uses the public encryption functions to distribute the shares

$$S_i = E(s_i) \quad i = 1, 2, \ldots, n.$$  

The shares can be verified with the $PubVerify$ algorithm with the property that

$$\exists u \forall A \in 2^{\{1,2,\ldots,n\}} : (PubVerify(\{S_i|i \in A\}) = 1) \implies$$

$$Recover(\{D_i(S_i)|i \in A\}) = u$$

and $u = S$, if the Dealer was honest. If the set of encrypted shares is good according to $PubVerify$ then the honest participants can decrypt them and recover the secret.

Fujisaki and Okamato [74] presents a practical and provably secure PVSS scheme which is $O(|S|)$ times more efficient than Stadler’s PVSS schemes where $|S|$ denotes the size of the secret. It can be incorporated into various cryptosystems based on the factoring and the discrete logarithm to transform them into Publicly Verifiable Key Escrow (PVKE) systems. In addition, those key escrow cryptosystems can be easily modified into the Verifiable Partial Key Escrow (VPKE) systems with the property of delayed recovery. Schoenmakers [186] extended this idea, such that the shareholders can provide a proof of correctness for each share.
3.3. Publicly Verifiable Secret Sharing

released in the reconstruction process. His approach is much simpler than Stadler’s and the followed Fujisaki-Okamoto’s scheme, but is only computationally secure. An information theoretic secure PVSS is proposed by Tang et al [208].

In 2005, Ruiz and Villar [181] proposed a new PVSS scheme that has a higher level of secrecy called indistinguishability (IND) of secrets based on the decisional composite residuosity assumption. In 2009, Heidarvand and Villar [94] gave two new secure definitions of publicly verifiable secret sharing, which capture the notion of indistinguishable shares of secret. Then they proposed a non-interactive PVSS scheme against the attacks of indistinguishability of secrets in the standard model based on the Decisional Bilinear Square (DBS) assumption, which is a natural variant of the standard Decisional Bilinear Diffie-Hellman (DBDH) assumption. In 2010, Jhanwar [113] proposed a PVSS scheme whose level of security is called semantic security based on the \((t,n)\)-multi-sequence of exponents Diffie-Hellman problem. In 2011, Wu and Tseng [220] proposed a pairing based PVSS scheme. For deducing the computational cost, they used the batch verification technique. They also showed that their scheme is a secure PVSS scheme under the bilinear Diffie-Hellman (BDH) assumption in the random oracle model. In fact, semantic security does not guarantee any level of secrecy, if an adversary mounts an active attack. Therefore it is very important to design a PVSS scheme against Adaptively Chosen Secret Attacks (CSA) in the standard model. In 2013 Jia et al [114] proposed a PVSS scheme based on the Chinese Remainder Theorem.

**Berry Schoenmakers Scheme**

In this, the shares distributed by the Dealer can be verified by any one involved and at the same time anybody can verify the shares released by the participant during the reconstruction. Participants not only release the shares but also provides a proof of the correctness of the shares released. The
Chapter 3. Extended Capabilities

security of the scheme is based on decisional Diffie-Hellman assumption. This scheme is much simpler than the schemes proposed by Stadler [203] and Fujisaki [74].

One of the important aspect to be considered is that PVSS doesn’t need a private channel between the Dealer and the participants. All communication is done through authenticated public channel using public key encryption. Hence the secret is only hidden computationally. The protocol proceeds in three stages

**Initialization**

In this step each participant $P_i$ registers with a public key, to be used in the public encryption method $E_i$. The actual participants $P_1, P_2, \ldots, P_n$ involved in the secret sharing scheme are the subset of the registered participants.

**Share Distribution**

The share distribution protocol consist of two steps.

1. **Distribution of shares:** The shares $s_i$ corresponds to the secret $s$ is generated by the Dealer first. The Dealer then publish encrypted shares $E_i(s_i)$ corresponds to each participant $P_i$. The Dealer also publish $PROOF_D$ to ensure that $E_i$ encrypt the share $s_i$. This also make a commitment and the participant can ensure that the reconstruction protocol will result in the same secret $s$.

2. **Verification of shares:** Any one knowing the public key of the encryption method $E_i$ can verify the shares. For each participant $P_i$ a non interactive verification algorithm can be run on $PROOF_D$ to verify that $E_i(s_i)$ is a correct encryption of a share for $P_i$. 

76
3.3. Publicly Verifiable Secret Sharing

Secret Reconstruction

The protocol consists of two steps

1. Decryption of shares: The participants decrypt the shares $s_i$ from $E_i(s_i)$. These participants then release share $s_i$ and also a string $PROOF_{P_i}$ that shows that the released share is correct.

2. The shares of the authorized set of participants are then pooled to reconstruct the secret. The participants are considered as cheaters based on $PROOF_{P_i}$.

Let us consider a $(t,n)$ threshold secret sharing scheme. The scheme can also be applied to any monotone access structure for which linear secret sharing schemes exist. Let $G_q$ denote a group of large prime order $q$, $g$ and $G$ are independently generated generators of the group. The discrete logarithm problem is hard in this group. The Dealer selects a random value $s$ from $\mathbb{Z}_q$ and then distributes the shares of the secret $S = G^s$. A protocol proposed by Chaum and Pederson [49] is used to prove that $\log_{h_1}(g_1) = \log_{h_2}(g_2)$, where $h_1 = g_1^\alpha$ and $h_2 = g_2^\alpha$, for generators $g_1, g_2, h_1, h_2 \in G_q$. Let $DLEQ(g_1, g_2, h_1, h_2)$ denote this protocol. The protocol is as follows

1. The prover will choose a $w$ randomly from $\mathbb{Z}_q$ and send $a_1 = g_1^w, a_2 = g_2^w$ to the verifier.

2. The verifier will send a challenge $c$ chosen randomly from $\mathbb{Z}_q$ to the prover.

3. The prover sends a response $r = w - \alpha c$ back.

4. The verifier checks that $a_1 = g_1^r h_1^c$ and $a_2 = g_2^r h_2^c$. 

77
In the Initialization phase, the participant selects a private key $x_i \in_R \mathbb{Z}_q^*$ and registers $y_i = G^{x_i}$ as public key.

For the distribution of shares among the participants $P_1, P_2, \ldots, P_n$, the Dealer picks a polynomial $p(x)$ of degree $t - 1$.

$$p(x) = \sum_{i=0}^{t-1} a_i x_i$$

where $a_0$ is set with the secret value. The Dealer keeps this polynomial secret and publishes the commitments $C_j = g^{a_j}$, for $0 \leq j \leq t - 1$. The Dealer also publishes the encrypted shares $Y_i = y_i^{p(i)}$, for $1 \leq i \leq n$ and $X_i = \prod_{j=0}^{t-1} C_j^\lambda_i$. The consistency of the shares can be proved by using

$$X_i = g^{a_1_i}, \quad Y_i = y_i^{a_2_i}$$

and using the $DLEQ(g, X_i, y_i, Y_i)$. The challenge $c$ for the protocol is computed by applying a cryptographic hash of $X_i, Y_i, a_1_i, a_2_i, 1 \leq i \leq n$.

In the verification phase, the verifier computes $X_i = \prod_{j=0}^{t-1} C_j^\lambda_i$ using $C_j$ values. Using $y_i, X_i, Y_i, r_i$ and $c$, the verifier computes $a_1_i, a_2_i$ as follows.

$$a_1_i = g^{r_i} X_i^c \quad a_2_i = y_i^{r_i} Y_i^c$$

and checks the hash value of $X_i, Y_i, a_1_i, a_2_i$ matches with $c$.

In the reconstruction phase, each participant can find the share $S_i$ by computing $Y_i^{1/x_i}$. They publish $S_i$ along with the proof of validity. It is accomplished by $DLEQ(G, y_i, S_i, Y_i)$, where $y_i = G^{\alpha}$ and $Y_i = S_i^{\alpha}$. The secret value $S = G^s$ is computed by

$$\prod_{i=1}^t S_i^{\lambda_i} = \prod_{i=1}^t (G^{p(i)})^{\lambda_i} = G^{\sum_{i=1}^t p(i) \lambda_i} = G^{p(0)} = G^s$$

where $\lambda_i = \prod_{j \neq i} \frac{j - i}{j - 1}$ is the Lagrange coefficient.

It is noted that the participant does not have to use the private key $x_i$ in the secret reconstruction, consequently the participant $P_i$ can use its
3.4. Cheater Detection and Identification

key \( x_i \) in several rounds of PVSS. The scheme is also homomorphic. The combined encrypted shares \( Y_1, Y_2 \) can be used to obtain \( G^{s_1}, G^{s_2} \). So the secret retrieved will be \( s = G^{s_1 + s_2} \). Compared to Stadler’s scheme which takes \( O(k^2n) \) time, where \( k \) is a security parameter, this scheme takes only \( O(kn) \) time, which is asymptotically optimal. The security of the scheme depends on breaking the Diffie-Hellman assumption.

3.4 Cheater Detection and Identification

Researchers have considered the problem of guarding against the presence of cheaters in threshold schemes. It is conceivable that any subset of the participants may attempt to cheat, to deceive any of the other participants by lying about the shadows they possess. There is also the possibility that the person distributing the shadows (the Dealer) may attempt to cheat. The Dealer might distribute an inconsistent set of shadows, so that the secret cannot be determined correctly or different subsets of participants would calculate different keys from the shadows they possess. If the cheating is done without the knowledge or cooperation of any of the participants, we refer to this form of cheating as *disruption*. However, if this cheating is done in cooperation with one or more of the participants, we call it *collusion*.

A threshold scheme is said to be unconditionally secure (against cheating), if the probability of successful cheating is limited to a specified probability even if the cheaters are assumed to have infinite computational resources. Under the assumption that the Dealer is honest, several constructions have been given for threshold schemes that are unconditionally secure against cheating.

The general assumptions made in secret sharing scheme is that the Dealer and the combiner are honest but participants can cheat by submitting corrupted shares during the reconstruction. Code based secret sharing provides a solution for this, proposed by McElice and Sarwate
Chapter 3. Extended Capabilities

[144] in 1981. The scheme can detect cheating or even identify the invalid shares and recover the correct secret by requiring more than minimum number of shares needed to determine the secret. Suppose that in a \((t, n)\) threshold scheme there are \(s > t\) shares and \(v\) of which are invalid. If \(s - v \geq t\), then cheating can be detected. If \(s - 2v \geq t\), then invalid shares can be identified and corrected.

The construction of Simmons [197] is more general in that it can be applied to most existing threshold schemes. This method detects cheating only if at least \(t + 1\) participants exchange their shadows. Define a set \(S\) of at least \(t\) shadows to be consistent, if all \(t\)-subsets of \(S\) determine the same key. Then a key is accepted as authentic only if there is a consistent subset of at least \(t + 1\) shadows that determine it. If \(t + e\) participants exchange shadows and there are at most \(e - 1\) cheaters among them, then they possess a consistent subset of at least \(t + 1\) shadows. Unfortunately, the only known method to determine the existence of a consistent set of \(t + 1\) shadows is an exhaustive search. One straightforward solution to the problem of cheating is to have the distributor of shares sign each share \(S_i\) with an unforgeable signature. This is the technique used by Rabin [174] when he used the Shamir’s scheme to solve the problem of agreement among distributed process that might cheat.

Tompa and Wall [213] showed that the Shamir’s scheme is not secure against cheating. A participant can cheat in reconstruction phase by submitting a wrong share and later he can obtain the correct value of the secret, but all other coalescing participants will get wrong secret. They propose modifications to the Shamir’s scheme which allow detection of cheaters with high probability and also prevent the cheater from obtaining the original secret. The following are the advantage of this scheme compared with the signature based scheme.

1. The security of all currently known signature schemes depend on the intractability of factorization and one way function. The scheme
proposed is secure even if the conspirators have unlimited computational resources.

2. The scheme is similar to Shamir’s scheme thus avoiding the complications of implementing an additional signature scheme.

Suppose $P_1 \in P$ is a cheater, he can perform the following steps, such that only he can derive the secret and fool others in a $(t,n)$ threshold Shamir’s scheme.

- Construct a polynomial $\Delta(x)$ of degree at most $t - 1$, such that $\Delta(0) = -1$ and $\Delta(2) = \Delta(3) = \cdots = \Delta(k) = 0$.
- Submit $s_1 + \Delta(1)$ as the pooled shadow.

If all the other participants present the true shadows, the reconstructed $(t-1)$ degree polynomial will be $q(x) + \Delta(x)$. Hence all the honest participant obtain the false secret $q(0) + \Delta(0) = q(0) - 1$. While the cheater $P_1$ can obtain the true secret by adding 1 to the computed result because he knows $\Delta(0)$.

The following is the modified Shamir’s scheme by Tompa and Wall so that the probability of undetected cheating is less than $\epsilon$, for any $\epsilon > 0$.

1. Choose a prime $p > max((s - 1)(t - 1)/\epsilon + k, n)$.
2. Choose $a_1, a_2, \ldots, a_{k-1}$ in $\mathbb{Z}_p$ randomly, uniformly and independently.
3. Let $q(x) = D + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1}$.
4. Choose $(x_1, x_2, \ldots, x_n)$ uniformly and randomly among all permutations of $n$ distinct elements from $1, 2, \ldots, p - 1$. Let $D_i = (x_i, d_i)$, where $d_i = q(x_i)$.

81
Chapter 3. Extended Capabilities

The key difference between this and Shamir’s scheme occurs in step 4. Suppose participants $P_1, P_2, \ldots, P_{k-1}$ fabricate values $(x'_1, d'_1), (x'_2, d'_2), \ldots, (x'_{k-1}, d'_{k-1})$ and send to participant $P_k$. The participant $P_k$ will reconstruct the incorrect secret $D'$ only if $q_D(x_k) = q(x_k)$ and $D' \neq D$. Thus for each polynomial $q'_D(x)$ with $D' \neq D$, the probability that $q'_D(x_k) = q(x_k)$ is at most $(k - 1)/(p - k)$. There are $s - 1$ legal but incorrect shares, so the fabricated values yield $s - 1$ corresponding polynomials. Any one of these polynomials would deceive participants $P_k$ with probability at most $(k - 1)/(p - k)$. Thus the probability of deceiving participant $P_k$ is at most $(s - 1)(k - 1)/(p - k) < \epsilon$. Even though cheaters can be detected with high probability, they obtain the secret but other participants gain no information about the secret. A solution to this is to use a dummy value, say $s$, that is never used as a value of the real secret. The true secret $D$ is now encoded as a sequence $D^1, D^2, \ldots, D^i$, where $D^i = D$ for some randomly chosen $i$ and $D^j = s$ for all $j \neq i$. Each element of this sequence is then divided into shares. When $k$ participants agree to pool their shares, they reconstruct $D^1, D^2, \ldots$ one at a time until some $D^i \neq s$ is obtained. If $D^i$ is not legal then cheating has occurred.

In summary the Dealer specifies a subset $K_0$ of the set of possible keys $\mathcal{K}$. A key will be accepted as authentic only if it is an element of $K_0$. If a set of $t$ participants calculate the key to be an element of $K \notin K_0$, then they realize that one of them is cheating. The probability of successful cheating is at most $1 - t|K_0|/|\mathcal{K}|$, even if participants conspire to cheat another participant. Even though participants can detect when cheating has occurred, they cannot determine who is cheating.

Rabin and Ben-or [176] developed a scheme based on Shamir’s threshold scheme in which the honest participants are able to identify cheaters. In this scheme every participant in $\mathcal{P}$ receives extra information along with his share over a finite field to guard against cheating. Indeed, each participant
3.4. Cheater Detection and Identification

$P_i$ in $P$ receives his share $d_i$ and $n - 1$ random elements $v_{ij}$, for $j = 1, \ldots, n$ and $j \neq i$. Moreover each participant $P_j$ in $P - P_i$ receives $n - 1$ pairs $(w_{ji}, z_{ji})$, for $i = 1, \ldots, n$ and $i \neq j$, where $w_{ji} \neq 0$ is a random element and $z_{ji}$ is calculate as $z_{ji} = d_i + v_{ij}w_{ji}$. When the participant $P_i$ wants to let $P_j$ know his share, he returns the pair $(d_i, v_{ij})$. Then $P_j$ can calculate $d_i + v_{ij}w_{ji}$, and he accepts $d_i$ only if the result is $z_{ji}$. The probability that the coalition of $n - 1$ participants cheat successfully the remaining honest participant is $1 - (1 - \frac{1}{|S|-1})^{n-k+1} \leq \frac{n-k+1}{|S|-1}$. Where $S$ is the set of secrets.

Brickell and Stinson [37] proposed a modified version of the Bickley’s construction [24] in which honest participants are able to identify cheaters. Brickell and Stinson considered a somewhat different scenario from Tompa and Woll. There is a honest participant and the remaining $n - 1$ participants form a coalition in order to deceive him. If $s$ is the correct secret, some $k - 1$ participants of the $n - 1$ cheaters could return forged shares in an attempt to force the $n$-th honest one to reconstruct a secret $s' \neq s$. If the honest participant can identify the false shares, he asks the remaining participants for another share. Then the $n - 1$ cheaters can return forged shares until at most $n - k + 1$ participants are identified as cheaters. In Brickell and Stinson’s construction even if there is only one honest participant and the remaining $n - 1$ participant form a coalition in order to deceive him is $\frac{n-k+1}{|S|-1}$, where $S$ is the set of secrets. The information given to participants is less in Brickell and Stinson’s scheme compared with the Rabin and Ben-Or’s scheme but the Brickell and Stinson’s scheme is not perfect and is not computationally efficient, if $n$ and $k$ are large. Conversely Rabin and Ben-Or’s scheme is perfect and can be implemented in polynomial time.

A generalized secret sharing sharing scheme with cheater detection and identification is proposed by Lin [134]. It is computationally secure and each participant holds only one single shadow. Any honest participant in this scheme can detect and identify who is cheating even when all of the
other participants corrupt together. An extended algorithm is also proposed
to protect the secret from the dishonest participant.

A t cheater identifier for \((k, n)\) Shamir’s Threshold scheme based on
orthogonal arrays and error correcting codes are proposed by Kurosawa
et al \[127\]. An optimal and easy scheme with smaller share size based on
Kurosawa’s scheme is proposed by Obana \[156\].

Carpentieri \[44\] present a perfect and unconditionally secure \((k, n)\)
threshold secret sharing scheme having the same properties of Rabin and
Ben-Or’s scheme. But the information given to each participant is smaller
in this scheme. Let \(GF(q)\) be a finite field with \(q\) elements, where \(q\) is a
prime power such that \(q > n\). Assume that the secret \(S\) is chosen in the
finite field \(GF(q)\) by a special participant called the Dealer. The Dealer is
denoted by \(D\) and assume \(D \notin P\). The construction is based on Shamir’s
threshold secret sharing scheme. When \(D\) wants to share the secret \(S\)
among the participants in \(P\), he gives a \(k\)-dimensional vector \(\bar{d}_i \equiv (d_{i,0}, \ldots, d_{i,k-1})\), where \(k \leq n\), over \(GF(q)\) as a share to participant
\(P_i\), for \(i = 1, \ldots, n\). The Dealer chooses the shares as follows. Let
\(a_1, \ldots, a_{k-1}\) be elements chosen uniformly at random in \(GF(q)\) and
unknown to all the participants. Let \(\alpha_1, \ldots, \alpha_n\) be distinct and non-null
elements in \(GF(q)\) known by all the participants. If \(q(x)\) is the polynomial
\(S + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}\), then \(d_{i,0} = q(\alpha_i)\) and \(d_{i,1}, \ldots, d_{i,k-1}\) are
elements chosen uniformly at random in \(GF(q)\), for \(i = 1, \ldots, n\). To guard
against cheating, \(D\) distributes extra information to the participants along
with their shares. The extra information consists of \(n - 1\) pairs of elements
in \(GF(q)\) for each participant \(P_j\) in \(P\). Let \(g_{j,i}\) be non null elements chosen
uniformly at random in \(GF(q)\), for \(i = 1, \ldots, n\) and \(i \neq j\). \(D\) calculates
\(b_{j,i} = g_{j,i} d_{i,0} + \alpha_j d_{i,1} + \cdots + \alpha_j^{k-1} d_{i,k-1}\) and then, he gives the participant
\(P_j\) the pair \((g_{j,i}, b_{j,i})\), for \(i = 1, \ldots, n\) and \(i \neq j\). When the participants \(P_i\)
return his share \(\bar{d}_i\), \(P_j\) can check the authenticity of \(\bar{d}_i\) by verifying that it
is a solution vector of the equation
\(g_{j,i} y_0 + \alpha_j y_1 + \cdots + \alpha_j^{k-1} y_{k-1} = b_{j,i}\),
where \( y_0, \ldots, y_{k-1} \) are the unknowns, \( g_{j,i}, \alpha_j, \ldots, \alpha_{j}^{k-1} \) are the coefficients and \( b_{j,i} \) is the constant for \( i = 1, \ldots, n \) and \( i \neq j \).

T.C.Wu and T.S.Wu [219] proposed a method to detect and identify cheaters. Arithmetic coding and one way hash functions are used to deterministically detect cheating and identify the cheaters no matter how many cheaters are involved in the secret reconstruction. Cheater detection and identification in CRT based schemes especially Mingotte and Asmuth-Bloom is proposed by Pasailua et al [165].

Harn and Lin [91] developed a scheme in 2009. They assumed that there are more than \( t \) participants are there in the secret reconstruction. Since there are more than \( t \) shares, it only requires \( t \) shares for reconstructing the secret. The redundant shares can be used for cheater detection and identification. Some flaws of this is reported by Ghodosi [78].

### 3.5 Robust Secret Sharing

Secret sharing schemes having the property that the correct secret can still be recovered even if some of the shares are invalid are called robust secret sharing scheme. Code based secret sharing scheme can be robust. Some secret sharing schemes have the capability to detect and identify cheating. But they are not necessarily robust. To achieve robustness, the shares in the schemes should contain additional information so that the shares can be checked for correctness. Rogaway and Bellare [179] studied this within a number of different models.

A \((k, n)\) threshold scheme that can identify \( r < k/2 \) cheaters can be used to create an almost robust \((k, n)\) threshold scheme that allows honest participants to obtain the secret under certain circumstances. If the secret \( s = k_1 \oplus k_2 \), then by giving each participant one share in a \((k, n)\) threshold scheme that can identify \( r \) cheaters with secret \( k_1 \), and one share in \((k - r, n)\) scheme that can identify \( r \) cheaters with secret \( k_2 \).
Chapter 3. Extended Capabilities

During the reconstruction, participants submit their first shares and they are checked for the presence of cheaters. If the cheaters are identified, the recovery is aborted. If no cheaters are noted in the first stage, participants submit their second share. Even if $r$ cheaters are identified $k - r$ honest participants can still recover $k_2$. The secret is then computed using $k_1$ and $k_2$.

3.6 Cheating Immune Secret Sharing

In the attack mentioned by Tompa et al [213], the cheater will gain the knowledge of the secret, but all other honest participants will get an invalid secret. The approach in cheating immune system is to prevent the cheater from knowing the secret. So the adversary does not have a personal gain rather than disrupting the recovery of the original secret. Honest participants are willing to sacrifice recovery of the secret if an adversary corrupts shares, so long as the adversary does not have an advantage over the honest participants with respect to the recovery of genuine secret.

Cheating immune secret sharing schemes were first proposed by Zhang, Xian-Mo and Pieprzyk [225]. They considered binary shares and boolean functions. Two notions were proposed, $t$-cheating immune, where an adversary who submits $t$ incorrect shares gains no advantage and a more general construction strictly $t$-cheating immune, where an adversary who submit up to $t$ incorrect shares gains no advantage. Properties and constraints of cheating immune scheme is mentioned in [58] by Stinson et al. A necessary condition for a secret sharing system to be cheating immune is specified in [33]. The known constructions for cheating immune system is for only $(n,n)$ schemes. It is an active research topic to construct cheating immune secret sharing schemes for more general structures. A cheating immune secret sharing scheme for a $(t,n)$ threshold
scheme is proposed using codes and cumulative arrays by Cruz and Wang [62].

### 3.7 Proactive Secret Sharing

The Secret Sharing scheme assumes long-lived shares. In a \((t, n)\) threshold scheme an adversary can corrupt \(n - t + 1\) shares in order to destroy the secret information. The adversary have entire life time of the shares to mount these attack. The solution to these problem is to periodically renew the shares without changing the secret in such a way that any information learned by the adversary about individual shares becomes obsolete after the shares are renewed. Similarly to avoid the gradual destruction of the information by corruption of shares, it is necessary to periodically recover lost or corrupted shares without compromising the secrecy of the recovered shares.

Proactive security for secret sharing was first suggested by Ostrovski and Yung in [158] in 1991, where they presented among other things, a proactive polynomial secret sharing scheme. The scheme uses the verifiable secret sharing scheme of [176]. Proactive security refers to security and availability in the presence of a mobile adversary. Herzberg et al [97] further specialized this notion to robust secret sharing schemes and gave a detailed and efficient proactive secret sharing scheme in 1995. Robust means that in any time period, the shareholders can reconstruct the secret value correctly.

In Herzberg et al [97] proactive approach, the lifetime of the secret is divided into periods of time (e.g., a day, one week, etc.). At the beginning of each time period the share holders engage in an interactive update protocol, after which they hold completely new shares of the same secret. Previous shares become obsolete and should be safely erased. As a consequence, in the case of a \((k, n)\) proactive threshold scheme, the adversary trying to learn the secret is required to compromise \(k\) locations during a single time period,
as opposed to incrementally compromising \( k \) locations over the entire secret life-time.

Thus the goal of the pro-active security scheme is to prevent the adversary from learning the secret or from destroying it. In particular any group of \( t \) non-faulty shareholders should be able to reconstruct the secret whenever it is necessary. The term pro-active refers to the fact that it’s not necessary for a breach of security to occur before secrets are refreshed, the refreshment is done periodically (and hence pro-actively).

The core properties of pro-active secret sharing

1. Renewal of existing shares without changing the secret. The shares that are exposed previously will not damage the secret and become useless.

2. Recovery of lost or corrupted shares without compromising the secrecy of the shares. i.e., reconstruction of lost or corrupted shares.

Pro-active Model Requirements

1. An adversary can reveal at most \( t-1 \) shares in any time period, where \( t-1 < n/2 \). This guarantees the existence of \( t \) honest shareholders at any given time. This time period should be synchronized with the share-renewal protocol.

2. Authenticated broadcast channel and an authenticated secret communication channels between any two participants.

3. Synchronization: the servers (shareholders) can access a common global clock so that the protocol can be applied in a certain time period.

4. Shares can be erased: every honest server (shareholder) can erase its shares in a manner that no attacker can gain access to erased data.
3.7. Proactive Secret Sharing

3.7.1 Basic model of Proactive Secret Sharing

This scheme is proposed by Herzberg [97]. Consider a \((t, n)\) threshold scheme by Shamir, where a polynomial \(f(x)\) of degree \(t - 1\) is used to distribute shares \(f(i)\) to each participant. In this protocol the shares are renewed without the Dealer’s involvement. The share holders will agree upon a new polynomial with the same secret \(K\) without revealing the old secret. The assumption made in this protocol is that the old shares are all valid and the participants are honest. After the initialization, at the beginning of each time period, all honest shareholders perform a share renewal protocol as follows.

1. Each \(i\)’th share holder \(i \in \{1, \ldots, n\}\) randomly pick \(t - 1\) numbers from the finite field and define a polynomial \(p_i(x)\) of degree \(t - 1\) with \(p_i(0) = 0\).

2. Each \(i\)’th share holder distributes the share’s of \(p_i(x)\) using verifiable secret sharing among the share holders.

3. Each share holder computes his new share by adding his old shares to the sum of the \(n\) new shares i.e.,

\[
h(i) = f(i) + \sum_{j=1}^{n} p_j(i)
\]

4. Each \(i\)’th share holder erases his old share \(f(i)\).

This protocol solves the problem against passive adversary who may try to learn the shares and obtain the secret. The active adversary however can cause the destruction of the secret by dealing inconsistent shares or by choosing a polynomial \(p_i(x)\) with \(p_i(0) \neq 0\). Therefore verifiability feature is added to the basic protocol to make sure that the shares are consistent. Feldman’s [71] verifiable secret sharing scheme can be used.
Detection of corrupted share is another important thing to consider. Participating share holders must make sure that shares of other share holders have not been corrupted or lost. The corrupted shares must be restored, if necessary. An adversary could cause the loss of the secret by destroying $n - t + 1$ shares otherwise. The shares may be corrupted due to disk crash or some hardware failure, which cause the server to be down.

The way to know that the share is modified by hacker or some other means is to save some fingerprint for each share that is common to all shareholders. The shareholders can periodically compare shares (using secure broadcast). The basic technique used to reconstruct the lost or corrupted share is to send sufficient information to the share holder $r$, who lost his share. These information can be used to recover the corrupted share without dealer’s involvement. The following is the algorithm:

1. Each $i$’th shareholder $i \in \{1, \ldots, r-1, r+1, \ldots, n\}$ randomly choose a polynomial $p_i(x)$ of degree $t-1$ where $p_i(r) = 0$ and $p_i(0) \neq 0$.

2. Each $i$’th share holder distributes shares $p_i(1), \ldots, p_i(n)$ using VSS among share holders (except for the $r$’th share holder).

3. Each $i$’th share holder (except $r$) receives $p_1(i), \ldots, p_{r-1}(i), p_{r+1}(i), \ldots, p_n(i)$ and calculate his new share and send it encrypted to $r$.

$$h(i) = f(i) + \sum_{j=1}^{r-1} p_j(i) + \sum_{k=r+1}^{n} p_k(i)$$

4. The $r$’th share holder decrypts these shares and interpolate them to recover $h(r) = f(r)$.

This protocol is secure only against an adversary that eavesdrops on $t-1$ or less shareholders, but cannot change their behavior.
3.8. Concluding Remarks

Jarecki [112] also come up with two methods of proactive secret sharing in 1995. One using Feldman’s [71] verifiable secret sharing scheme and the other one using Pedersen’s scheme [167]. Stinson and Wei [205] introduced a new verifiable secret sharing scheme and then a proactive scheme is developed using this. A combinatorial structure is introduced which makes the scheme more efficient. Cachin et al [41] introduced proactive crypto systems in asynchronous networks and presents an efficient protocol for refreshing the shares of a secret key for discrete logarithm-based sharing. Nikov et al [155] mentioned how to apply general access structure to proactive secret sharing.

Mobile Proactive Secret Sharing (MPSS) is proposed by [189]. MPSS is a new way to do proactive secret sharing in asynchronous networks. MPSS provides mobility. The group of nodes holding the shares of the secret can change at each resharing, which is essential in a long-lived system. MPSS additionally allows the number of tolerated faulty shareholders to change when the secret is moved, so that the system can tolerate more (or fewer) corruptions. This allows reconfiguration on the fly to accommodate changes in the environment.

Bai et al [5] proposed a proactive secret sharing scheme based on matrix projection method. An adaptive proactive secret sharing scheme is proposed by Wang [217]. In some environment, it needs to change not only the number of participants $n$ but also the threshold value $t$. An adaptive proactive secret sharing is to refresh the shares as $t$ and $n$ change.

3.8 Concluding Remarks

In this chapter we have considered some of the extended capabilities of secret sharing schemes. We have done a survey on various additional properties and also explored the constructions, which are efficient and
Chapter 3. Extended Capabilities

easy to implement. Verifiable secret sharing is an important construct, when the participant wants to check whether the shares issued by the Dealer are consistent. PVSS allows not only the participant but any one can verify the consistency of the shares.

Cheater detection and identification is a major issue. It is noted that Shamir’s scheme is not cheater resistant. A misbehaving participant may submit an invalid share during the reconstruction phase. This will result in, all genuine participant may receive wrong secret, where as only the cheater will obtain the correct secret. Secret sharing mechanism have to address this problem. We have considered several methods to detect cheating in secret sharing scheme. Not only the detection of cheating is important in secret sharing but also identification of the cheaters. So in general some desirable properties of secret sharing schemes are public verification of shares for ensuring the consistency and also the cheating detection and identification of cheaters.

The proactive secret sharing schemes prevents the perceptual leakage of the share information. The modification of shares in proper interval will make the intruder to gain no information about the secret even though he had a valuable share hacked over time. Robustness allow the secret to be reconstructed even if there is an invalid share submitted by a dishonest participant. Cheating immune system prevents the dishonest participant to recover the original secret after submitting a wrong share during the reconstruction phase.

In general, developing a good secret sharing scheme aims at incorporating these desirable features efficiently in the scheme. We have incorporated verifiability, cheating detection and identification in the proposed secret sharing schemes in the later chapters. Another important capability to be considered is the multi secret sharing, which is discussed in Chapter 6.