Chapter 6

Multi Secret Sharing

6.1 Introduction

There are several situations in which more than one secret is to be shared among participants. As an example, consider the following situation described by Simmon [199]. There is a missile battery and not all of the missiles have the same launch enable code. We have to devise a scheme which will allow any selected subset of users to enable different launch code. The problem is to devise a scheme which will allow any one or any selected subset of the launch enable codes to be activated in this scheme. This problem could be trivially solved by realizing different secret sharing schemes, one for each of the launch enable codes. But this solution is clearly unacceptable since each participant should remember too much information. What is really needed is an algorithm such that the same pieces of private information could be used to recover different secrets.

Some results of this chapter are included in the following paper.

Multi Secret Sharing

One common drawback of secret sharing scheme is that they are all one-time schemes. That is once a qualified group of participants reconstructs the secret $K$ by pooling their shares, both the secret $K$ and all the shares become known to everyone and there is no further secret. In other words, the share kept by each participant can be used to reconstruct only one secret.

Karnin, Greene and Hellman [117] in 1983 mentioned the multiple secret sharing scheme where threshold number of users can reconstruct multiple secrets at the same time. Alternatively the scheme can be used to share a large secret by splitting it into smaller shares. Franklin et al [72] in 1992 used a technique in which the polynomial-based single secret sharing is replaced with a scheme, where multiple secrets are kept hidden in a single polynomial. They also considered the case of dependent secrets in which the amount of information distributed to any participant is less than the information distributed with independent schemes. Both the schemes are not perfect. They are also one time threshold schemes. That is, the shares cannot be reused.

Blundo et al [28] in 1993 considered the case in which $m$ secrets are shared among participants in a single access structure $\Gamma$ in such a way that any qualified set of participants can reconstruct the secret. But any unqualified set of participants knowing the value of number of secrets might determine some (possibly no) information on other secrets. Jackson et al [110] in 1994 considered the situation in which there is a secret $S_k$ associated with each subset of $k$ participants and $S_k$ can be reconstructed by any group of $t$ participants in $k$ ($t \leq k$). That is each subset of $k$ participants is associated with a secret which is protected by a $(t, k)$-threshold access structure. These schemes are called multi-secret threshold schemes. They came up with a combinatorial model and optimum threshold multi secret sharing scheme. Information theoretic model similar to threshold scheme is
also proposed for multi-secret sharing. They have generalized and classified the multi-secret sharing scheme based on the following facts.

- Should all the secrets be available for potential reconstruction during the lifetime of the scheme or should the access of secrets be further controlled by enabling the reconstruction of a particular secret only after extra information has been broadcast to the participants.

- Whether the scheme can be used just once to enable the secrets or should the scheme be designed to enable multiple use.

- If the scheme is used more than once then the reconstructed secret or shares of the participants is known to all other participants or it is known to only the authorized set.

- The access structure is generalized or threshold in nature.

In 1994 He and Dawson [93] proposed the general implementation of multistage secret sharing. The proposed scheme allows many secrets to be shared in such a way that all secrets can be reconstructed separately. The implementation uses Shamir’s threshold scheme and assumes the existence of a one way function which is hard to invert. The public shift technique is used here. A \( t - 1 \) degree polynomial \( f(x) \) is constructed first as in Shamir’s scheme. The public shift values are \( d_i = z_i - y_i \), where \( z_i = f(x_i) \). The \( y_i \)'s are the secret shares of the participant. These \( y_i \)'s are then send to the participants secretly. For sharing the next secret \( h(y_i) \) is used, where \( h \) is the one way function. The secrets are reconstructed in particular order, stage by stage and also this scheme needs \( k n \) public values corresponds to the \( k \) secrets. The advantage is that each participant has to keep only one secret element and is of the same size as any shared secret.

In 1995 Harn [89] shows an alternative implementation of multi stage secret sharing which requires only \( k(n - t) \) public values. The
implementation become very attractive especially when the threshold value \( t \) is very close to the number of participants \( n \). In this scheme an \((n - 1)\) degree polynomial \( f(x) \) is evaluated at \((n - t)\) points and are made public. Any set of \( t \) participants can combine their shares with the \((n - t)\) public shares to interpolate the degree \((n - 1)\) polynomial. Multiple secrets are shared with the help of one way function as in He and Dawson scheme.

The desirable properties of a particular scheme depends on both the requirements of the application and also the implementation. Several multi secret threshold schemes and schemes based on general access structure are developed by the research community. In this chapter we only explore some of the important constructions of multi-secret sharing scheme using general access structure. We then propose a multi secret sharing scheme realizing the general access structure, which is based on Shamir’s scheme and hardness of discrete logarithm problem. The scheme is simple and easy to implement. The proposed scheme has many practical applications in situations where the participants set, access rules or the secret itself change frequently. When new participants are included or participants leave, there is no need of issuing new shares. Such situation often arise in key management, escrowed system etc.

### 6.2 Cachin’s Scheme

A computationally secure secret sharing scheme with general access structure, where all shares are as short as the secret is proposed by Christian Cachin [40] in 1995. The scheme also provides capability to share multiple secrets and to dynamically add participants on-line without having to redistribute new shares secretly to the current participants. These capabilities are achieved by storing additional authentic information in a publicly accessible place, which is called a
noticeboard or bulletin board. This information can be broadcast to the participants over a public channel. The protocol gains its security from any one-way function. The construction has the following properties.

- All shares must be transmitted and stored secretly once for every participants and are as short as the secret.

- Multiple secret can be shared with different access structure requiring only one share per participant for all secrets.

- Provides the ability for the dealer to change the secret after the shares have been distributed.

- The Dealer can distribute the shares on-line. When a new participant is added and the access structure is changed, already distributed shares remain valid. Shares must be secretly send to the new participants and the publicly readable information has to be changed.

Let the secret $K$ be an element of finite Abelian Group $G = \langle G, + \rangle$. The basic protocol to share a single secret is as follows.

1. The Dealer randomly chooses $n$ elements $S_1, S_2, \ldots, S_n$ from $G$ according to the uniform distribution and send them secretly to the participants over a secret channel.

2. For each minimal qualified subset $X \in \Gamma_0$, the Dealer computes

$$T_X = K - f(\sum_{x: P_x \in X} S_X)$$

and publishes $T = \{T_X | X \in \Gamma_0\}$ on the bulletin board.

In order to recover the secret $K$, a qualified set of participants $Y$ proceeds as follows.
1. The members of $Y$ agree on a minimal qualified subset $X \subseteq Y$.

2. The members of $X$ add their shares together to get $V_X = \sum_{x : P_x \in X} S_x$ and apply the one-way function $f$ to the result.

3. They fetch $T_X$ from the bulletin board and compute $K = T_X + f(V_X)$.

The shares of the participants in $X$ are used in the computation to recover the secret $K$. For the basic scheme where only one secret is shared, the shares do not have to be kept secret during this computation. However for sharing multiple secrets the shares and the result of their addition have to be kept secret.

In order to share multiple secrets $K^1, K^2, \ldots, K^h$ with different access structures $\Gamma^1, \Gamma^2, \ldots, \Gamma^h$ among the same set of participants $\mathcal{P}$, the Dealer has to distribute the private shares $S_i$ only once but prepares $\mathcal{T}^1, \mathcal{T}^2, \ldots, \mathcal{T}^h$ for each secret. The single secret sharing scheme cannot be applied directly for multi secret sharing because it is not secure. If a group of participants $X$ qualified to recover both $K^1$ and $K^2$ then any group $Y \in \Gamma^1$ can obtain $K^2$ as

$$K^2 = T_X^2 + T_Y^1 + f(V_Y) - T_X^1$$

To remedy this deficiency, the function $f$ is replaced by a family $F = f_h$ of one-way functions so that different one-way functions are employed for different secrets. The following protocol is used to share $m$ secrets.

1. The Dealer randomly chooses $n$ elements $S_1, S_2, \ldots, S_n$ from $G$ and send them securely to the participants as shares.

2. For each secret $K^h$ to share (with $h = 1, \ldots, m$) and for each minimal qualified subset $X \in \Gamma^h_0$, the Dealer computes

$$T_X^h = K^h - f_h(\sum_{x : P_x \in X} S_x)$$
6.2. Cachin’s Scheme

and publishes \( T^h = \{ T^h_X | X \in \Gamma^h_0 \} \) on the bulletin board.

In order to recover some secret \( K^h \), a set of participants \( Y \in \Gamma^h \) proceeds as follows.

1. The members of \( Y \) agree on a minimal qualified subset \( X \subseteq Y \).

2. The members of \( X \) add their shares together to get \( V_X = \sum_{x : P_x \in X} S_X \) and apply the one-way function \( f_h \) to the result.

3. They fetch \( T^h_X \) from the bulletin board and compute \( K^h = T^h_X + f_h(V_X) \).

The scheme does not demand a particular order for the reconstruction of the secrets as in He and Dawson scheme. The required family of functions \( F \) can be easily be obtained from \( f \) by setting \( f_h(x) = f(h + x) \), when \( h \) is represented suitably in \( G \). Because different one-way function \( f_h \) is used for each secret, it is computationally secure. But the shares have to be protected from the eyes of other participants during the reconstruction. Otherwise these participants could subsequently recover other secrets they are not allowed to know. Therefore the computation of \( f_h(V_X) \) should be done without revealing the secret shares.

In many situations, the participant of a secret sharing scheme do not remain the same during the entire life-time of the secret. The access structure may also change. In this scheme, it is assumed that the changes to the access structure are monotone, that is participants are only added and qualified subsets remain qualified. The scheme is not suitable for access structures which are non-monotonic. Removing participants is also an issue which is not addressed. In multi-secret sharing, the shares must be kept hidden to carry out the computation. Cachin suggest that computations involved in recovering \( K \) could be hidden from the participants using a distributed evaluation protocol proposed by
Goldreich et al [83]. For access to a predetermined number of secrets in
fixed order, a variant of one-time user authentication protocol of Lamport
[130] could be used.

6.3 Pinch’s Scheme

The Cachin’s scheme does not allow shares to be reused after the secret has
been reconstructed. A distributed computation sub protocol is proposed
using one way function. But it allows the secret to be reconstructed in a
specified order. Pinch [169] in 1996 proposed a modified algorithm based on
the intractability of the Diffie-Hellman problem in which arbitrary number
of secrets can be reconstructed without having to redistribute new shares.

Let \( M \) be a multiplicative group in which the Diffie-Hellman problem
is intractable. That is given elements \( g, \) \( g^x \) and \( g^y \) in \( M \), it is
computationally infeasible to obtain \( g^{xy} \). This is called Computational
Diffie Hellman Problem. This implies the intractability of the discrete
logarithm problem. If the discrete logarithm problem can be solved then
the Diffie-Hellman problem can also be solved. Suppose \( f : M \to G \) is a
one-way function, where \( G \) be the additive group modulo some prime \( p \)
and \( M \) be the multiplicative group to the same modulus, which will be
cyclic of order \( q \). The protocol proceeds as follows:

1. The Dealer randomly chooses secret shares \( S_i \), as integers coprime
to \( q \), for each participant \( P_i \) and send them through a secure
channel. Alternatively Diffie-Hellman key exchange can be used
using the group \( M \) to securely exchange \( S_i \).

2. For each minimal trusted set \( X \in \Gamma \), the Dealer randomly chooses \( g_X \)
to be a generator of \( M \) and computes

\[
T_X = K - f\left( g_X \prod_{x \in X} S_x \right)
\]
and publish \((g_X, T_X)\) on the notice board.

In order to recover the secret \(K\), a minimal trusted set \(X = P_1, \ldots, P_t\), of participants comes together and follow the protocol mentioned below.

1. Member \(P_1\) reads \(g_X\) from the notice board and computes \(g_X^{S_1} \) and passes the result to \(P_2\).

2. Each subsequent member \(P_i\), for \(1 < i < t\), receives \(g_X^{S_1 \cdots S_{i-1}}\) and raises this value to the power \(S_i\) to form
   \[ V_X = g_X^{\prod_{i=1}^{t-1} S_i} = g_X^{\prod_{x \in X} S_x} \]

3. On behalf of the group \(X\), the member \(P_t\) reads \(T_X\) from the notice board and can now reconstruct \(K\) as \(K = T_X + f(V_X)\).

If there are multiple secrets \(K_i\) to share, it is now possible to use the same one way function \(f\), provided that each entry on the notice board has a fresh value of \(g\) attached. There is a variant proposal which avoids the necessity for the first participant to reveal \(g^{S_1}\) at the first step. The participant \(P_1\) generates a random \(r \pmod q\) and passes the result of \(g_X^{rS_1}\) to \(P_2\). The participant \(P_t\) will pass \(g_X^{rS_1 \cdots S_{t-1}}\) back to \(P_1\). \(P_1\) can find \(w\) such that \(rw \equiv 1 \pmod q\) and raises \(g_X^{rS_1 \cdots S_n}\) to the power \(w\) to form
   \[ V_X = g_X^{\prod_{i=1}^{t-1} S_i} = g_X^{\prod_{x \in X} S_x} \]

Ghodosi et al [77] showed that Pinch’s scheme is vulnerable to cheating and they modified the scheme to include cheating prevention technique. In Pinch’s scheme a dishonest participant \(P_i \in X\) may contribute a fake share \(S'_i = \alpha S_i\), where \(\alpha\) is a random integer modulo \(q\). Since every participant of an authorized set has access to the final result \(g_X^{S_1 \cdots S'_i \cdots S_t}\), the participant \(P_i\) can calculate the value
   \[ \left( g_X^{S_1 \cdots S'_i \cdots S_t} \right)^{\alpha^{-1}} = g_X^{S_1 \cdots S_i \cdots S_t} = g_X^{\prod_{x \in X} S_x} = V_X \]
and hence obtain the correct secret, where as the other participants will get an invalid secret.

The cheating can be detected by publishing $g_X^V$ corresponds to the every authorized set $X$ in the initialization step by the Dealer. Every participants $x \in X$ can verify whether $g_X^V = g_X^V'$, where $V_X'$ is the reconstructed value. However this cannot prevent cheating or cheaters can be identified. The cheating can be prevented by publishing extra information on the notice board. Let $C = \sum_{x \in X} g_x^S$. For each authorized set $X$, the Dealer also publishes $C_X = g_X^C$. At the reconstruction phase, every participant $P_i \in X$ computes $g_i^S$ and broadcasts it to all participants in the set $X$. Thus every participant can computes $C_X$ and verifies $C_X = g_X^C$. If the verification fails, then the protocol stops. If there exist a group of collaborating cheaters, they can cheat in the first stage. Yeun et al [222] proposed a modified version of the Pinch’s protocol which identifies all cheaters regardless of their number, improving on previous results by Pinch and Ghodosi et al.

## 6.4 RJH and CCH scheme

An efficient computationally secure on-line secret sharing scheme is proposed by Re-Junn Hwang and Chin-Chen Chang [101] in 1998. In this each participant hold a single secret which is as short as the shared secret. They are selected by the participants itself, so a secure channel is not required between the Dealer and the participants. Participants can be added or deleted and secrets can be renewed with out modifying the secret share of the participants. The shares of the participants is kept hidden and hence can be used to recover multi secrets. The scheme is multi use unlike the one time use multi secret sharing scheme.

In Cachin’s and Pinch’s schemes, the Dealer has to store the shadow of each participant to maintain the on-line property. The Dealer storing
the shares is an undesirable property in secret sharing scheme. This scheme avoids the problem and provides great capabilities for many applications. The scheme has four phases: initialization phase, construction phase, recovery phase and reconstruction/renew phase.

Assume that there are \( n \) participants \( P_1, P_2, \ldots, P_n \), sharing a secret \( K \) with the monotone access structure \( \Gamma = \{ \gamma_1, \gamma_2, \ldots, \gamma_t \} \). In the initialization phase the Dealer select two strong primes \( p, q \) and publishes \( N \) on the public bulletin, where \( N \) is the multiplication of \( p \) and \( q \). The Dealer also chooses another integer \( g \) from the interval \([N^{1/2}, N]\) and another prime \( Q \) which is larger than \( N \) and publishes them. Each participant can select an integer \( S_i \) in the interval \([2, N]\) and computes \( U_i = g^{S_i} \pmod{N} \). Each participant keeps \( S_i \) secret and send the pseudo share \( U_i \) and the identifier \( ID_i \) to the Dealer. If certain different participant select same shadow, the Dealer asks for new shadows or alternatively the Dealer can select the shares and send to the participants securely. But this need a secure channel. Finally Dealer publishes \((ID_i, U_i)\) of each participant \( P_i \) in the public bulletin.

In the construction phase the Dealer computes and publishes some information for each qualified subset in access structure \( \Gamma \). The participants of any qualified subset \( \gamma_j \) can cooperate to recover the shared secret \( K \) by using these information and the values generated from their shadows in the recovery phase. The public information corresponds to each qualified set is generated as follows.

- Randomly select an integer \( S_0 \) from the interval \([2, N]\) such that \( S_0 \) is relatively prime to \( p - 1 \) and \( q - 1 \).
- Compute \( U_0 = g^{S_0} \pmod{N} \) and \( U_0 \neq U_i \) for all \( i = 1, 2, \ldots, n \).
- Generate an integer \( h \) such that \( S_0 \times h \equiv 1 \pmod{\phi(N)} \).
- Publish \( U_0 \) and \( h \) on the public bulletin.
For each minimal qualified subset $\gamma_j = P_{j1}, P_{j2}, \ldots, P_{jd}$ of $\Gamma_0$, the dealer computes public information $T_j$ as follows.

Compute
$$H_j = K \oplus (U_j^{S_0} \mod N) \oplus (U_j^{S_0} \mod N) \oplus \ldots \oplus (U_j^{S_0} \mod N).$$

Use $d+1$ points
$$(0, H_j), (ID_{j1}, (U_j^{S_0} \mod N)), \ldots, (ID_{jd}, (U_j^{S_0} \mod N))$$
to construct a polynomial $f(X)$ of degree $d$.

$$f(x) = H_j \times \prod_{k=1}^{d} \frac{(X - ID_{jk}) / (-ID_{jk}) + (P_{jl}^{S_0} \mod N) \times (X / ID_{jl}) \times \prod_{k=1, k \neq l}^{d} \frac{(X - ID_{jk}) / (ID_{jl} - ID_{jk})}{\text{mod } Q}}$$

where $d$ is the number of participants in qualified subset $\gamma_j$.

Compute and publish $T_j = f(1)$ on the public bulletin.

In the recovery phase participants of any qualified subset can cooperate to recover the shared secret $K$ as follows.

Each participant gets $(U_0, h, N)$ from the public bulletin.

Each participant $P_{ji}$, computes and provides $S_{ji}' = U_0^{S_{ji}} \mod N$, where $S_{ji}'$ is the pseudo share of $P_{ji}$. If $S_{ji}' \equiv U_{ji} \mod N$, then $S_{ji}'$ is the true shadow else it is false and the participant $P_{ji}$ is the cheater.
6.4. RJH and CCH scheme

- Get $T_j$ from the public bulletin and use $d+1$ points $(1, T_j), (ID_{j1}, S'_{j1}), \ldots, (ID_{jd}, S'_{jd})$ for Lagrange interpolation to reconstruct the degree ‘$d$’ polynomial $f(X)$:

$$f(X) = T_j \times \prod_{k=1}^{d} \frac{(X - ID_{jk})/(1 - ID_{jk})}{\prod_{k \neq l} (X - ID_{lk})/(ID_{lk} - ID_{jk})} + \sum_{l=1}^{d} \left[(S'_{jl} \times (X - 1/ID_{jl} - 1) \times \prod_{k=1}^{d} (X - ID_{jk})/(ID_{jl} - ID_{jk}) \right] \pmod{Q}$$

- Compute $H_j = f(0)$ and recover the secret $K = H_j \oplus S'_{j1} \oplus S'_{j2} \oplus \cdots \oplus S'_{jd}$.

When new participants join the group, the access structure changes. The Dealer then performs the construction phase and publish the new public information. The older participants share remain the same. When the participants dis-enrolled, the corresponding minimal qualified subset should be deleted from the access structure. The shared secret should be renewed for security consideration. Public information must be changed in this case, but the rest of the authorized participants still hold the same shadows. Changing the shared secret can also be done by modifying the public values but the same shadows can be reused.

Adding a new subset can also be done easily. If the new qualified subset contains an old minimal qualified subset in the access structure, then nothing needs to be done. If the new access subset is a minimal qualified subset of some old set, the old ones shall be deleted from the access structure and the public information is updated according to the new access structure. Canceling a qualified subset needs the shared secret to be renewed. The public information corresponds to the rest of the
qualified subset must be modified. The public information corresponds to the canceled subset is of no use and is removed. It is noted that the Dealer does not need to collect the shadows of all the participants to reconstruct the secret sharing scheme again.

To share multiple secrets $K_1, K_2, \ldots, K_n$ with the access structure $\Gamma_1, \Gamma_2, \ldots, \Gamma_n$, each participant holds only one share $S_i$ for these $n$ secrets. For each shared secret $K_i$ the Dealer select a unique $S_i^0$ and publishes the corresponding $h_i, U_{0i}$. The Dealer also generate and publishes the information $T_{ij}$ for each qualified subset $\gamma_{ij}$ in minimal access structure $\Gamma_i$. The participants of each qualified subset $\gamma_{ij}$ in $\Gamma_i$ can cooperate to recover the shared secret $K_i$ by performing the recovery phase.

### 6.5 Sun’s Scheme

In Pinch’s scheme high computation overhead is involved and also sequential reconstruction is used in the recovery phase. In 1999 Sun [207] proposed a scheme having the advantages of lower computation overhead and parallel reconstruction in the secret recovery phase. The security of the scheme is only based on one-way function not on any other intractable problem.

Let $f$ be a one way function with both domain and range $G$. The following protocol is used to share $m$ secrets $K^{[h]}$ with access structures $\Gamma^{[h]}$ for $h = 1, \ldots, m$.

1. The Dealer randomly chooses $n$ secret shares $S_1, \ldots, S_n$ and send them to the participants through a secret channel.

2. For every shared secret $K^{[h]}$ and for every minimal qualified subset $X \in \Gamma^{[h]}_0$, the Dealer randomly chooses $R_X^{[h]}$ in $G$ and computes

$$T_X^{[h]} = K^{[h]} - \sum_{x: P_x \in X} f(R_X^{[h]} + S_x)$$
and publishes \( H^{[h]} = \{(R_X^{[h]}, T_X^{[h]})|X \in \Gamma^{[h]}_0\} \) on the notice board.

In order to recover the secret \( K^{[h]} \), a set of participants \( Y \in \Gamma^{[h]} \) proceeds as follows

1. The members of \( Y \) agree on a minimal qualified subset \( X \subseteq Y \), where \( X = \{P_1, \ldots, P_t\} \).

2. Each member \( P_i \) reads \( R_X^{[h]} \) from the notice board and computes \( f(R_X^{[h]} + S_i) \) and sends the result to \( P_t \), who is designated as secret re-constructor.

3. \( P_t \) receives \( f(R_X^{[h]} + S_i) \), for \( 1 \leq i \leq t - 1 \) and reconstructs the secret \( K^{[h]} = T_X^{[h]} + \sum_{i=1}^{t} f(R_X^{[h]} + S_i) \).

Once the secret is reconstructed, it become public. \( f(R_X^{[h]} + S_i) \) is unique for every secret and every authorized set. Most of the implementations of one way functions are based on permutations, substitution and XOR operation. Therefore the computation is much faster than the exponentiation. The step 2 of the reconstruction phase can proceed parallely, where as in Pinch’s scheme the construction is sequential. Cheating can be detected by putting additional information \( f(K^{[h]}) \) on the notice board for every shared secret. Any one can verify the correctness of the computed secret. The scheme can also detect cheaters by putting additional information \( C_X^{[h]} = f(f(R_X^{[h]} + S_i)) \) for every secret \( K^{[h]} \), every authorized set \( X \) and for every participant \( P_t \). The scheme is dynamic. Participants or new access structure can be added by distributing shares to the new participants and update public information on the notice board. The previously distributed shares remain valid. When some participants or some access structures need to be deleted, the shared secret should be renewed. The Dealer only need to update the information on bulletin board.
6.6 Adhikari’s Scheme

An efficient, renewable, multi use, multi-secret sharing scheme for general access structure is proposed by Angsuman Das and Avishek Adhikari [59] in 2010. The scheme is based on one way hash function and is computationally more efficient. Both the combiner and the participants can also verify the correctness of the information exchanged among themselves in this. The scheme consist of three phases. The Dealer phase, pseudo-share generation phase and the combiner’s phase.

Let \( \mathcal{P} = \{P_1, P_2, \ldots, P_n\} \) be the set of participants and \( S_1, S_2, \ldots, S_k \) be the \( k \) secrets to be shared by a trusted Dealer. Each secret is of size \( q \) bits. \( \Gamma_{S_i} = \{A_{i1}, A_{i2}, \ldots, A_{it}\} \) be the access structure corresponds to the secret \( S_i \) and \( A_{il} \) is the \( l \)’th qualified subset of the access structure of the \( i \)’th secret \( S_i \).

In the dealing phase, the Dealer \( D \) chooses a collision resistant one-way hash function \( H \), which takes as argument a binary string of arbitrary length and produces an output a binary string of fixed length \( q \), where \( q \) is the length of each secret. The Dealer also choose randomly \( x_\alpha \), the shares of size \( q \) and send to the participants through a secure channel.

In the pseudo share generation phase, a pseudo share corresponds to each secret and for each authorized set is generated from the participants secret share in the following way

\[
S_{ij} = S_i \bigoplus \left\{ \bigoplus_{\alpha: P_\alpha \in A_{ij}} H(x_\alpha \parallel i_l \parallel j_m) \right\}
\]

where \( i_l \) represent the \( l \) bit representation of the number of secret. i.e., \( l = \lfloor \log_2 k \rfloor + 1 \) and \( m = \lfloor \log_2 t \rfloor + 1 \), \( t \) is the maximum size of an authorized subset among the access structures corresponds to different secrets. The Dealer then publishes the values \( S_{ij}, H(S_j), H^2(x_\alpha \parallel i_l \parallel j_m) \).
In the combining phase, the participants of an authorized subset \( A_{ij} \) of \( \Gamma_{Si} \) submit the pseudo share \( H(x_\alpha \parallel i_l \parallel j_m) \). The pseudo share is then XOR with \( S_{ij} \) to get the secret \( S_i \) by the combiner.

\[
S_i = S_{ij} \bigoplus \left\{ \bigoplus_{\alpha : \alpha \in A_{ij}} H(x_\alpha \parallel i_l \parallel j_m) \right\}
\]

The combiner can verify the pseudo share given by the participant by checking it with the public value \( H^2(x_\alpha \parallel i_l \parallel j_m) \). The participants can check whether the combiner is giving them back the correct secret \( S_i \) by verifying it with the public value \( H(S_i) \).

Adhikari and Roy \cite{180} also proposed a similar scheme with polynomial interpolation. In this scheme, for each authorized subset in the access structure corresponds to a secret, a polynomial of degree \( m-1 \) is created with the constant term as the secret \( S_i \), where \( m \) is the number of participants in the authorized subset.

\[
f_q^{S_i}(x) = S_i + d_1^{i_q}x + d_2^{i_q}x^2 + \ldots + d_{m^{iq}-1}^{i_q}x^{m^{iq}-1}
\]

For each participant \( P_{iq}^b \in A_{iq} \) in \( \Gamma_{Si} \), the Dealer compute pseudo share \( U_{iq}^b = h(x_{iq}^b \parallel i_l \parallel q_m) \), where \( x_i \) is the secret share of the participant and \( i = 1, \ldots, k; q = 1, \ldots, l; b = 1, \ldots, m \). The Dealer also computes \( B_{iq}^b = f_q^{S_i}(ID_{iq}^b) \). Finally the shift values are computed and published corresponds to each secret and each authorized subset \( M_{iq}^b = B_{iq}^b + U_{iq}^b \).

In the reconstruction phase, the pseudo shares of authorized set of participant can be added with the public information to obtain \( B_{iq}^b = f_q^{S_i}(ID_{iq}^b) = M_{iq}^b + U_{iq}^b \). The secret can be reconstructed by interpolation using these \( m \) values.

\[
S_i = \sum_{b \in \{1,2,\ldots,m_{iq}\}} B_{iq}^b \prod_{r \in \{1,2,\ldots,m_{iq}\}, r \neq b} \frac{-ID_{iq}^b}{ID_{iq}^b - ID_{iq}^r}
\]
It is noted that the computational complexity is more in this case, compared with the previous scheme.

### 6.7 An Efficient Multi Secret Sharing with General Access Structure

The scheme is based on Shamir and the hardness of the Discrete Logarithm Problem (DLP). The participant has to keep only a single share for sharing multiple secret. The shares are generated by the participants and send it to the Dealer. Hence there is no need for a secure channel between the Dealer and the participant. The pseudo shares are sent to the Dealer and it is difficult to get the shares from the pseudo shares because of the complexity of the discrete logarithm problem. Shared secret, participants set and the access structures can be changed dynamically without updating participants secret share. The degree of the polynomial used in Shamir’s scheme is only one, so the computational complexity is also less.

The proposed secret sharing have three phases.

1. Initialization
2. Secret Sharing
3. Secret Reconstruction

These phases are explained in detail.
6.7. An Efficient Multi Secret Sharing with General Access Structure

6.7.1 Initialization Phase

Let \( P = P_1, P_2, \ldots, P_n \) be the set of participants. \( K_1, K_2, \ldots, K_k \) be the set of secrets to be shared according to the access structure \( \Gamma_1, \Gamma_2, \ldots, \Gamma_k \), where \( \Gamma_i = \{ \gamma_{i1}, \gamma_{i2}, \ldots, \gamma_{it} \} \) is the access structure corresponds to the secret \( K_i \).

- Select two large prime \( p \) and \( q \) and let \( n = p \times q \).
- Select an integer \( g \) from \([\sqrt{n}, n]\) such that \( g \neq p \) or \( g \neq q \) and is a generator.
- Choose another prime \( m \) larger than \( n \). The Dealer publishes \( g, n, m \) on the public bulletin.
- Each participant randomly select an integer \( s_i \) from \([2, n]\) as secret share and compute \( ps_i = g^{s_i} \pmod{n} \) corresponds to each secret \( K_i \).
- The pseudo shares \( ps_i \) are send to the Dealer, who will then publish them in the public bulletin board.

6.7.2 Secret Sharing

In this phase, the Dealer will share the secrets corresponds to each access structure by publishing the values in the bulletin board, which is used by the participants to later reconstruct the secret.

- Dealer randomly select an integer \( s_0_i \) from \([2, n]\) such that \( s_0_i \) is relatively prime to \( \phi(n) \) and compute \( ps_0_i = g^{s_0_i} \pmod{n} \) corresponds to each secret \( K_i \).
- Find \( h0_i \) such that \( s_0_i \times h0_i \equiv 1 \pmod{\phi(n)} \).
Multi Secret Sharing

- Select an integer \( a \) from \([1, m - 1]\) and construct a polynomial
  \( f_i(x) = K_i + a \times x \pmod{m} \).
- Select \( t \) distinct random integers from \( d_{i1}, d_{i2}, \ldots, d_{it} \) from \([1, m - 1]\)
to denote the \( t \) qualified sets in \( \Gamma_i \).
- Compute \( f_i(1) \) and for each subset \( \gamma_{ij} = \{P_{ij}, P_{2ij}, \ldots, P_{lij}\} \) compute
  \[
  H_{ij} = f_i(d_{ij}) \bigoplus ps_1^{s_0}, \pmod{n} \bigoplus ps_2^{s_0}, \pmod{n} \bigoplus \ldots \bigoplus ps_t^{s_0} \pmod{n}
  \]
- The Dealer then publish
  \( ps_0, h_0, f_i(1), H_{i1}, H_{i2}, \ldots, H_{it}, d_{i1}, d_{i2}, \ldots, d_{it} \)
corresponds to each secret \( K_i \) and the access structure \( \Gamma_i \).
- The Dealer also publishes \( F(K_i, d_{ij}) \) corresponds to each secret and
each authorized access set which can be used by the participant for
verification after the secret recovery, where \( F \) is a two variable one
way function.

6.7.3 Secret Reconstruction

The participants from any authorized subset \( (\Gamma_i) \) can reconstruct the secret
\( K_i \) as follows.

- If \( \gamma_{ij} = \{P_{1ij}, P_{2ij}, \ldots, P_{lij}\} \) want to reconstruct \( K_i \), each participant
  compute \( x_{kij} = ps_0^{s_k}, k = 1, \ldots, l \). These values are then delivered to
  the designated combiner.
- The combiner computes
  \[
  f_i(d_{ij})t = H_{ij} \bigoplus x_{1ij} \bigoplus x_{2ij} \bigoplus \ldots \bigoplus x_{lij}
  \]
6.7. An Efficient Multi Secret Sharing with General Access Structure

Using \( f_i(1) \), \( f_i(d_{ij}) \) and \( d_{ij} \)'s, he can reconstruct the polynomial and hence recover the secret.

\[
\begin{align*}
  f_i(x) &= f_i(1) \times \frac{(x - d_{ij})}{1 - d_{ij}} + f_i(d_{ij}) \times \frac{(x - 1)}{d_{ij} - 1} \\
  &= \frac{x \times f_i(1) - d_{ij} \times f_i(1) - x \times f(d_{ij}) + f(d_{ij})}{1 - d_{ij}}
\end{align*}
\]

- The shared secret \( K_i = f_i(0) \).
- Each participant of the authorized set can exchange \( x_{ij} \) with other participants in the group and each member can compute the secret individually. This doesn’t need a specified combiner and it also avoids the transmission of secret from the combiner to the participants.
- Each participant can verify the given \( x_{ij} \) by the other participants and also the recovered secret by using the public values.

6.7.4 Analysis and Discussions

In the proposed scheme, the degree of the used Lagrange polynomial \( f(x) \) is only 1 and we can construct \( f(x) \) very easily. The other operation is just XOR operation which can also be computed very efficiently. Each participant select his share and compute the pseudo share \( ps_i = g^{s_i} \pmod{q} \). This avoids the computational quantity of the Dealer. This also avoids the need for a secure channel.

The proposed scheme does not need special verification algorithm to check whether each participant cheats or not. In the secret reconstruction phase, the combiner can check whether \( x_i \) is a true share by checking \( x_i^{h_{0i}} = ps_i \pmod{m} \). That is \( x_i^{h_{0i}} = (ps_i^{s_i})^{h_{0i}} = (g^{a_i h_{0i}})^{s_i} = g^{s_i} = ps_i \pmod{m} \). Each participant can verify the secret after recovery by computing the two variable one way function \( F(K_i, d_{ij}) \) and compare the result with the public value.
Table 6.1: Comparison of multi secret sharing schemes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>share size same as secret</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>use of one way function</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>use of discrete logarithm</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>use of interpolation</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>shares remain secret during reconstruction</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>dealer knows the share</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>shares can be reused</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>dynamic</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>verifiability</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In the reconstruction phase, each participant $P_{ij}$ in $\gamma_{ij}$ only provides a public value $x_{ij}$ and he does not have to reveal the secret share $s_i$. It is difficult to get the secret share from the public value $x_{ij}$ and $ps_i$, because the discrete logarithm problem is hard to solve. The scheme is computationally secure. The shares can be reused and hence the scheme is a multi use scheme. The polynomial $f(x)$ can be reconstructed only if two points are known. The point $(1, f(1))$ is known publicly but the second point can be obtained only by the authorized set of participants using their private shares.

The important property of the proposed scheme is that the shared secret, the participant set and the access structure can be changed.
6.7. An Efficient Multi Secret Sharing with General Access Structure

dynamically without updating any participant’s secret shadow. In order
to update the secret, the Dealer need to create a new polynomial \( f(x) \)
and update \( f(1) \). If a new qualified set is to be added then \( H_{t+1} \) and \( d_{t+1} \)
need to be added. New participant can be added accordingly. The public
information corresponds to each modified authorized set must be
recomputed and the old information must be updated in the public
bulletin. Deleting a participant or deleting the authorized set containing
the participant needs, deleting the public information corresponding to
the access set. However for security reasons the secret also need to be
updated. The scheme has following important properties.

1. The scheme can share multiple secrets, each with a specified access
structure.

2. The participant has to hold only a single share in order to share
multiple secrets.

3. The size of the share is as short as the secret.

4. Participants select their secret shares and the Dealer need not know
the shares of the participants. This avoids the need of a secure
channel.

5. The scheme is multi use i.e., the participants can reuse the shares
after a secret is recovered.

6. Each participant can verify the shares provided by the others in the
recovery phase.

7. The Dealer can modify the secret or add new secret with out
modifying the participants secret shadow.

8. After the secret is recovered, the participants can verify the validity
of the recovered secret.
9. The access structures can be dynamically modified. Only the public
values need to be modified in this case also.

The table 6.1 summarize and compares the important properties of
existing schemes and the proposed scheme.

6.8 Concluding Remarks

In this chapter we give a brief summary of the important constructions
for multi-secret sharing having threshold and generalized access
structures. We explore more on multi secret sharing realizing general
access structure. The important technique used for the constructions are
based on one way functions, discrete logarithm problem and Shamir’s
secret sharing technique. The schemes based on discrete logarithm
problem and hash functions provide only computational security because
the security depends on the computational complexity of these problems.
But for many of the cryptographic application with polynomial time
bounded adversary, the computational security is sufficient. For
maintaining the unconditional security, large number of shares must be
kept by the participant. The number of shares that must be kept is
proportional to the number of secret to be shared.

The public values in the bulletin board of each scheme is proportional
to the number of authorized subset in an access structure corresponds to
each key. There will be at least one public value corresponds to each
authorized subset in the access structure corresponds to a key. There are
also additional public parameters used for the security of the scheme. The
computational complexity depends on the complexity of the one way
function used or the modular exponentiation. But these operations can be
efficiently done in polynomial time. The most commonly used one way
functions like LFSR, MD5, SHA are all based on simple XOR,
permutation and substitution operation. So these schemes can be implemented in polynomial time. Modular exponentiation is time consuming with large exponent but efficient algorithm exist for the fast computation. The share generation and reconstruction in the Shamir’s scheme, which uses polynomial interpolation can also be implemented efficiently.

All the scheme mentioned assumes that the Dealer is a trusted person. Cheating detection mechanisms are also proposed in some schemes with the help of additional public parameters. The combiner can verify the share submitted by the participants and the participant can also check the reconstructed secret. However the security is computational. If the computational problem is solved, the secret can be revealed by an adversary. The mathematical model, security notions and computational security for multi-secret sharing is proposed by Javier Herranz et al [95] [96] in 2013.

The major concern in the multi-secret sharing is the large number of public values and the computational complexity. Only computational security can be achieved in all the schemes mentioned, where security depends on the security of some computationally hard problem. Multi-secret sharing schemes have found numerous application in implementing authentication mechanisms, resource management in cloud, multi policy distributed signatures, multi policy distributed decryption etc.

In this chapter, we also give an efficient construction of a multi secret sharing scheme with generalized access structure. The scheme is multi use and hence the shares can be reused by the participants. The participant select their secret shadows and the secret can be reconstructed by any participant in the authorized subset. No secure channel is required because the secrets or the secret shares are never send through the channel. The scheme is also verifiable because each participant can verify the shares of the
Multi Secret Sharing

other participants during the reconstruction phase and also the participants can verify the reconstructed secret. The shared secret, access structure or the participants set can be dynamically modified without modifying the participants secret shadow. The scheme is also computationally efficient and can be implemented easily.