Secret Sharing Schemes with Extended Capabilities and Applications

Thesis submitted to

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

in partial fulfillment of the requirements

for the award of the degree of

DOCTOR OF PHILOSOPHY

under the Faculty of Technology by

Binu V. P
Register No:4133

Under the guidance of
Dr. A. Sreekumar

Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India.

August 2016

Secret Sharing Schemes with Extended Capabilities and Applications

Ph.D. thesis

Author:
Binu V. P
Research Scholar
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India.
Email: binuvp@gmail.com

Research Advisor:
Dr. A. Sreekumar
Associate Professor
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India.
Email: askcusat@gmail.com

Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, Kerala, India.

August 2016

To My Dear Teachers

&

Loving Family

Dr. A. Sreekumar
Associate Professor
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, India.

26^{th} August 2016

Certificate

Certified that the work presented in this thesis entitled "Secret Sharing Schemes with Extended Capabilities and Applications" is based on the authentic record of research carried out by Shri. Binu V. P under my guidance in the Department of Computer Applications, Cochin University of Science and Technology, Kochi-682 022 and has not been included in any other thesis submitted for the award of any degree.

A. Sreekumar
(Supervising Guide)

Phone : +91 484 2577602 +91 484 2556057 Email: askcusat@gmail.com

Dr. A. Sreekumar
Associate Professor
Department of Computer Applications
Cochin University of Science and Technology
Kochi - 682 022, India.

26^{th} August 2016

Certificate

Certified that the work presented in this thesis entitled "Secret Sharing Schemes with Extended Capabilities and Applications" submitted to Cochin University of Science and Technology by Sri. Binu V. P for the award of degree of Doctor of Philosophy under the faculty of Technology, contains all the relevant corrections and modifications suggested by the audience during the pre-synopsis seminar and recommended by the Doctoral Committee.

A. Sreekumar
(Supervising Guide)

Phone : +91 484 2577602 +91 484 2556057 Email: askcusat@gmail.com

Declaration

I hereby declare that the work presented in this thesis entitled "Secret Sharing Schemes with Extended Capabilities and Applications" is based on the original research work carried out by me under the supervision and guidance of Dr. A. Sreekumar, Associate Professor, Department of Computer Applications, Cochin University of Science and Technology, Kochi–682 022 and has not been included in any other thesis submitted previously for the award of any degree.

Binu V. P

Kochi– 682 022
26^{th} August 2016

Acknowledgment

My first and foremost heart-felt gratitude goes to my wonderful supervisor Dr. A. Sreekumar, who introduced me to the world of 'Cryptography'. I consider myself very fortunate to have him as my advisor. I have learned a lot from him over the past few years. The time spent with him and the conversation we had really boosted my spirits and ignited the passion of learning. Many foundation stones of my dream are laid by him. He has also been a fun and enthusiastic partner to discuss various puzzles and riddles.

I am greatly thankful to Dr. B. Kannan, Head of The Department, Computer Applications, CUSAT, for the motivation, support and guidance. His expertise in taking out the best out of a student is beyond appreciation. His inspiration can lift any student to excellence. His guidelines in my initial days was of immense help for me. I would also like to thank him for being very patient with me and for having faith in me. Furthermore, I thank him for providing an excellent research atmosphere at the Research Lab.

I sincerely thank Dr. K. V. Pramod former HOD, DCA, CUSAT for his insightful suggestions, fruitful discussions and critical remarks. His dedication and energy are infectious. His jovial and affectionate nature is memorable. I will be forever indebted to him.

I also would like to acknowledge the two backbones of the department Dr. M. Jathavedan and Prof. S. Malathi for the constant encouragement and support. The tea time we had together in the evening was a real place for fruitful discussions on several philosophical topic.

I would like to express my gratitude to Dr. G. Santhosh Kumar, Department Of Computer Science, CUSAT, with whom I had many insightful discussions, which have bettered my understanding of various topics. He is a constant supporter and doesn't have any hesitation in lending his helping hand for students. I also took a course subject under

his guidance. We had several joint work along with two of his MTech students Mrs. Divya G. Nair and Mrs. Sreela S. R. Their contribution in implementing some of the key algorithms are remarkable.

Several other people from whom I draw lot of inspirations are Prof. Shine N. Das (CE, Munnar), Prof. Santhosh Kumar M. B (Department of IT, CUSAT) and Prof. Prasanth G. Narasimha Shenoi (Govt College, Chittur).

I am also thankful to other faculty members, office staff, librarian and non teaching staff of the DCA, who helped me during various stages of my research.

It has been a delight working in Research Lab of DCA. Thanks to the lively ambiance maintained by the past and present students of the lab. I enjoyed every bit of my life in this lab which have created many great researchers in the past. The lab staffs were very cooperative. They have extended their helping hand in many occasions. My lab mates Bino, Ramkumar, Sunil, Jashir, Syam, Justin, Arun, Vinu, Sukrut and Vijith deserve special acknowledgments. I also acknowledge my ex-lab-mates Jomy, Simily, Jessy and Remya. My every moment of research and personal life has been a cherishing experience in their company.

I also acknowledge the tremendous support of my family during this endeavor. My parents K. M. Vijayamma and H. Purushothaman, who have remarkable influence on shaping my career. I always find a perfect companion in my wife Padma who is truly a pillar of support for me. Moreover the endless love of my cute little Ammu and Sambu inspires life a lot.

Above all, I thank the supreme power who created the universe and gave the mankind the supreme knowledge.

Binu V. P

Preface

This dissertation deals with the development of secret sharing schemes with several extended capabilities and also their applications. We have considered schemes with single and multi secret sharing. Schemes developed are also having extended capabilities like verifiability cheating detection and cheater identification. Simple schemes based on number theory and XOR operations are developed which are useful for several applications. Elliptic curve and pairing based multi secret sharing schemes are developed which are more secure. Threshold and Generalized access structure is realized. Two prominent applications are considered in which secret sharing based solutions provides a better alternative compared with cryptographic techniques. E-voting using Secure Multiparty Computation and CTS (Cheque Truncation System) using secret image sharing techniques are the two applications developed.

Security is a big challenge in the recent scenario where all of us connected to a public network and the data are usually stored on large servers of service provider instead of in the owner's machine. Any body can steal important data of an organization which is available in a public place. Even the service provider itself is not trustable in many situations. But business organization need to protect data from disclosure. One way to

protect secret information is by using conventional encryption. But what happens when the encrypted information is corrupted or when the secret key is lost. This means that there is only security but there is no reliability. Secret sharing address this problem and finds solutions for both security and reliability. Instead of storing the valuable data in a single place, it is distributed and stored at several places. When the need arises they can be reconstructed from the distributed shares with fault tolerance.

The original motivation of secret sharing was to safeguard cryptographic keys from loss. The loss of a cryptographic key is equivalent to data loss as we cannot retrieve the original data back with out the encryption key. It is desirable to create backup copies of important keys but greater the number of copies made greater the risk. Secret sharing provides an efficient solution to this problem by protecting important information being lost, modified, destroyed or getting into wrong hands.

The idea of *secret sharing* is to start with a secret and divide it into pieces called *shares* or *shadows*, which are distributed amongst users such that the pooled shares of designated subsets of users allow reconstruction of the original secret.

A particularly interesting class of secret sharing schemes is *threshold scheme* for which the designated sets consist of all set of t or more participants. Such schemes are called t out of n *threshold schemes* or simply (t, n) schemes, where n is the total number of participants. Another class in which any authorized subset of participants can collate and access the secret data are called *generalized secret sharing* schemes.

A secret sharing scheme may be served as a *shared control scheme* if shares from two or more users are required to enable a critical action such as opening a bank vaults, launching a nuclear missile etc. Reduced trust is the reason behind this as we want to distribute the trust among many users. This enhances availability and confidentiality. Secret sharing schemes

are found numerous applications when shared control is required such as sharing a key to open a secret (key escrow/key back up). A major drawback of public key cryptography is the dominance of a certain authority, therefore we wish to allow several authorities to participate in the creation of keys, distributing them, signing them etc. Based on this several cryptographic protocols have come up such as shared signature, threshold encryption, threshold decryption etc.

Real-world applications require more capabilities than threshold schemes can offer. We may wish to have a more complicated list of authorized coalitions than just the subset with t or more participants. In this situation more generalized schemes are introduced. In these schemes secret can be retrieved when the authorized set of participants as mentioned in the access structure collate together. It is noted that the secret sharing schemes are not secure. An untrusted dealer may send invalid shares or the participant may send wrong shares during the reconstruction phase. Verifiability, cheater detection and identification are the major requirement in any secure secret sharing scheme. The capability to retrieve several secrets when the authorized set of participant collate is an added advantage and has got several real life applications. These schemes are called multi secret sharing scheme. Major contribution of this thesis is in the development of secret sharing schemes with these extended capabilities having threshold and generalized access structure.

The use of Elliptic curve in cryptography have made a significant advancement and provides more security with less computational power. There is not much work done in the area of secret sharing, where elliptic curve can be effectively utilized. We explore the fundamentals of elliptic curve and then an important construct called Bilinear pairing, which can be effectively utilized to build secret sharing schemes with several extended capabilities.

Secret sharing schemes are highly versatile cryptographic primitives and are employed in vast range of real world applications which include secure storage of electronic data, electronic voting, online auctions, secure multiparty computation, generalized oblivious transfer, broadcast encryption, visual cryptography etc. We have considered secret sharing based secure multi party computation for e-voting and secret image sharing method to develop two important applications in this area.

In brief, our work in this thesis has made significant advancement in the state-of-the-art research on multi secret sharing. There is only a little contribution in the literature mentioning the use of elliptic curve and pairing in secret sharing. Thus our work has made a significant contribution to the field of multi secret sharing using elliptic curve and pairing. The applications developed are also provide a better alternatives compared with the existing schemes which are based on computationally complex cryptographic techniques.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Secret sharing schemes are developed as a technique to safe guard cryptographic keys. Later it has found several useful applications in the various cryptographic protocols. The widespread use of cloud computing makes the distributed storage at remote cloud servers. The secure storage and distribution of data on third party servers pose serious threat when the service provider itself is not trustable. Encryption decryption makes large amount of computational overhead and also complicated key management. Multiple encryptions are the major issue when we want to share a document with several users. The secret sharing based key management and access policies help to keep document secure by encrypting according to a policy defined using user attributes. This is a very prominent applications used now a days in Cipher text Policy Attributed Based Encryptions (CPABE) over cloud. Another research area where the secret sharing schemes have found useful applications is secret image sharing. Secure storage and transmission of confidential images like medical images can be done with out using encryptions by secret sharing technique. Secure multi party computations where secret sharing technique and homomorphism of secret sharing plays a major

role. Oblivious transfer, secure key distribution, implementation of effective access control mechanism, threshold encryption decryption, threshold signature generation, broadcast encryption are all major areas where the secret sharing schemes are used as the basic building blocks.

This dissertation deals with development of efficient secret sharing schemes having several extended capabilities. These schemes are based on number theoretic concepts, XOR operations and elliptic curves. They are simple and can be easily implemented. These schemes can be used efficiently in several application scenarios. Multi secret sharing with verifiability, cheating detection and cheater identification are the major extended capabilities achieved. Both threshold and generalized access structure based secret sharing schemes are considered. We have done an investigation on the use of elliptic curve and pairing for the construction of secure secret sharing schemes and developed generalized and threshold multi secret sharing schemes based on them. Two prominent applications are also developed using the secret sharing techniques.

In this chapter we start with the motivation and the problem definition. The basic secret sharing schemes are reviewed then, which helps in understanding the core concept and developments in this area of study. We then emphasize on major contributions of the thesis. Lastly, we describe the chapter wise organization of this thesis.

1.1 Motivation

Liu in [136] considers a combinatorial problem. There are eleven scientists working on a secret project. They want to enforce more security and avoids individual trust by keeping the secret documents of the project in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. Two questions arise from this problem are, what is the minimum number of locks which are needed and minimum number of

keys each scientist must carry for achieving security. He stated that since for every group of five scientist there must be a dedicated lock which they cannot open. It results in $\binom{11}{5}$=462 locks. A scientist has to combine with five other to open a lock. There are $\binom{10}{5}$=252 ways of choosing five scientist out of ten, which shows the minimum number of keys each scientist must carry. It is noted that even for this small problem, the solution requires large number of locks and keys which is not feasible.

So the motivation behind the development of secret sharing scheme is to share a secret among n participants in such a way that t or more of them (where $t \leqslant n$) can join together to retrieve the secret. Shamir and Blakley in 1979 independently suggested non mechanical solutions to this problem. Shamir's scheme is based on polynomial interpolation, where as Blakley's scheme is based on geometry. Shamir's scheme is perfect and ideal, where as Blakley's scheme is not so. In Shamir scheme each participant has to keep only one share which is of same size as the secret (ideal) and also less that t participant cannot deduce any information about the secret (perfect). These schemes are called (t, n) threshold schemes.

The motivation was to safeguard cryptographic keys. The security of the secret key used in cryptography were very important. The key kept in a single location is highly unreliable since a single misfortune may make the information inaccessible. Computer breakdown, sabotage, sudden death of a person knowing the secret etc may leads to this situation. An obvious solution is to store the keys at multiple locations. But this makes the situation even worst and provides lot of opportunities for hackers. The secret sharing based solution provided a perfect key management where less than t pieces of information doesn't give any information about the secret. Even if $n - t$ shares are corrupted, the secret key can be recovered. This provides both secrecy and reliability.

Other than security and reliability there is a trade off between safety and convenience in using several applications. For example, a company wants to

digitally sign all its documents. If all the executives are given the companies signature it is convenient, but the signature may be misused. The solution is to distribute the keys in such a way that only when certain specified number of participant collate, they can sign a document. This provides more safety. So when the participants of conflicting interest have to collate and also they are mutually suspicious, the secret sharing schemes are the ideal solution. We can also give more power to individuals by giving more shares. There are several critical applications where collective controls are needed rather than individual control such as opening a bank vault, launching a missile etc.

Another reason for the study and development of secret sharing schemes is the reduced trust. Key escrow system of US government are broken and misused. Secret sharing based solution provides a secure way to distribute the secret key among the authorities and the key is reconstructed based on a court order. The security of the outsourced data is also critical, when the service provider itself is untrusted. The distributed storage using secret sharing based mechanism helps to avoid the single point of failure. The access control policies can also be established using secret sharing based key distribution. Cipher-text Policy Attribute Based Encryption(CPABE) is a way to encrypt documents according to a policy described using an access structure. The access structure is specified by using the attributes of the user. The key is then shared according to the access structure using secret sharing technique. If the user key attributes matches with the encryption policy, he will be able to decrypt the encrypted document.

Key agreement among the parties are also important when the shared data need to be accessed by several people. Broadcast encryption, where an encrypted document can only be decrypted by certain set of participant is widely used in digital broadcast. Secret sharing provides an efficient solution for the same. Secure multi party computation is another motive. Several parties jointly compute certain functions with out

revealing ones own secret. Secret sharing based solutions are very effective in implementing the secure computation among parties using participants share. E-voting is a special case of secure multi-party computation. Secret sharing based e-voting schemes are less complicated and computationally efficient compared with the homomorphic encryption schemes used for e-voting. Another interesting and widely used application is the use of secure storage and transmission of confidential images. Secret image sharing based technique provides efficient solution compared with the conventional image encryption.

1.2 Review of Secret Sharing Schemes

A secret sharing scheme is a method of protecting a secret among a set of participants in such a way that only certain specified subset of participants can reconstruct the secret. The secret sharing scheme is initialized by a trusted Dealer by making shares of information related to the secret called *shares* or *shadows*. The shares are then send securely to each participants. Authorized subset of participants (defined by the *access structure*) can collaborate and reconstruct the secret by pooling of their shares. So the secret sharing process mainly consist of two stages *Share Distribution* and *Secret Reconstruction*.

In *Share Distribution*, there is a trusted Dealer ($\mathcal{D}$) who generates the shares of the secret and sends it securely to the participants. The secret is then destroyed.

In *Secret Reconstruction*, the participants belong to a qualified set can pool their shares and reconstruct the secret. We can also consider the case where the participants belonging to a qualified set submit their shares to a trusted combiner. The combiner then compute the secret and send it to the participants.

The access structure specifies the qualified sets of participant, who can retrieve the secret. Secret sharing scheme partitions the set of all participants into two. Those who are able to retrieve the secret called *authorized sets* and those who are unable to recover the secret called *unauthorized sets*. Most of the schemes consider the access structure with *monotone property*. This intuitively means that if a group can recover the secret, so can a larger group. In the case of an unauthorized group, if a group cannot recover the secret, neither can a smaller group. That is given a subset of participants which form an authorized set then any super set of this set will also be an authorized set.
The following are the two fundamental requirements of any secret sharing scheme.

- **Recoverability:** Authorized subset of participants should be able to recover the secret by pooling their secret shares.
- **Privacy:** Unauthorized subset of participants should not learn any information about the secret by combining their shares.

Development of secret sharing scheme started as a solution to the problem of safeguarding cryptographic keys. The cryptographic keys are very important in security. The encrypted data cannot be retrieved back if the key is lost. The storage of key at a particular location makes the key to be tampered or hacked by an intruder. There is a single point of failure. Secret sharing provides a robust key management scheme that is secure and reliable. The secret key is secured by distributing it among n participants and t or more of the participants can recover it by pooling the shares. Thus the authorized set is any subset of participants containing more than t members. This scheme is denoted as (t, n) *threshold scheme*. An attacker has to destroy at least $n - t + 1$ pieces or the security breaches need exposure of t pieces. Knowledge of less than t pieces will not reveal any information about the secret.

Definition 1.2.1. Let t, n be positive integers with $t \leqslant n$. A (t, n)-threshold secret sharing scheme is a method of sharing secret K among a set of n participants in such a way that any t participants can compute the value of K, but no group of less than t participant can get any information about K.

Threshold secret sharing schemes were introduced independently by Shamir [190] and Blackley [24] and since then much work has been put into the investigation of such schemes. Linear construction were most efficient and widely used. A threshold secret sharing scheme is called *perfect*, if less than t shares give no information about the secret. Shamir's scheme is perfect while Blakley's scheme is non perfect. Both the Blakley's and the Shamir's constructions realize t-out-of-n shared secret schemes. However, their constructions are fundamentally different. Polynomial based constructions are used by Shamir where as Vector space constructions are used by Blackley in their seminal paper.

McEliece and Sarwate [144] made an observation that Shamir's scheme is closely related to Reed-Solomon codes [177]. The error correcting capability of this code can be translated into desirable secret sharing properties. Karnin et al [117] realize threshold schemes using linear codes. Massey [143] introduced the concept of minimal code words and provided that the access structure of a secret sharing scheme based on a $[n, k]$ linear code is determined by the minimal codewords of the dual code.

Number theoretic concepts are also introduced for threshold secret sharing scheme. The Mingotee scheme [146] is based on modulo arithmetic and *Chinese Remainder Theorem (CRT)*. A special sequence of integers called Mingotte sequence is used here. The shares are generated using this sequence. The secret is reconstructed by solving the set of congruence equation using CRT. The Mingotte's scheme is not perfect. A

perfect scheme based on CRT is proposed by Asmuth and Bloom [2]. They also uses a special sequence of pairwise co-prime positive integers.

Kothari [123] gave a generalized threshold scheme. A secret is represented by a scalar and a linear variety is chosen to conceal the secret. A linear function known to all trustees is chosen and is fixed in the beginning, which is used to reveal the secret from the linear variety. The n shadows are hyperplanes containing the liner variety. Moreover the hyperplanes are chosen to satisfy the condition that the intersection of less than t of them results in a linear variety, which projects uniformly over the scalar field by the linear function used for revealing the secret. Thus as more shadows are known more information is revealed about the linear variety used to keep the secret, however no information is revealed until the threshold number of shadows are known. He had shown that Blakley's scheme and Karnin's scheme are equivalent and provided algorithms to convert one scheme to another. He also stated that the schemes are all specialization of generalized linear threshold scheme. Brickell [34] also give a generalized notion of Shamir and Blackley's schemes using vector spaces.

Researchers have investigated (t, n) threshold secret sharing extensively. Threshold schemes that can handle more complex access structures have been described by Simmons [199] like weighted threshold schemes, hierarchical scheme, compartment secret sharing etc. They were found a wide range of useful applications. Threshold schemes are also developed based on orthogonal arrays [60], graph decompositions [27], matrix projection [5] etc. Sreekumar et al [201] in 2009, developed threshold schemes using POB (Permutation Ordered Binary) system based on Visual cryptography.

Shamir [190] discussed the case of sharing a secret between the executives of a company such that the secret can be recovered by any three executives, or by any executive and any vice-president, or by the

president alone. This is an example of *hierarchical secret sharing* scheme. The Shamir's solution for this case is based on an ordinary $(3, n)$ threshold secret sharing scheme. Thus, the president receives three shares, each vice-president receives two shares and finally every executive receives a single share. The above idea leads to the so-called weighted (or multiple shares based) threshold secret sharing schemes. In these schemes, the shares are pairwise disjoint sets provided by an ordinary threshold secret sharing scheme. Benaloh and Leichter have proven in [15] that there are access structures that can not be realized using such scheme.

A more general construction is based on, which subset of participants can reconstruct the secret and which subset cannot. Realizing the secret sharing schemes for an arbitrary access structure is considered by Ito et al [107]. It is based on Shamir's scheme. The idea is to distribute shares to each authorized set of participants using multiple assignment scheme, where more than one share is assigned to a participant, if he belongs to more than one minimal authorized subset. Efficient schemes realizing the arbitrary access structure is developed by several authors later. Vector space construction [35], combinatorial design [128], linear block codes [20], matroids [26] and cumulative arrays [107] are the most suggested construction for generalized secret sharing schemes.

Brickell and Davenport [36] developed generalized secret sharing scheme based on Matroid theory. Simmons had done considerable research in secret sharing schemes based on geometry technique [199]. But the implementation of these schemes are not efficient. Generalized schemes based on Chinese Remainder Theorem and determinants are developed by Iftene [103] [105].

A major problem with secret sharing based on generalized access structure is that, the size of the share is exponential in the number of parties in the access structure. So we have to minimize the information, different users hold as their share.

Secret sharing schemes have been studied in an *information-theoretic* security model where the security is independent of the computing capabilities of an adversary. This can however be relaxed and some schemes have been defined for *computationally secure* models where the schemes relies on the difficulty of mathematical problem. The information-theoretic security model permits a notion of *perfect privacy*. In *perfect secret sharing* scheme, unauthorized sets do not learn any information about the secret via their shares.

The efficiency of the secret sharing scheme is measured with *information rate*. The *information rate* of a secret sharing scheme is the ratio of the size of the secret with the size of the largest share. Most of the perfect secret sharing scheme for general monotone access structures are *linear secret sharing schemes*. In linear secret sharing scheme the secret is computed as a linear combination of any set of shares. Perfect schemes for which information contained in the share equal to information contained in the key are called *ideal*. *Ideal* schemes do not exist for all monotone access structures.

Traditional secret sharing model does not consider the malicious behavior of the Dealer or the Participants. The traditional model assumes that there is a Trusted Dealer and Honest Participants completely following the protocol. A passive adversary is considered who can capture shares, but shares are not corrupted. But in the real time scenario both the Dealer and the participants may misbehave. The Dealer may give inconsistent shares to the participants, from which they will not be able to reconstruct a secret. The participant may also cheat by giving wrong shares during the reconstruction. In this attack only the malicious participant will be able to learn the correct secret where as others will get wrong secret value.

Verifiable secret sharing (VSS) schemes [53] address the malicious behavior of the Dealer. The protocol allows the participant to verify that

consistent dealing is performed by the Dealer. The share holders can convince that the shares are t consistent, meaning that every subset of t shares out of n defines the same secret. There are two types of verifiable secret sharing protocols. Interactive proof and Non Interactive proof. These protocols allows shares to be verified without revealing the shares. Benaloh proposed the interactive verifiable secret sharing scheme based on secret sharing homomorphism [17] [18]. Non interactive schemes are proposed by Feldman [71] and Pedersen [167]. In this scheme, the share holders will not communicate with other share holders or with the Dealer. The Dealer sends extra information to each participant during the distribution of shares and each participant can verify the consistency of his share with this extra information. The scheme makes use of homomorphic encryption property. Verifiable Secret Sharing have found applications in secure multi party computation and e-voting [14] [176].

Stadler [203] proposed a Publicly Verifiable Secret Sharing (PVSS) scheme. In this not only the participant but everybody is able to verify that the shares have been correctly distributed. The scheme can be used with threshold or more general monotone access structure. It is based on ElGamal's cryptosystem [68]. Different proposals are made with applications in e-voting and key-escrow system [31] [74] [186]. An information theoretic secure PVSS is proposed in [208]. The use of Elliptic Curve and Pairing for PVSS is proposed in [220].

Cheating in secret sharing is a major security issue. The participant may give wrong shares during the reconstruction phase and hence all other participants except the cheater will get wrong secret. We need mechanism to determine whether cheating occurred or not. If cheating occurred, the protocol should not proceed further. There are also constructions which can identify, who is the cheater. This adds more complexity compared with cheating detection schemes. *Cheating detection* and *cheater identification* is a major security requirement in secret sharing protocol, which adds more

reliability. One straightforward solution to the problem of cheating is to have the distributor of shares sign each share 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 Woll [213] mentioned cheating in Shamir's scheme and proposed a cheating detection scheme. Several proposals have been made to detect cheaters in threshold secret sharing schemes [37] [44] [135] [159].

Code based secret sharing provides a solution for cheating detection and cheater identification, proposed by McElice and Sarwate [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. Brickell and Stinson [37] proposed a modified version of the Blackley's construction in which honest participants are able to identify cheaters. Number of shares and also the size of the shares is an issue in the modified scheme. The scheme is having the cheating detection and cheater identification capabilities. 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 a single shadow. 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]. A t cheater identifier for (t, n) Shamir 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]. 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 (i.e., 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].

Secret Sharing schemes having the property that, the correct secret can still be recovered even if some of the submitted shares are invalid are called *robust secret sharing* schemes. Additional information is needed to achieve robustness. Schemes which can detect or identify cheaters are not robust. Robust secret sharing schemes are based on error correcting codes. Rogaway and Bellare [179] studied this within a number of different models.

Another type of approach was proposed by Pieprzyk and Zhang [225] by introducing the concept of *cheating immune secret sharing* scheme. In this, submission of corrupted shares will not give any advantage to the cheaters over the honest participants in the recovery of original secret. The advantage of cheating immune scheme is that it does not require extra information on the shares or additional shares during reconstruction. 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 *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 now 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].

Secret sharing schemes assumes long lived shares. This will help attackers gaining knowledge about shares and eventually obtain the information about threshold number of shares and hence able to recover

the secret. There is also chance that the shares may be corrupted or lost due to hardware failure. One way to provide security against perceptual leakage was to periodically refresh the shares in such a way that any information learned by the adversary about individual shares becomes obsolete after the shares are renewed. These schemes are called *proactive secret sharing* schemes. In proactive secret sharing, the shares are modified in such a way that the old shares or the old shares combined with new shares which is less than the given threshold will not give any information about the secret. Proactive security for secret sharing was first suggested by Ostrovski and Yung in [158] in 1991. The basic robust model is proposed by Herzberg [97]. Jarecki [112] come up with two methods of proactive secret sharing using VSS scheme. *Mobile Proactive Secret Sharing* (MPSS) is proposed by Schultz et al [189]. This provides mobility i.e., the shares of the secret hold by a group of nodes can change at each re-sharing, which is necessary in a long-lived system. Bai et al [5] proposed a proactive secret sharing scheme based on matrix projection method.

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. One common drawback of all secret sharing scheme is that they

are 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. So schemes are needed where same share can be used for obtaining multiple secrets.

Secret sharing schemes where several secrets are shared are called *multi secret sharing* schemes. In multi secret sharing scheme, participants only needs to keep a single share. Many secrets are shared independently with out refreshing the shadows. The Dealer uses a public bulletin board for publishing the public information needed for reconstructing the secret. The participants uses pseudo shares, which is computed from the original share and the public information for the reconstruction of multiple secrets. Reconstruction of a secret thus will not reveal any information about the secret share and also remaining secrets that have not been reconstructed. There are *multistage secret sharing* scheme where multiple secrets are revealed stage by stage with each secret revealed in one stage. In *single stage secret sharing* scheme all the shared secrets are revealed in single stage of the protocol. 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. 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. Both the schemes are not perfect. They are also one time threshold schemes. That is, the shares cannot be reused. Once the secret is reconstructed, both the secret and all the shares become known to everyone.

Multi secret sharing schemes are further classified according to the access structure i.e., threshold or generalized multi secret sharing [28] [110]. 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. This needs a public bulletin board, where public values are posted. In 1995 Harn [89] shows an alternative implementation of multi stage secret sharing which requires less public values. There are lot of constructions for the threshold multi secret sharing scheme. In 2000, Chien et al [52] proposed a (t, n) multi-secret sharing scheme based on the systematic block codes. In order to reduce the complexity of the secret reconstruction, Yang et al [221] proposed an alternative scheme based on Shamir's secret sharing in 2004 (YCH scheme). But there are more public values required in Yang's scheme than in Chien's scheme. Motivated by these concerns, a new (t, n) multi-secret sharing scheme is proposed by Pang and Wang [161] in 2004. The scheme is as easy as Yang's scheme in the secret reconstruction and requires the same number of public values as Chie's scheme. Chao-Wen Chan et al [45] proposed a multi-secret sharing scheme, which is based on CRT (Chinese Remainder Theorem) and polynomial. Verifiability in multi secret sharing is applied in [61] [193] [227]. Hash function based multi-secret sharing are proposed recently by Javier Herranz et al [95] and Jun Shao [192] in 2014.

The Elliptic curve cryptography was introduced by Koblitz [121] and Miller [147]. Elliptic curves were found numerous applications in cryptography [147]. Developed as a public key crypto system, it is found more secure with small key size compared with other public key crypto system. Elliptic Curve Discrete Logarithm Problem (ECDLP) is much harder compared with the Discrete Logarithm Problem (DLP). So the computational cost can be reduced while maintaining the same level of security with small key size. In 1993 Meneze's et al [145] introduced pairing. Pairing is introduced to show an attack on elliptic curve discrete logarithm problem and later found useful applications. Pairing on elliptic curve have found useful applications in identity based encryption,

threshold cryptography and signature schemes, multi party key exchange etc [67]. The use of elliptic curve and pairing have found applications in secret sharing schemes very recently. Several schemes based on threshold and generalized access structure is proposed and they have found useful applications.

Pairing can be used to introduce verifiability and cheating detection in secret sharing scheme with more security. Chen Wei et al [218] in 2007 proposed a dynamic threshold secret sharing scheme based on bilinear maps. A threshold multi secret sharing scheme based on elliptic curve discrete logarithm is proposed by Runhua Shi et al [194] in 2007. Sharing multiple secrets which are represented as points on elliptic curve using self pairing[133] is proposed by Liu et al [137] in 2008. In Wang's et al scheme, the number of secrets must be less than or equal to the threshold and also more public values must be changed when the secret need to be updated. Eslami et al [69] in 2010 proposed a modified scheme which avoids these problems. Several publicly verifiable secret sharing schemes are proposed based on pairing, but most of them are single secret sharing schemes [212] [220] [223]. An efficient One Stage Multi Secret Sharing(OSMSS) is proposed recently in 2014 by Fatemi et al [70]. Elliptic curves are also used for the construction of Generalized secret sharing schemes with monotone access structure. Cheating detection is incorporated in this scheme using Bilinear pairing [100] [226].

Simmons [199], Stinson [204] and Beimel [10] had done excellent reviews of secret sharing schemes and their terminologies. In this thesis we provide an explication of threshold secret sharing schemes and generalized secret sharing scheme constructions. Multi secret sharing based on generalized monotone access structure is also explored. We also consider the adversary model and explored the extended capabilities to handle the malicious participants or Dealer. We have described only two important applications of secret sharing schemes here. E-voting based on

Secure Multi-party Computation and Cheque Truncation System (CTS) based on secret image sharing technique. But there are many such areas where secret sharing schemes can be effectively utilized like authenticated group key transfer protocol [92], broadcast encryption [19], visual cryptography, distributed computing etc [14] [21] [48] [55] [63] [85] [153] [152] [191] [209]. Elliptic curve pairing and their applications are reviewed by Dutta et al [67].

1.3 Preliminaries

In this section we give some definitions and notations associated with secret sharing schemes.

In secret sharing, the secret is divided among n participants in such a way that only designated subset of participants can recover the secret. But any subset of participants which is not a designated set cannot recover the secret.

Let $\mathcal{P} = \{P_i | i = 1, 2, \ldots, n\}$ be the set of participants. The secret be K. The set of all secret is represented by $\mathcal{K}$. The set of all shares $S_1, S_2, \ldots, S_n$ is represented by $\mathcal{S}$.

A set of participants who can recover the secret is called an *access structure* or *authorized set* and a set of participants which is not an authorized set is called an *unauthorized set* or *forbidden set*. So the power set of $\mathcal{P}$, $2^{\mathcal{P}}$ can be partitioned into two classes.

1. The class of authorized sets $\mathcal{A}(\Gamma)$ is called the *access structure.*

2. The class of unauthorized sets $\mathcal{A}^c(\Gamma^c) = 2^{\mathcal{P}} \setminus \mathcal{A}$.

We assume that $\mathcal{P}, \mathcal{K}, \mathcal{S}$ are all finite sets and there is a probability distribution on $\mathcal{K}$ and $\mathcal{S}$. We use $H(\mathcal{K})$ and $H(\mathcal{S})$ to denote the entropy of $\mathcal{K}$ and $\mathcal{S}$ respectively.

In a secret sharing scheme there is a special participant called *Dealer* $\mathcal{D} \notin \mathcal{P}$, who is trusted by everyone. In order to set up a secret sharing scheme, the Dealer chooses a secret $K \in \mathcal{K}$ and distribute privately the shares $S_1, S_2, \ldots, S_n$ to the participants.

In secret reconstruction phase, participants of an access set pool their shares together and recover the secret. Alternatively participants could give their shares to a combiner to perform the computation for them. Thus a secret sharing scheme for the access structure $\mathcal{A}$ is the collection of two algorithms:

Distribution Algorithm: This algorithm has to be run in a secure environment by a trustworthy party. The algorithm uses the function

$$f: \qquad \mathcal{K} \times \mathcal{P} \longrightarrow 2^{\mathcal{S}}$$

which for a given secret $K \in \mathcal{K}$ and a participant $P_i \in \mathcal{P}$, assigns a set of shares from the set $\mathcal{S}$ that is $f(K, P_i) = S_i \subseteq \mathcal{S}$ for $i = 1, \ldots, n$.

Recovery Algorithm: This algorithm has to be executed collectively by cooperating participants. We can consider the combiner as a process embedded in a tamper proof module and all participants have access to it. Also the combiner outputs the result via secure channels to cooperating participants. The combiner applies the function

$$g : \mathcal{S}^t \longrightarrow \mathcal{K}$$

to calculate the secret. For any authorized set of participants $g(S_1, \ldots, S_t) = K$, if $P_1, \ldots, P_t \subseteq \mathcal{A}$. If the group of participant belongs to an unauthorized set, the combiner fails to compute the secret.

A secret sharing scheme is called perfect, if for all sets B, $B \subset \mathcal{P}$ and $B \notin \mathcal{A}$, if participants in B pool their shares together they cannot reduce their uncertainty about S. That is, $H(K) = H(K|\mathcal{S}_B)$, where $\mathcal{S}_B$ denote

the collection of shares of the participants in B. It is known that for a perfect secret sharing scheme $H(S_i) \geqslant H(K)$.

An access structure $\mathcal{A}_1$ is *minimal* if $\mathcal{A}_2 \subset \mathcal{A}_1$ and $\mathcal{A}_2 \in \mathcal{A}$ implies that $\mathcal{A}_2 = \mathcal{A}_1$. Only *monotone access structure* is considered for the construction of the scheme in which $\mathcal{A}_1 \in \mathcal{A}$ and $\mathcal{A}_1 \subset \mathcal{A}_2$ implies $\mathcal{A}_2 \in \mathcal{A}$. The collection of minimal access sets uniquely determines the access structure. The access structure $\mathcal{A}$ in terms of minimal access structure is represented by $\mathcal{A}_{min}$.

For an access structure $\mathcal{A}$, the family of unauthorized sets $\mathcal{A}^c = 2^{\mathcal{P}} \setminus \mathcal{A}$ has the property that given an unauthorized set $B \in \mathcal{A}^c$ then any subset $C \subset B$ is also an unauthorized set. An immediate consequence of this property is that for any access structure $\mathcal{A}$, the set of unauthorized sets can be uniquely determined by its *maximal set*. We use $\mathcal{A}^c_{max}$ to denote the representation of $\mathcal{A}^c$ in terms of maximal set.

For all $B \in \mathcal{A}$, if $|B| \geqslant t$, then the access structure corresponds to a (t, n) threshold scheme.

Information Rate

The size of the share is very important in secret sharing scheme. In Shamir's scheme the share size is same as the secret size. However in generalized scheme the share size is larger than the secret size. The practical relevance of this issue is that the security of any system tend to degrade as the amount of information that must be kept secret. Secondly if the shares given to the participants are too long then the memory requirement will be more and also the share distribution algorithm become in efficient. Therefore it is important to derive significant upper and lower bound on the information rate of secret sharing scheme. Several authors have mentioned about the information rate of a secret sharing scheme as a parameter for efficiency. Karnin, Greene and Hellman [117] have introduced the notion of entropy

in secret sharing. They stated that to recover the secret from t shares

$$H(K|S_1, S_2, \ldots, S_t) = 0$$

and the condition for perfect secrecy is that $t-1$ shares provides absolutely no information about the secret K is achieved by

$$H(K|S_1, S_2, \ldots, S_{t-1}) = H(K)$$

He also proved that perfect threshold secret sharing should satisfy the condition that

$$H(S_i) \geqslant H(K), i = 1, 2, \ldots, n$$

Their approach was limited to threshold schemes. Capocelli et al [43] extended the scheme for generalized scheme and obtained some bounds on the size of shares.

Brickell [34] defined the information rate as $\rho = \frac{log|\mathcal{S}|}{log|\mathcal{K}|}$, where $\mathcal{K}$ is the set of possible secrets and $\mathcal{S}$ is the set of possible shares. He called the secret sharing scheme as ideal, if it is perfect and has information rate $\rho = 1$. Simmons also defined a related notion in [198]. Stinson came up with a good definition [204]. The generalized secret sharing scheme with distribution rule is used as a model to measure the efficiency of the secret sharing scheme by information rate.

Let

$$S_i = \{f(P_i) : f \in (\mathcal{F})\}, 1 \leqslant i \leqslant n$$

S_i is the set of possible shares that P_i might receive. The secret key K comes from a finite set $\mathcal{K}$. If we use binary encoding then K can be represented as a bit string of length $log_2 \mid \mathcal{K} \mid$. In a similar way, a share given to a participant P_i can be represented by a bit string of length $log_2 \mid S_i \mid$. Intuitively P_i receives $log_2 \mid S_i \mid$ bits of information in his share. But the

information content of the secret is $log_2 \mid \mathcal{K} \mid$ bits. The information rate for P_i is the ratio

$$\rho_i = \frac{log_2 \mid \mathcal{K} \mid}{log_2 \mid S_i \mid}$$

The information rate of the scheme is defined as

$$\rho = min\{\rho_i : 1 \leqslant i \leqslant n\}$$

High information rate is the desirable property. It is noted that for perfect secret sharing scheme $\rho \leqslant 1$. $\rho = 1$ is the optimal situation and such schemes are referred as *ideal schemes*.

Definition 1.3.1. Information rate for a particular user is the bit size ratio (size of the shared secret)/(size of the user's share). The information rate for a secret sharing scheme itself is the minimum such rate over all users.

Remark 1.3.1. In any perfect secret sharing scheme (size of the share) $\geqslant$ (size of the secret). Consequently, all perfect secret sharing scheme must have information rate $\leqslant 1$.

In the rest of this chapter, we will briefly describe the early constructions of simple threshold secret sharing schemes.

1.4 Unanimous Consent Control Scheme

In this scheme the secret $K \in \mathbb{Z}_m$ is divided among n users ($m \geqslant n+1$), all of whom are required in order to recover the secret K. Karnin, Greene and Hellman [117] have proposed this very simple unanimous consent scheme.

1. The Dealer generates $n-1$ independent random numbers $S_1, \ldots, S_{n-1}$ as shares from $\mathbb{Z}_m$, where $0 \leqslant S_i \leqslant m-1, 1 \leqslant i \leqslant n-1$.

2. The participants P_1 through P_{n-1} are given shares S_i, while P_n is given $S_n = K - \sum_{i=1}^{n-1} S_i \pmod{m}$.

3. The secret K is recovered by combining all the shares as

$$K = \sum_{i=1}^{n} S_i \pmod{m}$$

.

The scheme can be easily generalized to any group. If $K = \{0,1\}^l$, bit-wise XOR defines the group operation. Both the secret and shares are of l bits in this case. This scheme is perfect. It is important to design more efficient unanimous consent schemes because they are the building blocks of the generalized secret sharing schemes. Sreekumar et al [202] proposed an efficient (n, n) scheme based on a number system called Permutation Ordered Binary (POB) number system.

Remark 1.4.1. The individual key component in a split control scheme should be full-length. This provides greater security than partitioning a b bit secret K into n pieces of $\frac{b}{n}$ bits each.

1.5 Threshold Secret Sharing Schemes

The problem with secret splitting is that all n participants must collaborate to recover the secret. If one share is lost then it is impossible to recover the secret. There are application scenarios where controlled access is necessary. For example in a bank three tellers are employed to open a vault. Combination of any two or all three can open the vault but a single person is not allowed to do so. These problems can be solved by means of threshold secret sharing schemes.

Definition 1.5.1. A (t, n) threshold scheme $(1 < t \leqslant n)$ is a method by which a trusted party called Dealer compute secret shares $S_i, 1 \leqslant i \leqslant n$ from a secret K and securely distribute among a finite set $\mathcal{P}$ of n participants in such a way that any t participants or more can compute the value of K. But no group of less than t participants can do so.

The example mentioned above is a $(2, 3)$ threshold secret sharing scheme. The unanimous consent control is a (n, n) threshold scheme. These threshold schemes help to achieve both availability and confidentiality.
Availability: greater than or equal to t parties can recover the secret K.
Confidentiality: less than t parties have no information about the secret K.

Definition 1.5.2. A (t, n) threshold secret sharing scheme is perfect, if $t - 1$ or fewer shares give no information about the secret.

Shamir [190] and Blakley [24] proposed their threshold secret sharing primitives independently in 1979. Shamir's scheme is based on Lagrange interpolating polynomials where as the latter is based on vector subspaces and projective geometry. Karnin et al [117] approach can be viewed as a deterministic version of Blakley's scheme and includes Shamir's method as a special case. The Shamir's scheme is perfect where as Blakley's scheme is not so efficient and also not perfect. Mignotte [146] developed a threshold scheme based on a special sequence of numbers called Mingotte sequence and Chinese Remainder Theorem. The scheme is not perfect. Asmuth and Bloom [2] also developed a schemes based on Mingotte's scheme. It is a perfect threshold scheme and also having less computational complexity than Shamir's scheme.

1.5.1 Shamir's Threshold Secret Sharing Scheme

Shamir [190] has proposed a scheme based on Lagrange interpolating polynomials. For a (t, n) threshold scheme, $\mathcal{D}$ pick a random $t-1$ degree polynomial $q(x) = a_0 + a_1 x + \ldots + a_{t-1}x^{t-1}$ in which a_0 is the secret $K \in \mathbb{F}_p$ and $a_i \in \mathbb{F}_p$, where $p \geqslant n+1$ is a prime. Dealer then generate n shares $S_1 = q(1), \ldots, S_i = q(i), \ldots, S_n = q(n)$ and securely distribute them to the participants. Given any subset t of these S_i values (together with their identifying indices), we can find the coefficients of $q(x)$ by interpolation and then evaluate $K = q(0)$. Knowledge of just $t-1$ of these values on the other hand does not suffice in order to calculate K.
Let

$$q(x) = a_0 + a_1 x + \ldots + a_{t-1}x^{t-1} \quad , where \quad a_0 = K.$$

The n shadows are computed by evaluating $q(x)$ at n different values $x_1, \ldots, x_n$, and $x_i \neq 0$ for any i

$$S_i = q(x_i) \qquad , 1 \leqslant i \leqslant n$$

Each point (x_i, S_i) is a point on the curve defined by the polynomial. The values $x_1, \ldots, x_n$ need not be secret and could be user identifiers or simply the numbers through $1, \ldots, n$. Because t points uniquely determine the polynomial $q(x)$ of degree $t-1$, the secret K can be constructed from t shadows. If P is the set of participants and $A \subseteq P$ such that $|A| \geqslant t$ then $q(x)$ can be constructed using the Lagrange interpolation formula with t shares of the participants:

$$q(x) = \sum_{j=1}^{t} \left(S_{ij} \cdot \prod_{1 \leqslant k \leqslant t, k \neq j} \frac{x - x_{ik}}{x_{ij} - x_{ik}} \right)$$

Since $K = q(0)$ there is no need for generating the polynomial. So we can

rewrite the formula as

$$K = q(0) = \sum_{j=1}^{t} \left(S_{ij} \cdot \prod_{1 \leqslant k \leqslant t, k \neq j} \frac{x_{ik}}{x_{ik} - x_{ij}} \right)$$

if

$$b_j = \prod_{1 \leqslant k \leqslant t, k \neq j} \frac{x_{ik}}{x_{ik} - x_{ij}}$$

Then

$$K = \sum_{j=1}^{t} (S_{ij} \cdot b_j)$$

Hence the secret can be computed as a linear combination of t shares. If we choose $x_i's$ as $1, \ldots, n$ then the computation become very simple.

A linear algebraic way of interpreting this is, given t points of $q(x)$, the following system of equations can be generated.

$$\begin{aligned} K + a_1 x_1 + a_2 x_1^2 + \ldots + a_{t-1} x_1^{t-1} &= S_1 \\ K + a_1 x_2 + a_2 x_2^2 + \ldots + a_{t-1} x_2^{t-1} &= S_2 \\ \vdots \quad \vdots & \\ K + a_1 x_t + a_2 x_t^2 + \ldots + a_{t-1} x_t^{t-1} &= S_t \end{aligned}$$

The above system has t linear equations and t unknowns $K, a_1, a_2, \ldots, a_{t-1}$. We can rewrite the equations as

$$A. \begin{pmatrix} K \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_t \end{pmatrix}$$

where the coefficient matrix A is a Vandermonde matrix

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{t-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{t-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_t & x_t^2 & \cdots & x_t^{t-1} \end{pmatrix}$$

It is noted that the matrix A is square having rank t so the determinant is non zero and is

$$det(A) = \prod_{1 \leqslant i,j \leqslant t} (x_j - x_i)$$

The system of equations has a unique solution for $K, a_1, \ldots, a_{t-1}$ and hence we can recover K.

It is noted that less than t participant cannot get any information about the secret, because they cannot rule out any of the possibilities for the secret in $\mathbb{F}_p$.

Example 1.5.1. Let us consider a $(3, 5)$ threshold scheme. $q(x) = 2x^2 + 3x + 5$ over the field $\mathbb{Z}_{11}$. The secret $K = 5$. The shares are $q(1) = S_1 = 10, q(2) = S_2 = 8, q(3) = S_3 = 10, q(4) = S_4 = 5, q(5) = S_5 = 4$. Participants P_1, P_2, P_4 can pool their shares and retrieve the secret using the Lagrange Interpolation as:

$$10 * \frac{2}{2-1} * \frac{4}{4-1} + 8 * \frac{1}{1-2} * \frac{4}{4-2} + 5 * \frac{1}{1-4} * \frac{2}{2-4} = 5 \pmod{11}$$

The arithmetic used is modular. The set of integers modulo a prime number p forms a field in which interpolation is possible. The prime p is chosen which is bigger than both K and n. The coefficients $a_1, \ldots, a_{t-1}$ in $q(x)$ are randomly chosen from a uniform distribution over the integers $0, \ldots, p-1$ and the shares $S_1, \ldots, S_n$ are computed modulo p. If the number K is large, it is advisable to break it into shorter block of bits for easy arithmetic. Efficient $O(nlog^2n)$ algorithms for polynomial evaluation and

interpolation are discussed in [1] and [66].
Some of the useful properties of Shamir's (t, n) threshold scheme are:

- The size of the share S_i does not exceed the size of the secret K.
- When t is fixed, shares can be dynamically added.
- It is easy to change the shares without changing the original secret K-all we need is a new polynomial $q(x)$ with the same free term.
- We can get a hierarchical scheme, where the number of shares given to each user is proportional to the user's importance.
- The scheme is perfect and ideal.

1.5.2 Blakley's Threshold Scheme

The Blakley's scheme is based on geometry and has gained much attention for the development of secret sharing schemes. In Balkley's scheme [24], the secret is an element of the vector space $\mathbb{F}_q^t$. The shares are any n distinct $(t-1)$ dimensional hyperplanes that contain the secret. The t dimensional hyperplane is a set of the form

$$\{(x_1, \ldots, x_t) \in \mathbb{F}_q^t | \alpha_1 x_1 + \cdots + \alpha_t x_t = \beta\}$$

where $\alpha_1, \ldots, \alpha_t$ and β are arbitrary elements of the field $\mathbb{F}_q$.

To realize a (t, n) threshold scheme, the secret is represented as a point P in the projective t dimensional plane $\mathbb{F}_q^t$. There are $(q^t - 1)/(q - 1)$ hyperplanes that contain P. The Dealer randomly select a hyperplane and distribute it as a share to the participant. It is noted that t hyperplane will intersect at P and fewer than t hyperplanes will intersect only in some subspace containing P. Thus fewer than t participants are able to recover the subspace, but cannot figure out the secret correctly.

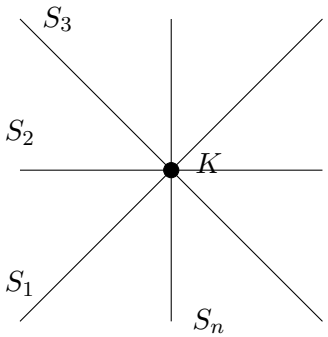

Figure 1.1: Blackley's scheme for threshold t=2

The scheme is not perfect because the coalition of more participants will get partial information of the subspace containing the secret and they have a better chance of guessing the secret. The scheme is improved by Simmons [199] to make it perfect using an affine space instead of projective spaces.

The secret can be obtained by intersecting any t shares. A $(2, n)$ scheme is shown in the Figure 1.1. It is not so efficient compared with Shamir's scheme. The scheme is also not perfect because participants get partial information about the hyperplane where the secret lies.

1.5.3 Karnin-Greene-Hellman Scheme(KGH)

KGH [117] scheme is based on vectors. $n + 1$ column vectors $A_0, A_1, A_2, \ldots, A_n$ of size t are chosen such that any t of them have full rank. If B is a row vector of size t, then the secret $K = BA_0$. The shares are generated as $S_i = BA_i, 1 \leqslant i \leqslant n$. If any t of the n shadows are known then B can be determined and the secret K is obtained evaluating BA_0. If less than t shadows are known then B cannot be determined and hence the secret cannot be revealed.

Karnin et al showed that Shamir's and Blakley's schemes are special cases of their threshold scheme. The scheme can also be extended to protect more than one secret. They addressed the problem of deliberate tampering of the shares by trustees which can be identified by using one way function [64].

1.5.4 Brickell's Scheme

Brickell [34] also give a generalized notion of Shamir and Blackleys schemes. The basic secret sharing scheme mentioned is as follows.

The secret is an element in some finite field $\mathbb{GF}(q)$. The Dealer chooses a vector $a = (a_0, \ldots, a_t)$ for some t, where each $a_j \in \mathbb{GF}(q)$ and a_0 is the secret. Denote the participants by P_i, for $1 \leqslant i \leqslant n$. For each P_i, the Dealer will pick a $t-$dimensional vector v_i over $\mathbb{GF}(q)$. All of the vectors v_i, for $1 \leqslant i \leqslant n$ will be made public. The share that the Dealer gives to P_i will be $S_i = v_i \cdot a$. Let e_i denote the i^{th} $t-$ dimensional unit coordinate vector (i.e., $e_1 = (1, 0, \ldots, 0)$).

Proposition 1.1. Let $\mathcal{P} = \{P_{i1}, P_{i2} \ldots P_{ik}\}$ be a set of participants

- The participants in $\mathcal{P}$ can determine the secret if the subspace $< v_{i1}, \ldots, v_{ik} >$ contains e_1.
- The participants in $\mathcal{P}$ receive no information about the secret if the subspace $< v_{i1}, \ldots, v_{ik} >$ does not contain e_1.

Theorem 1.5.1. *Linear combination of k vectors can retrieve the secret, if it spans e_1.*

Proof. Let M be the matrix with rows $v_{i1}, \ldots, v_{ik}$ and $s = (s_{i1}, \ldots, s_{ik})$. Let w be the vector such that $w \cdot M = e_1$. Then $w \cdot M \cdot a = a_0$. Hence $w \cdot s = a_0$. □

They also shown similar techniques for multilevel and compartmental threshold schemes.

1.5.5 Genaralized Linear Threshold Scheme

Kothari [123] gave a generalized threshold scheme. A secret is represented by a scalar and a linear variety is chosen to conceal the secret. A linear functional known to all trustees is chosen and is fixed in the beginning, which is used to reveal the secret from the linear variety. The n shadows are hyperplanes containing the liner variety. Moreover the hyperplanes are chosen to satisfy the condition that the intersection of less than t of them results in a linear variety which projects uniformly over the scalar field by the linear functional used for revealing the secret. The number t is called the threshold. Thus as more shadows are known, more information is revealed about the linear variety used to keep the secret. However no information is revealed until the threshold number of shadows are known.

He had shown that Blakley's scheme and Karin's scheme are equivalent and provided algorithms to convert one scheme to another. He also stated that the schemes are all specialization of generalized linear threshold scheme. The generalized linear threshold scheme allows linear variety of positive dimension to conceal the secret. This fact is utilized in constructing a hierarchical threshold scheme. The hierarchical threshold scheme uses a chain of linear varieties to keep a secret and allows multiple thresholds for hierarchy of trustees.

Remark 1.5.1. The schemes mentioned in sections 1.5.1 to 1.5.5 are called **Linear Threshold** schemes because they employee common principles from linear algebra and the secret is represented as a linear combination of shares.

1.5.6 Mingotte's Scheme

The Mingotee scheme [146] is based on modulo arithmetic and **Chinese Remainder Theorem (CRT)** [65]. A special sequence of integers called Mingotte sequence is used here.

Definition 1.5.3. Let n be an integer $n \geqslant 2$, and $2 \leqslant k \leqslant n$.A (k, n) Mingotte sequence is a sequence of pairwise coprime positive integers $p_1 < p_2 < \cdots < p_n$ such that $\prod_{i=0}^{k-2} p_{n-i} < \prod_{i=1}^{k} p_i$

Given a Mingotte sequence the (k, n) scheme works as follows

- The secret K is chosen as a random integer such that $\beta < K < \alpha$, where $\alpha = \prod_{i=1}^{k} p_i$ and $\beta = \prod_{i=0}^{k-2} p_{n-i}$.
- The n shares are generated as $S_i = K \pmod{p_i}$,$(1 \leqslant i \leqslant n)$.
- Given k distinct shares $S_1, S_2, \ldots, S_k$, the secret K is recovered using the CRT, as a unique solution modulo $p_1 \ldots p_k$ of the system of congruences.

$$\begin{cases} x & \equiv S_{i1} \pmod{p_{i1}} \\ & \vdots \\ x & \equiv S_{ik} \pmod{p_{ik}} \end{cases}$$

Mingotte's scheme is not perfect, but it can lead to small shares. This scheme is extended by Iftene [105] by introducing the generalized Mingotte sequences whose elements are not necessarily pairwise coprime.

1.5.7 Asmuth-Bloom Scheme

This scheme is proposed by Asmuth and Bloom [2]. It also uses a special sequence of pairwise coprime positive integers $p_0 < p_1 < \cdots < p_n$ such that

$$p_0 \cdot \prod_{i=0}^{k-2} p_{n-i} < \prod_{i=1}^{k} p_i$$

Given a publicly known Asumuth-Bloom sequence, the scheme works as follows:

- The secret K is chosen as a random integer from $\mathbb{Z}_{p_0}$.
- The n shares are generated as $S_i = K + r \cdot p_0 \pmod{p_i}$, for all $(1 \leqslant i \leqslant n)$ where r is an arbitrary integer such that $K + r \cdot p_0 \in \mathbb{Z}_{p_1 \cdots p_k}$.
- Given k distinct shares $S_1, S_2, \ldots, S_k$, the secret K is recovered using $S_0 \pmod{p_0}$, where S_0 is the unique solution of the system of congruences using CRT.

$$\begin{cases} x & \equiv s_{i1} \pmod{p_{i1}} \\ & \vdots \\ x & \equiv s_{ik} \pmod{p_{ik}} \end{cases}$$

The scheme is not perfect. The probabilities of the shares of $k-1$ participants with respect to two different keys are not the same, but asymptotically equal. A larger value of p_0 will eventually leads to smaller difference between these two probabilities. This difference approaches zero when p_0 grows to infinity. Goldreich, Ron and Sudan [84] have proposed choosing $p_0, p_1, \ldots, p_n$ as prime numbers of the same size. Quisquater Preneel and Vandewalle [173] have proven that by choosing $p_0, p_1, \ldots, p_n$ as consecutive primes asymptotically perfect and ideal schemes can be obtained.

1.6 Extended Threshold Schemes

1.6.1 Weighted Threshold Secret Sharing Scheme

Consider the situation in a company where three executives can recover a secret or by an executive and a vice president, or by the president alone.

Shamir's [190] solution is by using a $(3, n)$ threshold scheme. The idea is to give more shares to more important persons. Thus the president receive three shares, each vice president receives two shares and executive receive only one share. Any three shares can be combined to retrieve the secret.

Weighted Threshold schemes are generalization of this scenario where each user is assigned a positive weight and the secret can be reconstructed if and only if the sum of the weights of the participants is greater than or equal to a fixed threshold.

Definition 1.6.1. Let $n \geqslant 2$, $w = (w_1, \ldots, w_n)$ be a sequence of positive integers, and t is a positive integer such that $2 \leqslant t \leqslant \sum_{i=1}^{n} w_i$, where w_i are the weights and t is the threshold of the scheme. The secret sharing scheme having the access structure

$$\mathcal{A} = \{A \in P(\{1, 2, \ldots, n\}) | \sum_{i \in A} w_i \geqslant t\}$$

is referred to us the (w, t, n) **weighted threshold secret sharing** scheme.

Shamir's idea was to use a (t, n) threshold scheme where each participant is assigned weight 1. Information rate of threshold schemes corresponds to specific access structure is given by Morillo et al [149]. Beimel et al [12] characterize the weighted threshold access structure that are ideal. Monotone circuit for monotone weighted threshold is also mentioned in [13]. Chinese Remainder Theorem is used by Iftene [106] for the construction of weighted threshold scheme.

Remark 1.6.1. (t, n) Threshold secret sharing scheme is nothing else than (w, t, n) weighted threshold secret sharing scheme with $w_1 = w_2 = \cdots = w_n = 1$.

Benaloh and Leichter proved [15] that there are monotone access structures that are not weighted threshold.

1.6.2 Hierarchical Secret Sharing Schemes

In case of hierarchical (or multilevel) secret sharing, the set of users are partitioned into some levels $l_1, l_2, \ldots, l_m$ depending on their hierarchy with l_1 at the highest level and l_m at the lowest level. A level threshold t_j is assigned to the j^{th} level, for all $1 \leqslant j \leqslant m$. We can naturally assume that $t_1 \leqslant t_2 \leqslant \cdots \leqslant t_m$. There is a level called initialization level. The secret can be recovered if and only if the number of participants from this level or higher levels is greater than or equal to the initialization level threshold.

Definition 1.6.2. Let $L = \{l_1, l_2, \ldots, l_m\}$ be a partition of $\{1, 2, \ldots, n\}$ and $T = (t_1, t_2, \ldots, t_n)$ is the sequence of level thresholds, where $1 \leqslant t_j \leqslant \mid l_j \mid$, for all $1 \leqslant j \leqslant m$ and $t_1 < t_2 < \cdots < t_m$. Then the (l, t) multilevel access structure is given by

$$\mathcal{A} = \{A \in P(\{1, 2, \ldots, n\}) \mid (\exists j \in \{1, \ldots, m\})(\mid A \cap \cup_{i=1}^{j} l_i \mid \geqslant t_j)\}$$

Multilevel level secret sharing scheme has been considered for the first time by Simmons [199] and then by Brickell [34]. Brickell proved that there exist ideal schemes for the multilevel access structure. Ghodosi et al [79] proposed an ideal scheme based on the extension of threshold scheme.

1.6.3 Compartmented Schemes

In this scheme the users are partitioned into compartments $c_1, c_2, \ldots, c_m$. Besides a global threshold t, each compartment is assigned a threshold t_j. The secret can be recovered if and only if the number of participants from any compartment is greater than or equal to the corresponding compartment threshold t_j and the total number of participants is greater than or equal to the global threshold t.

Definition 1.6.3. Let $C = \{c_1, c_2, \ldots, c_m\}$ be a partition and $T = (t_1, t_2, \ldots, t_m)$ is the compartmental threshold, where $1 \leqslant t_j \leqslant |c_j|$ for all $1 \leqslant j \leqslant m$ and a global threshold t such that, $\sum_{j=1}^{m} t_j \leqslant t \leqslant n$. The (C, T, t) compartment access structure is

$$\mathcal{A} = \{A \in P(\{1, 2, \ldots, n\}) \mid (|A| > t) \wedge (\forall j \in \{1, \ldots, m\})(\mid A \cap C_j \mid \geqslant t_j)\}$$

Simmons [198] proposed compartmented scheme using geometry techniques. Brickell [36] and Ghodosi et al [79] suggested ideal schemes for the compartmented access structure. Tassa and Dyn [210] introduced a new class of access structure called compartmented access structures with lower bounds. Iftene [105] suggested a compartmented scheme based on Chinese Remainder Theorem.

1.7 Error Correcting Codes and Secret Sharing

McEliece and Sarwate [144] made an observation that Shamir's scheme is closely related to Reed-Solomon codes [141]. The error correcting capability of this code can be translated into desirable secret sharing properties. Reed and Solomon [177] introduced a code having the following property. An m bit message is coded as n bits and is transmitted. If one transmits n bits, the additional $n-m$ bits are redundant and allow one to recover the original message in the event that noise corrupts the signal during transmission and causes some bits of the code to be in error. A multiple-error correcting code of order s consists of a code which maps m-tuples of zeros and ones into n-tuples of zeros and ones, where m and n both depend on s. A decoding procedure which recovers the message completely, assuming no more than s errors occur during transmission in the vector of n bits.

Let $(\alpha_1, \alpha_2, \ldots, \alpha_{r-1})$ be a fixed list of the non zero elements in a finite field F with r elements. In one form of Reed-Solomon coding an information word $a = (a_0, a_1, \ldots, a_{k-1}), a_i \in F$ is encoded into code word $D = (D_1, D_2, \ldots, D_{r-1})$, where $D_i = \sum_{j=0}^{k-1} a_j \alpha_i^j$. The secret is $a_0 = -\sum_{i=1}^{r-1} D_i$, while the pieces of the secret are D_i's.

If given h shares but t of these are in error. Then by applying errors and erasures decoding algorithm, it is possible to recover D and a provided that $h - 2t \geqslant k$. This shows that if t pieces have been tampered, the secret can still be accessed by legitimate users provided that at least $k + t$ valid pieces are available. In the case of a (k, n) threshold scheme, the opponent must tamper $\lfloor (n-k)/2 \rfloor$ pieces to ensure that the secret is inaccessible.

Karnin et a [117] realize threshold schemes using linear codes. Massey [143] introduced the concept of minimal code words and provided that the access structure of a secret sharing scheme based on a $[n, k]$ linear code is determined by the minimal codewords of the dual code.

The approach to construct a general scheme based on linear code is as follows. Choose an $[n+1, t, d]$ code C. Let $K \in \mathbb{F}_q$ denote the secret and $G = (g_0, g_1, \ldots, g_n)$ be the generator matrix of code C. If $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be the set of participants, then Linear Secret Sharing can be constructed using the Error Correcting Code as follows.

- G is known publicly to every one.
- To share a secret K, the dealer randomly select a vector

  $$v = (v_0, v_1, \ldots, v_{t-1})$$

  such that $K = v.g_0$.
- Each participant P_i receives a share $S_i = v.g_i$, for $i = 1, \ldots, n$.

It is noted that the shares of the participants $P_{i1}, P_{i2}, \ldots, P_{il}$ can reconstruct the secret K, if g_0 can be represented as a linear combination of $g_{i1}, g_{i2}, \ldots, g_{il}$.i.e.,$g_0 = a_1 g_{i1} + \ldots + a_l g_{il}$. We have

$$K = v.g_0 = v.\sum_{j=1}^{l} a_j g_{ij} = \sum_{j=1}^{l} a_j \; v.g_{ij} = \sum_{j=1}^{l} a_j s_{ij}$$

Let

$$\Gamma = \{A : A \subset \mathcal{P}, g_0 \text{ is the linear combination of } g_i \text{ corresponds to } P_i \in A\}$$

Then it is noted that Γ is a monotone increasing collection of subset of $\mathcal{P}$.

Remark 1.7.1. For any q-ary $[n+1, t, d]$ code C, a LSS (Linear Secret Sharing Scheme) realizing a monotone access structure can be constructed. An $[n+1, t, d]$-MDS (Maximum Distance Separable) code can be used to construct a (t, n) threshold scheme. Reed-Solomon code is an example which can be used to construct a (t, n) threshold scheme similar to Shamir's scheme.

1.8 Quasi-Perfect Secret Sharing Scheme

In a perfect secret sharing scheme, the size of the share is at least the size of the of secret i.e. $S_i \geqslant K$. Quasi-Perfect secret sharing schemes are one in which the size of the shares can be smaller than that of the secret. These schemes are called Ramp schemes, introduced by Blakley and Meadows [25].

Let $\mathcal{A} \subseteq 2^{\mathcal{P}}$ be an access structure, which is *monotone decreasing*, if for any $A \in \mathcal{A}$ and $B \subseteq A$, then $B \in \mathcal{A}$.

Definition 1.8.1. Let $\Gamma, \mathcal{A} \subseteq 2^{\mathcal{P}}$ be monotone increasing and monotone decreasing access structure respectively such that $\Gamma \cap \mathcal{A} = \emptyset$. A quasi-perfect secret sharing scheme with access structure Γ and adversary structure $\mathcal{A}$ having the following properties

(i) $H(K|S_A) = H(K)$ for any $A \in \mathcal{A}$.

(ii) $H(K|S_I) = 0$ for any $I \in \Gamma$.

(iii) $0 \leqslant H(K|S_B) \leqslant H(K)$ for any $B \in 2^{\mathcal{P}} \setminus (\mathcal{A} \cup \Gamma)$.

The scheme is *non-perfect*, if there exist $B \in 2^{\mathcal{P}} \setminus (\mathcal{A} \cup \Gamma)$ such that

$$0 < H(K|S_B) < H(K)$$

It is noted that any perfect secret sharing scheme realizing the access structure Γ is quasi perfect with a monotonically decreasing access structure $\mathcal{A}$. The *ramp scheme* is an example of this.

Definition 1.8.2. A (d, t, n) ramp scheme is a quasi-perfect secret sharing scheme realizing a monotone increasing access structure Γ and a monotone decreasing access structure $\mathcal{A}$ such that

$$\mathcal{A} = \{A \subseteq \mathcal{P} : |A| \leqslant d\}$$
$$\Gamma = \{I \subseteq \mathcal{P} : |I| \geqslant t\}$$

We represent $\mathcal{A}$ and Γ as $\mathcal{A}_{d,n}$, $\Gamma_{t,n}$ respectively.

1.9 Thesis contribution

Security is a major challenge in the digital storage and transmission of data. Secret sharing protocols provides solutions to several security problems including secure key management, distributed access control,

secure distributed storage and transmission and secure multi party computation. The major contribution of this dissertation is in the development of secret sharing protocol and also exploring the use of its in typical application areas. This section is devoted to mention various contributions made by us in the area of secret sharing.

- We started with the development of simple schemes which are application oriented. There are several application areas where $(2, 3)$ or $(2, 4)$ threshold secret sharing schemes are widely used. Simple and efficient secret sharing schemes using number theory and XOR operations are developed for sharing data and images. Development of the algorithms and their analysis is one major development in the area of study. XOR based schemes are simple and easy to use compared with Shamir's scheme. The use of these scheme in the area of distributed data storage and secret image sharing are also explored.

- A specially designed number system called POB (Permutation Ordered Binary) system developed by Sreekumar et al [201] is studied. They have suggested only the threshold secret sharing schemes using POB and we extended its use to build secret sharing scheme realizing more general access structures. Cumulative arrays are used along with POB to build the scheme. The scheme is not ideal but it is memory efficient when the storage become a constraint.

- Major contribution of the dissertation is in the development of secret sharing schemes with extended capabilities. We have considered multi secret sharing scheme as an extension with added capabilities like verifiability, cheating detection, cheater identification, dynamism etc. A detailed study of the existing mutli secret sharing scheme is done.

Analyzed their drawbacks and complexities. We then developed a multi secret sharing with general access structure. Cheating detection and cheater identification is also incorporated to make it more robust.

- The use elliptic curve in secret sharing is a growing research area. We investigated the use of Elliptic curve and pairing in designing secure and reliable secret sharing schemes. We have developed two schemes in this direction. A threshold multi secret scheme using elliptic curve and self pairing and also a multi secret sharing scheme with general access structure using bilinear pairing. These scheme are less complex and easy to implement compared with the existing proposals for multi secret sharing scheme using elliptic curve. The proposed schemes also having several additional capabilities and also having less public parameters.

- We have considered two application areas of secret sharing in this thesis. Multi party computation with application to E-voting and secret image sharing with application to Cheque Truncation System (CTS). The additive homomorphic property of Shamir's scheme along with encoding and decoding of votes is the key component. The vote tallying along with the votes gained by each contesting candidate can be obtained. Cheque Truncation System (CTS) or Image-based Clearing System (ICS), in India is a project undertaken by the Reserve Bank of India (RBI) in 2008 for faster clearing of cheques. We propose a scheme based on secret image sharing as a replacement of the existing scheme by RBI. The proposed schemes avoids the complicated encryption decryption process and key management in the existing scheme.

1.9.1 List of Publications

As part of the research work various papers were presented and published in peer reviewed International Journals as well as in Conference proceedings. They are listed below:

1. Binu V. P., A. Sreekumar.,"Lossless Secret Image Sharing Scheme", International Journal of Computational Intelligence and Information Security. Vol-4,No-4, P-42-48, April 2013, ISSN: 1837-7823.

2. Binu V. P., A. Sreekumar., "An Epitome of Multi Secret Sharing Schemes for General Access Structure",International Journal of Information Processing, 8(2), 13-28, 2014, ISSN : 0973-8215.

3. Binu V. P., A. Sreekumar.,"Efficient Multi Secret Sharing with Generalized Access Structures", International Journal of Computer Applications 07/2014; 90(12). DOI:10.5120/15769-4446.

4. Binu V. P., A. Sreekumar., "Simple and Efficient Secret Sharing Schemes for Sharing Data and Image" International Journal of Computer Science and Information Technologies, Vol. 6 (1), 2015, 404-409.ISSN:0975-9646.

5. Sreela S. R., G. Santhosh Kumar, Binu V. P., "Secret Image Sharing Based Cheque Truncation System with Cheating Detection." International Journal of Information Processing, 8(4), 56-67, 2014, ISSN : 0973-8215.

6. Binu V. P., Divya G Nair, Sreekumar A.,"Secret Sharing Homomorphism and Secure E-voting", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 22 (2015) pp 42934-42941.

7. Binu V. P., Sreekumar A.,"Threshold Multi Secret Sharing Using Elliptic Curve and Pairing", International Journal of Information Processing, 9(4), 100-112, 2015, ISSN : 0973-8215.

8. Binu V. P., Sreekumar A.,"Secure and Efficient Secret Sharing Scheme with General Access Structures based on Elliptic Curve and Pairing", Wireless Personal Communications-Springer, ISSN: 0929-6212. DOI 10.1007/s11277-016-3619-8.(Accepted for publication)

9. Binu V. P, Sreekumar A,"An improved Lossless Secret Image Sharing Scheme". National conference in Security Monitoring (NCSM-2013) on 15th & 16th February 2013, Amruta School of Arts & Science, Cochin, Kerala, India http://asaskochi.com/news/wordpress/ ?p=458 (Best Paper Award)

10. Binu V. P, Sreekumar A,"Generalized Secret Sharing using Permutation Ordered Binary System", Sapience'14 - International Conference on Security and Authentication , 27th to 28th March 2014, Sree Narayana Gurukulam College,Ernakulam, Kerala, India ISBN: 978-93-83459-32-2, http://conference.bonfring.org/conferenceproceedings.php?id=1486.

11. Binu V.P.,"Secret Sharing and Applications",(presented as an invited talk), National Seminar on Algebra and Number Theory(NSANT-2014), Pavanatma College, Iduki, Kerala, India. http://www.mathematicspavanatma.org/sites/default/files/Abstract.pdf.

12. Nair D.G., Binu V.P., Kumar, G.S., "An Effective Private Data Storage and Retrieval System using Secret Sharing Scheme Based on Secure Multi-party Computation", International Conference on Data Science & Engineering (ICDSE), 2014 , pp.210,214, 26-28 Aug. 2014 doi: 10.1109/ICDSE.2014.6974639.IEEEXplore.

13. Divya G. Nair, Binu V. P, G. Santhosh Kumar,"An Improved E-Voting Scheme using Secret Sharing based Secure Multi-Party Computation", Eighth International Conference on Computer Communication Networks (ICCN 2014), Banglore, Elsevier, ISBN :9789351072539,P-17.

14. S. R. Sreela, Binu V. P, G. Santhosh Kumar,"Establishing Security in Cheque Truncation System using Secret Image Sharing", Eighth International Conference on Computer Communication Networks (ICCN 2014), Banglore, Elsevier, ISBN :9789351072539, P-29.

1.10 Organization of the Thesis

The work aims to develop secret sharing schemes with several capabilities to ensure security and trust. Two application areas are also mentioned. We have taken care to provide a good account of literature survey and the theoretical background of the topic of study.

The Thesis is organized into 11 chapters. In Chapter 1, we give a brief introduction, survey of secret sharing schemes, preliminaries, review of threshold secret sharing scheme and also the Thesis contribution and organization. This provides a basic introduction to the reader about the topic *secret sharing* and also the thesis contribution and its organization.

In Chapter 2, we present Generalized Secret Sharing Schemes. Review of secret sharing schemes realizing the general access structure is given. This helps to understand about the monotone access structure and secret sharing schemes to realize the generalized access structure. It has got wide applications.

In Chapter 3, several extended capabilities of the secret sharing schemes are mentioned. Verifiability, Cheating detection, Cheater identification etc are the major concern. Knowing about these helps in developing more reliable and secure secret sharing schemes.

In Chapter 4, we present some simple and efficient secret sharing schemes. These scheme are based on simple XOR and number theoretic operations. They have found useful applications in secret image sharing and distributed data storage in cloud. Threshold $(2, 3)$ and $(2, 4)$ schemes are mentioned with algorithms for secret sharing and secret retrieval.

In Chapter 5, a new generalized secret sharing scheme using cumulative array and POB (Permutation Ordered Binary) system is mentioned. The POB system have great potential for efficient secret sharing constructions and are based on XOR operation. Algorithm for (n, n) secret sharing using POB is given. This scheme is then combined with cumulative arrays to construct more general access structure based secret sharing schemes.

In Chapter 6, multi secret sharing schemes are mentioned. Multi secret sharing with generalized access structures are explored in detail and a new multi secret sharing scheme with general access structure is proposed. The scheme is simple and easy to implement.

In Chapter 7, Elliptic curve and Pairing is discussed. This chapter gives a basic introduction about the Elliptic curve and Pairing for the reader. We explored the use of elliptic curve and pairing in developing secret sharing schemes. The use of Elliptic curve and Pairing helps to develop secret sharing schemes with more security and also with various extended capabilities.

In Chapter 8, we present a multi secret sharing scheme having extended capabilities with general access structure based on Elliptic curve and Bilinear pairing. Verifiability, cheating detection and cheater identification is done by using pairing. This scheme outperforms several existing scheme based on elliptic curve and pairing.

In Chapter 9, threshold multi secret sharing using Elliptic curve and Self Pairing are explored. An implementation of the scheme using SAGE and Python is done. The Python modules developed are useful in building cryptographic applications using secret sharing schemes.

In Chapter 10, we present applications based on secret sharing schemes developed in chapter 4 and also Shamir's scheme. We make use of the additive homomorphism in Shamir's secret sharing scheme to implement a secure and efficient E-voting scheme with capability to count individual votes of each contesting candidate. We also present a modified and easy to implement Cheque Truncation System (CTS) using simple secret image sharing technique.

In Chapter 11, we present the summary of major proposals. There are several application areas where the secret sharing schemes can be effectively utilized. We also mention future directions in this regard.

Chapter 2

Generalized Secret Sharing

2.1 Introduction

In the previous chapter, we mentioned that in a (t, n) threshold secret sharing scheme any t of the n participants should be able to determine the secret. A more general situation is to specify exactly which subsets of participants should be able to determine the secret and which subset should not. In this chapter we give the secret sharing constructions based on generalized access structure.

Shamir [190] discussed the case of sharing a secret between the executives of a company such that the secret can be recovered by any three executives, or by any executive and any vice-president, or by the president alone. This is an example of *hierarchical secret sharing* scheme. Shamir's solution for this case is based on an ordinary $(3, n)$ threshold secret sharing scheme. Thus the president receives three shares, each vice-president receives two shares and finally every executive receives a single share.

The above idea leads to the so-called weighted (or multiple shares based) threshold secret sharing schemes. In these schemes, the shares are

pairwise disjoint sets of shares provided by an ordinary threshold secret sharing scheme. Benaloh and Leichter have proven in [15] that there are access structures that cannot be realized using such scheme. The theorem and proof with an example stated by them is given below.

Theorem 2.1.1. *There exist monotone access structure for which there is no threshold scheme exists.*

Proof. Consider the access structure $\mathcal{A}$ defined by the formula $\mathcal{A}_{min} = \{\text{AB,CD}\}$ and assume that a threshold scheme is to be used to divide a secret value K among A, B, C, and D such that only those subsets of A, B, C, D which are in $\mathcal{A}$ can reconstruct K.

Let a, b, c and d respectively denote the weight (number of shares) held by each of A, B, C and D. Since A together with B can compute the secret, it must be the case that $a + b \geqslant t$, where t is the value of the threshold. Similarly, since C and D can together compute the secret, it is also true that $c + d \geqslant t$. Now assume without loss of generality that $a \geqslant b$ and $c \geqslant d$. (If this is not the case, the variables can be renamed). Since $a + b \geqslant t$ and $a \geqslant b, a + a \geqslant a + b \geqslant t$. So $a \geqslant t/2$. Similarly $c \geqslant t/2$. Therefore $a + c \geqslant t$. Thus A together with C can reconstruct the secret value K. This violates the assumption of the access structure. □

Definition 2.1.2. A perfect secret sharing scheme realizing general access structure Γ and sharing a key K among a set of participant $\mathcal{P}$ satisfy the following properties

1. If an authorized set of participant $P \in \mathcal{P}$ pool their shares, then they can determine the secret K.

2. If an unauthorized set of participant $U \in \mathcal{P}$ pool their shares, then they can determine nothing about the secret K.

Several researchers address this problem and introduced secret sharing schemes realizing the general access structure. We give some of the important constructions of secret sharing schemes with general access structure in this chapter.

2.2 Ito, Saito and Nishizeki's construction

Ito, Saito and Nishizeki [107] proposed a secret sharing scheme realizing the general access structure in 1987. The description of the scheme is as follows.
Let K be the secret and $S_1, \ldots, S_n$ are the shadows or shares. Each share S_i is distributed to participants P_i $(i \leqslant i \leqslant n)$ in such a way that

- if $P' = P_{i1}, \ldots, P_{il} \subset \mathcal{P}$ is a qualified subset of persons, then K can be reconstructed from their shadows $S_{i1}, \ldots, S_{il}$ and
- if $P' = P_{i1}, \ldots, P_{il} \subset \mathcal{P}$ is not a qualified subset, then K cannot be reconstructed from their shadows $S_{i1}, \ldots, S_{il}$.

The family of all the qualified subset is called the access structure of the scheme.

Definition 2.2.1. Let $\mathcal{P}$ be the set of participants, then the subset $\mathcal{A} \subseteq 2^{\mathcal{P}}$ contains authorized subsets of participants who can reconstruct the secret is called the *access structure* of the secret sharing scheme.

If P' is a qualified subset, then any subset P'' with $P' \subset P''$ must be so. Thus if $\mathcal{A} \subset 2^{\mathcal{P}}$ is an access structure of a scheme, then $\mathcal{A}$ satisfies $B \subset \mathcal{A}$ *and* $B \subset C \subset \mathcal{P}$ *imply* $C \in \mathcal{A}$. This intuitively means that if a group can recover the secret, so can a larger group. This property of access structure is called *monotone* property. Benaloh and Leichter called such access structures *monotone access structure* in [15].

The family of minimum sets in $\mathcal{A}$ is denoted by $\mathcal{A}_{min}$ or $\mathcal{A}_0$

$$\mathcal{A}_{min} = \{A \in \mathcal{A} | (\forall B \in \mathcal{A} \setminus A)(\neg(B \subseteq A))\}$$

The minimum authorized subsets of $\mathcal{A}$ is denoted by $\mathcal{A}_{min}$ and is called the basis of $\mathcal{A}$. Since $\mathcal{A}$ consist of all subsets of $\mathcal{P}$ that are supersets of a subset in $\mathcal{A}_{min}$, we say that $\mathcal{A}$ is the closure of $\mathcal{A}_{min}(\mathcal{A}_0)$.

$$\mathcal{A} = \{C \subseteq P : B \subseteq C, B \in \mathcal{A}_0\}$$
$$\mathcal{A} = closure(\mathcal{A}_0)$$

Example 2.2.1. Let $\mathcal{P} = \{P_1, P_2, P_3, P_4\}$
and $\mathcal{A}_{min} = \{\{P_1, P_2\}, \{P_3, P_4\}\}$ then $\mathcal{A} = \{\{P_1, P_2\}, \{P_1, P_2, P_3\}, \{P_1, P_2, P_4\}, \{P_1, P_2, P_3, P_4\}, \{P_3, P_4\}, \{P_1, P_3, P_4\}, \{P_2, P_3, P_4\}\}$

Remark 2.2.1. In the case of threshold (t, n) scheme the basis consist of all subsets of exactly t elements. We also use the notation Γ to represent an access structure and Γ_0 to represent the minimal access structure.

Given the general monotone access structure $\mathcal{A}$, the scheme is realized by assigning several shadows of a (k, n) threshold scheme to each person. Shamir's scheme can be used where each authorized set of participants are given sufficient number of shares so that they can retrieve the secret by combining their shares. The multiple assignment scheme realizing the general access structure follows.

Multiple Assignment Scheme[107]

1. Choose two integers k, m and a prime power q such that $k \leqslant m < q$ and let $F = GF(q)$.
2. Choose $a_1, \ldots, a_{k-2}$ in F and $a_{k-1} \in F - \{0\}$ randomly.
3. Let $f(x) = d + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$. (d is the data to be distributed or shared)

4. Choose distinct elements $x_1, \ldots, x_m \in F - \{0\}$, let $d_j = f(x_j)$ $(1 \leqslant j \leqslant m)$, and let $\mathcal{S} = \{(x_1, d_1), \ldots, (x_m, d_m)\}$.

5. Choose $S_i \subset \mathcal{S}$ $(1 \leqslant i \leqslant n)$, and assign S_i to P_i for each $i, 1 \leqslant i \leqslant n = |\mathcal{P}|$.

Let $\mathcal{P}$ be the set of participants and $\mathcal{S}$ be the set of shares then the assignment of shares S_i to P_i is considered as a function $g : \mathcal{P} \longrightarrow 2^{\mathcal{S}}$ such that $g(P_i) = S_i$. The multiple assignment scheme has the following access structure.

$$\mathcal{A} = \left\{ Q \subset \mathcal{P} : \left| \bigcup_{p \in Q} \right| \geqslant k \right\}$$

A simple scheme mentioned by Beimel [10] in which the secret $K \in \{0, 1\}$ and let $\mathcal{A}$ be any monotone access structure. The Dealer distribute the shares of the secret independently for each authorized set $B \in \mathcal{A}$, where $B = \{P_{i1}, \ldots, P_{il}\}$.

- The Dealer chooses $l - 1$ random bits $r_1, \ldots, r_{l-1}$.
- compute $r_l = K \oplus r_1 \oplus r_2 \oplus \cdots \oplus r_{l-1}$ and
- Dealer distributes share r_j to P_{ij}.

For each set $B \in \mathcal{A}$, the random bits are chosen independently and each set in $\mathcal{A}$ can reconstruct the secret by computing the exclusive-or of the bits given to the set. The unauthorized set cannot do so.

The disadvantage with multiple share assignment scheme is that the share size depends on the number of authorized set that contain P_j. A simple optimization is to share the secret K only for minimal authorized sets. Still this scheme is inefficient for access structures in which the number of minimal set is big (Eg:$(n/2, n)$ scheme). The share size grows exponentially in this case.

2.3 The Monotone Formula Construction

Benaloh and Leichter [15] developed a secret sharing scheme for an access structure based on monotone formula. This generalizes the multiple assignment scheme of Ito, Saito and Nishizeki [107]. The idea is to translate the monotone access structure into a monotone formula. Each variable in the formula is associated with a trustee in P and the value of the formula is *true* if and only if the set of variables which are true corresponds to a subset of P which is in the access structure. This formula is then used as a template to describe how a secret is to be divided into shares.

The monotone function contains only AND and OR operator. In order to distribute secret K into shares such that $P_1\ OR\ P_2$ can reconstruct K. In this case P_1 and P_2 can simply both be given values K. If $P_1\ AND\ P_2$ need to reconstruct secret then P_1 can be given value K_1 and P_2 can be given value K_2 such that $K = K_1 + K_2 \pmod m$, $(0 \leqslant K \leqslant m)$, K_1 is chosen randomly from $\mathbb{Z}_m$, K_2 is $(K - K_1) \pmod m$. More exactly for a monotone authorized access structure $\mathcal{A}$ of size n, they defined the set $\mathcal{F}_\mathcal{A}$ as the set of formula on a set of variables $\{v_1, v_2, \ldots, v_n\}$ such that for every $\mathcal{F} \in \mathcal{F}_\mathcal{A}$, the interpretation of $\mathcal{F}$ with respect to an assignation of the variables is true if and only if the variables having the value true correspond to a set $A \in \mathcal{A}$. They have shown that an access structure can be represented as a formula which contains only $\wedge$ operators and $\vee$ operators by splitting the secret across these operators.

Thus we can inductively define the shares of a secret K with respect to a formula $\mathcal{F}$ as follows:

$$Shares(K, \mathcal{F}) = \begin{cases} (K, i), & \text{if } F = v_i,\ 1 \leqslant i \leqslant n; \\ \bigcup_{i=1}^{k} Shares(K, F_i), & \text{if } F = F_1 \vee \cdots \vee F_k; \\ \bigcup_{i=1}^{k} Shares(S_i, F_i), & \text{if } F = F_1 \wedge \cdots \wedge F_k, \end{cases}$$

For the case $\mathcal{F} = F_1 \wedge F_2 \wedge \cdots \wedge F_k$, we can use any (k, k)-threshold secret sharing scheme for deriving some shares $S_1, \ldots, S_k$ corresponding to the secret K.

Remark 2.3.1. We can build a monotone circuit that recognizes the access structure corresponds to the formula and then build the secret sharing scheme from the description of the circuit. This is called monotone circuit construction. Let C be a monotone boolean circuit then the monotone circuit construction yield a perfect secret sharing scheme realizing the access structure $\mathcal{A}(C)$

Example 2.3.1. Consider the access structure $\mathcal{A}_{min} = \{\{P_1, P_2, P_4\}, \{P_1, P_3, P_4\}, \{P_2, P_3\}\}$, the Boolean formula corresponds to $\mathcal{A}_{min}$ is

$$\mathcal{A}_{min} = (P_1 \wedge P_2 \wedge P_4) \vee (P_1 \wedge P_3 \wedge P_4) \vee (P_2 \wedge P_3)$$

Let K is the secret to be shared. The value K is given to the three input wires of the final OR gate. The expression $P_1 \wedge P_2 \wedge P_4$ is implemented by giving the three shares to the input of the AND gate using a threshold $(3, 3)$ scheme. The three input wires are assigned values $a_1, a_2, K - a_1 - a_2$, all arithmetic is done in $\mathcal{Z}_m$. In a similar way, the three input corresponds to $P_1 \wedge P_3 \wedge P_4$ are assigned values $b_1, b_2, K - b_1 - b_2$. Finally the two input wires corresponding to $P_2 \wedge P_3$ are assigned values $c_1, K - c_1$. The Figure 2.1 shows the schematic diagram of the monotone circuit constructed [204]. The shares received by the four participants are

$$\begin{aligned}
P_1 &\longleftarrow (a_1, b_1) \\
P_2 &\longleftarrow (a_2, c_1) \\
P_3 &\longleftarrow (b_2, K - c_1) \\
P_4 &\longleftarrow (K - a_1 - a_2, K - b_1 - b_2)
\end{aligned}$$

It is noted that each of the subsets $\{P_1, P_2, P_4\}, \{P_1, P_3, P_4\}$ and $\{P_2, P_3\}$ can compute the secret K. Any unauthorized subset cannot compute K, either because some necessary piece of random information is missing, or because all the shares possessed by the subset are random.
We can obtain a different scheme realizing the same access structure by rewriting the formula in *conjunctive normal form*. This corresponds to the original construction of Ito et al [107]. The conjunctive normal form is

$$(P_1 \vee P_2) \wedge (P_1 \vee P_3) \wedge (P_2 \vee P_3) \wedge (P_2 \vee P_4) \wedge (P_3 \vee P_4)$$

The following shares are distributed in this case to each participant.

$$\begin{aligned} P_1 &\longleftarrow (a_1, a_2) \\ P_2 &\longleftarrow (a_1, a_3, a_4) \\ P_3 &\longleftarrow (a_2, a_3, K - a_1 - a_2 - a_3 - a_4) \\ P_4 &\longleftarrow (a_4, K - a_1 - a_2 - a_3 - a_4) \end{aligned}$$

Remark 2.3.2. If C is a monotone boolean circuit, then a perfect secret sharing scheme realizing the access structure $\mathcal{A}(C)$ can be built.

2.4 Vector Space Construction

In this section, we will consider an ideal scheme for general access structure know as *vector space construction*. Brickell [36] developed this technique using vector spaces.
Let $\mathcal{A}$ is an access structure and $(\mathbb{Z}_p)^d$ denote the vector space of all d-tuples over $\mathbb{Z}_p$, where p is prime and $d \geqslant 2$. Suppose there is a function

$$\phi : P \longrightarrow (\mathbb{Z}_p)^d$$

which satisfies the property

$$(1, 0, \ldots, 0) \in < \phi(P_i) : P_i \in B > \Leftrightarrow B \in \mathcal{A} \tag{2.1}$$

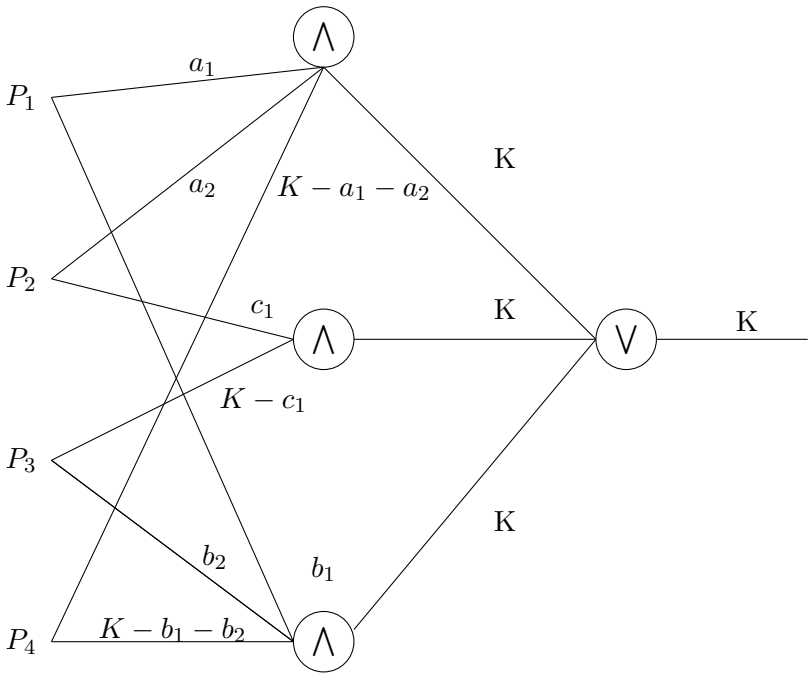

Figure 2.1: Monotone Circuit Construction

This means that the vector $(1, 0, \ldots, 0)$ can be represented as a linear combination (modulo p) of vectors in the set $\{\phi(P_i) : P_i \in B\}$ only if B is an authorized subset. If we have $\{\phi(P_1), \phi(P_2), \ldots, \phi(P_n)\}$ satisfying the above property then the Dealer($\mathcal{D}$) can give $\phi(P_i)$ to P_i. These vectors can also be made public.
Suppose $\mathcal{D}$ wants to share a key $k \in \mathbb{Z}_p$, then he create a vector

$$\bar{a} = (k, a_2, \ldots, a_n)$$

where $a_2, \ldots, a_n$ are chosen independently at random from $\mathbb{Z}_p$. $\mathcal{D}$ then compute the shares as

$$y_i = \bar{a} \cdot \phi(P_i), \quad 1 \leqslant i \leqslant n$$

$\mathcal{D}$ then give y_i to P_i. An ideal secret sharing scheme is constructed in the following way. For every vector $\bar{a} = (a_1, a_2, \ldots, a_d) \in (\mathbb{Z}_p)^d$, a distribution rule $f_{\bar{a}} \in \mathcal{F}_{\bar{a}}$ is defined , where

$$f_{\bar{a}(x)} = \bar{a} \cdot \phi(x)$$

for every $x \in P$, "." is the inner product modulo p operation. The key is given by

$$k = a_1 = \bar{a} \cdot (1, 0, \ldots, 0)$$

If B is an authorized subset then the participants in B can compute k. Since

$$(1, 0, \ldots, 0) \in \langle \phi(P_i) : P_i \in B \rangle$$

we can write

$$\begin{aligned}
(1, 0, \ldots, 0) &= \sum_{i:P_i \in B} c_i \cdot \phi(P_i) \qquad c_i \in \mathbb{Z}_p \\
\bar{a} \cdot (1, 0, \ldots, 0) &= \bar{a} \cdot \sum_{i:P_i \in B} c_i \cdot \phi(P_i) \\
k &= \sum_{i:P_i \in B} c_i (\bar{a} \cdot \phi(P_i)) \\
k &= \sum_{i:P_i \in B} c_i \cdot y_i
\end{aligned}$$

This shows that participants in B can compute the secret k by linear combination of the shares that they hold.

Example 2.4.1. [204] Consider an access structure having the basis

$$\mathcal{A}_{min} = \{\{P_1, P_2, P_3\}, \{P_1, P_4\}\}$$

Let $d = 3$ and define the vectors $\phi(P_i)$ as follows

$$\phi(P_1) = (0, 1, 0)$$
$$\phi(P_2) = (1, 0, 1)$$
$$\phi(P_3) = (0, 1, -1)$$
$$\phi(P_4) = (1, 1, 0)$$

These equations satisfy the property in 2.1.

$$\phi(P_4) - \phi(P_1) = (1, 0, 0)$$

$$\phi(P_2) + \phi(P_3) - \phi(P_1) = (1, 0, 0)$$

Hence

$$(1, 0, 0) \in \langle \phi(P_1), \phi(P_2), \phi(P_3) \rangle$$

and

$$(1, 0, 0) \in \langle \phi(P_1), \phi(P_4) \rangle$$

This shows that the authorized subset will span $(1, 0, 0)$. Now consider an unauthorized subset P_2, P_3, P_4. Suppose that

$$(1, 0, 0) \in \langle \phi(P_2), \phi(P_3), \phi(P_4) \rangle$$

$$(1, 0, 0) = c2\phi(P_2) + c3\phi(P_3) + c4\phi(P_4)$$

where $c_2, c_3, c_4 \in \mathbb{Z}_p$. This is equivalent to set of equations

$$c_2 + c_4 = 1$$
$$c_3 + c_4 = 0$$
$$c_2 - c_3 = 0$$

It is noted these set of equations doesn't have any solutions. This is true for any other unauthorized subsets as well.

Suppose that $p = 127$ then if $k = 99, a_2 = 55, a_3 = 38$. Then the four shares as follows

$$\begin{aligned} y_1 &= 55 \\ y_2 &= 10 \\ y_3 &= 17, \qquad \textit{and} \\ y_4 &= 27 \end{aligned}$$

Suppose P_1, P_2, P_3 wants to compute the secret then it is noted that

$$(1, 0, 0) = \phi(P_2) + \phi(P_3) - \phi(P_1)$$

and Hence

$$k = 1 \cdot y_2 + 1 \cdot y_3 - 1 \cdot y_1 \pmod{127}$$

$$k = 10 + 17 - 55 \pmod{127} = 99$$

Remark 2.4.1. It is noted that **Shamir's (t,n) Threshold Scheme** is a special case of the vector space construction where $d = t$ and $\phi(P_i) = \{1, x, x^2, \ldots, x^{t-1}\}$, where x_i is the x coordinate given to P_i.

2.5 General Model using Distribution Rules

This model is similar to that of Brickell's scheme [38]. In this model, a secret sharing scheme is represented as a special set ($\mathcal{F}$) of distribution rules. The distribution rule is a set of function

$$f : \mathcal{P} \cup D \longrightarrow \mathcal{K} \cup \mathcal{S}$$

which satisfies the conditions $f(D) \in \mathcal{K}$ and $f(p_i) \in \mathcal{S}$, where $\mathcal{K}$-is the Key set, $\mathcal{S}$-share set, $\mathcal{P}$-participant set, $f(D)$ is the secret key being shared and $f(P_i)$ is the share given to P_i.

If $\mathcal{F}$ is a set of distribution rules and $k \in \mathcal{K}$, then the distribution rule corresponds to $f(D) = k$ is

$$\mathcal{F}_k = \{f \in \mathcal{F} : f(D) = k\}$$

If $k \in \mathcal{K}$ is the value of the secret that Dealer($\mathcal{D}$) wishes to share, then $\mathcal{D}$ will choose a random distribution rule $f \in \mathcal{F}_k$ and use it to distribute shares. The set of distribution rules are public knowledge.

The following is an example from [204], where $n = 6$, $\mathcal{K} = \{0, 1\}$ and $\mathcal{S} = \{0, 1, 2\}$. The distribution rules are given in table 2.1 and the basis for the access structure is

$$\mathcal{A}_{min} = \{\{P_1, P_2\}, \{P_2, P_3\}, \{P_3, P_4\}, \{P_4, P_5\}, \{P_5, P_6\}, \{P_6, P_1\}\}$$

For example if P_1, P_2 receive the shares $1, 1$, they know that the distribution

	D	p_1	p_2	p_3	p_4	p_5	p_6
f_1	0	0	0	1	1	2	2
f_2	0	0	0	2	2	1	1
f_3	0	1	1	2	2	0	0
f_4	0	1	1	0	0	2	2
f_5	0	2	2	0	0	1	1
f_6	0	2	2	1	1	0	0
f_7	1	0	1	1	2	2	0
f_8	1	0	2	2	1	1	0
f_9	1	1	2	2	0	0	1
f_{10}	1	1	0	0	2	2	1
f_{11}	1	2	0	0	1	1	2
f_{12}	1	2	1	1	0	0	2

Table 2.1: Distribution rules

function is either f_3 or f_4. But $f_3(D) = f_4(D) = 0$. Knowledge of one share restrict possible distribution rules to four out of 12. However two of these four rules correspond to the secret being 0 and the other two correspond to the secret being 1. If any unauthorized subset, for example $\{P_1, P_4\}$ pool their shares, they will get two possible rules but they corresponds to different values of the secret.

2.6 Monotone Span Program

Brickell's vector space construction [36] always result in efficient general access structure based secret sharing scheme, in which the participant gets one share. However, it is not possible to build vector space based constructions for every general access structure. To combat this, Monotone Span Programs (MSP) [116] can be used to build Linear Secret Sharing Schemes (LSSS) for an arbitrary access structure. A secret sharing scheme is said to be linear, if the Dealer and the participants use only linear operations to compute the shares and the secret. Span programs are linear algebraic model of computation. It is noted that each monotone span program give rise to a linear secret sharing scheme. Beimel proposed this scheme and also defined a lower bound for the share size [9] [11].

Definition 2.6.1. A secret sharing scheme over set of participant P is said to be linear over $\mathbb{Z}_p$ if

- The shares of each participant is a vector in $\mathbb{Z}_p$.
- There exist a matrix $A \in Z_p^{l \times n}$ with row labels $\rho(i) \in P, \forall i \in [l]$

  The shares of secret k are computed as $A.v$, where $v = (k, r_2, \ldots, r_n), k \in \mathbb{Z}_p$ and $r_2, \ldots, r_n \in_R \mathbb{Z}_p$.

Definition 2.6.2. A monotone span program $\mathcal{M}$ is a quadruple $(\mathbb{F}, M, \epsilon, \rho)$, where

- $\mathbb{F}$ is a field.
- M is a $m \times d$ matrix.$M \in \mathbb{F}^{m \times d}$ with $d \leqslant m$.
- $\rho : \{1, \ldots, m\} \to \{1, \ldots, n\}$.
- $\epsilon = (1, 0, \ldots, 0) \in \mathbb{F}^d$ called target vector or object vector.
- ρ labels each row i of M for the participant $p_{\rho(i)} \in P$.
- A monotone span program $\mathcal{M}$ is said to compute an access structure $\mathcal{A}$ if

$$G \in \mathcal{A} \Leftrightarrow \epsilon \in span(M_G)$$

A span program in which the labels of rows are only positive integers is called monotone span program. Monotone span program compute monotone functions. The size of $\mathcal{M}$ is the number of rows in M. The following is a simple example of secret sharing using MSP.

Example 2.6.1. Let $\mathcal{M} = (\mathbb{F}_{17}, M, \epsilon, \rho)$, where

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{pmatrix}$$

$$P_1 = \rho(3), P_2 = \rho(1) = \rho(2), P_3 = \rho(4)$$

Let $\mathcal{B} = \{P_1, P_2\}$ is an authorized access set and $\mathcal{C} = \{P_1, P_3\}$ is an unauthorized access set.So

$$M_B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{pmatrix} \qquad M_C = \begin{pmatrix} 1 & 3 & 9 \\ 1 & 4 & 16 \end{pmatrix}$$

It is noted that

$$(3\ 14\ 1).M_B = \epsilon \ \ i.e., span(M_{\mathcal{B}}) = \epsilon \Rightarrow B \in \mathcal{A}$$

But

$$span(M_C) \notin \epsilon \Rightarrow C \notin \mathcal{A}.$$

2.7 Cumulative Secret Sharing Scheme

Cumulative secret sharing schemes are used to realize arbitrary access structure in the secret sharing scheme. Cumulative schemes were first introduced by Ito et al [107] and then used by several authors to construct a general scheme for arbitrary access structures.

The scheme is based on multiple share assignment. A cumulative boolean array [80] based on unauthorized access structure of the secret sharing scheme is used to distribute multiple shares to each participant. When the authorized set of participants collate, they will be able to retrieve the secret. Unauthorized set of participants cannot retrieve any useful information about the secret. The scheme is also perfect.

Cumulative scheme of Ito et al [107] uses Shamir threshold [190] scheme where as Blakley's scheme is used by Jackson et al in [109]. A simple scheme using cumulative array and Karnin-Greene-Hellman threshold scheme [117] proposed by Ghodosi et al [80]. More details about cumulative secret sharing scheme is mentioned in Chapter 5.

Simmons [199] proposed cumulative map, Jackson [109] proposed a notion of cumulative array. Ghodosi et al [80] introduced simpler and more efficient scheme and also introduced capabilities to detect cheaters. Generalized cumulative arrays in secret sharing is introduced by Long [139].

2.8 Concluding Remarks

In this Chapter we have considered secret sharing scheme realizing the general access structure. The share size is a major concern in the design of generalized secret sharing scheme. The share size grows exponentially in many cases. The generalized secret sharing scheme have found applications recently in cloud based data storage. Attribute based encryptions are gaining more attention. The general strategy is, encryptions are done based on a boolean formula consist of attributes of the user. Secret sharing schemes are used to distribute keys according to a tree structure corresponds to the boolean formula. A particular user will be able to decrypt the data only if his attributes matches with the encrypted attributes. It is noted that general access structure can be converted into a boolean formula in Disjunctive Normal Form. Monotone circuit and Monotone span program can be used to realize a LSSS for any monotone access structure.

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,

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 $E(x * y) = E(x) * E(y)$. 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 ${Z_q}^*$, where q is a prime such that q divides $p - 1$).

- The Dealer generates a random polynomial $q(x)$ of degree $t - 1$, $q(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{t-1}x^{t-1}$, 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.

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

- user i can verify the shares by testing

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

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 $E(a_0) = g^5 \pmod p, E(a_1) = g^2 \pmod p, E(a_2) = g^1 \pmod p$,
$E(a_3) = g^2 \pmod p$. Suitable p must be chosen. User 2 verifies the share by checking
$E(q(2)) = g^{29} \pmod p$ is equal to

$$E(a_0 + (a_1 \times 2^1) + (a_2 \times 2^2) + (a_3 \times 2^3)) = g^{5+4+4+16} = g^{29} \pmod p$$

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

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

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

2. $\mathcal{D}$ 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 $F(x) = S + F_1 x + \cdots + F_{k-1}x^{k-1}$. $\mathcal{D}$ chooses $G_1, \ldots, G_{k-1} \in \mathbb{Z}_q$ at random and uses G_i when committing to F_i, for $i = 1, \ldots, k-1$. $\mathcal{D}$ broadcasts $E_i = E(F_i, G_i)$, for $i = 1, \ldots, k-1$.

3. Let $G(x) = t + G_1 x + \cdots + G_{k-1}x^{k-1}$ and let $t_i = G(i)$, for $i = 1, \ldots, n$. Then $\mathcal{D}$ 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

$$E(S_i, t_i) = \prod_{j}^{k-1} E_j^{i^j}$$

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

$$E_j = E_j^{'} E_j^{''} \qquad , \text{for } j = 0, 1, \ldots, k-1$$

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 $E(S_i, t_i) = \prod_j^{k-1} E_j^{i^j}$ then (S_i, t_i) is also a correct share of S_i. That is

$$g^{S_i} h^{t_i} = E_0 E_1^i \cdots E_{k-1}^{i^{k-1}}$$

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:

$$\begin{aligned} E_j &= E_j^a \qquad , \text{for } j = 0, 1, \ldots, k-1 \\ S_i &= aS_i^{'} \pmod q \\ t_i &= at_i^{'} \pmod q \end{aligned}$$

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:

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

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

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

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

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

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

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

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.

Secret Reconstruction

The protocol consist of two steps

1. *Decryption of shares:* The participant 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 participant 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 scheme 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 select a random value s from $\mathbb{Z}_q$ and then distribute 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 send a response $r = w - \alpha c$ back.

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

In the Initialization phase, the participant select 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 publish the commitments $C_j = g^{a_j}$, for $0 \leqslant j \leqslant t-1$. The Dealer also publishes the encrypted shares $Y_i = y_i^{p(i)}$, for $1 \leqslant i \leqslant n$ and $X_i = \prod_{j=0}^{t-1} C_j^{i^j}$. The consistency of the shares can be proved by using

$$X_i = g^{p_i}, \qquad Y_i = y_i^{p_i}$$

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_{1i}, a_{2i}, $1 \leqslant i \leqslant n$.

In the verification phase, the verifier computes $X_i = \prod_{j=0}^{t-1} C_j^{i^j}$ using C_j values. Using y_i, X_i, Y_i, r_i and c, the verifier computes a_{1i}, a_{2i} as follows.

$$a_{1i} = g^{r_i} X_i^c \qquad a_{2i} = y_i^{r_i} Y_i^c$$

and checks the hash value of X_i, Y_i, a_{1i}, a_{2i} 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 $\lambda_i = \prod_{j \neq i} \frac{j}{j-i}$ 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

key x_i in several round of PVSS. The scheme is also homomorphic. The combined encrypted shares $Y_{i1}Y_{i2}$ 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

[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 $s - v \geqslant t$, then cheating can be detected. If $s - 2v \geqslant t$, 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 ϵ, 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 $q(x) = D + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$.
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 $d_i = q(x_i)$.

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 participant 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 $\mathcal{K} \notin K_0$, 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

P_i in $\mathcal{P}$ 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 $\mathcal{P} - P_i$ 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 knows 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 $1 - (1 - \frac{1}{|S|-1})^{n-k+1} \leqslant \frac{n-k+1}{|S|-1}$. Where S is the set of secrets.

Brickell and Stinson [37] proposed a modified version of the Blckley'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 $\mathcal{D}$ and assume $\mathcal{D} \notin P$. The construction is based on Shamir's threshold secret sharing scheme. When $\mathcal{D}$ wants to share the secret S among the participants in $\mathcal{P}$, he gives a k-dimensional vector $\bar{d_i} \equiv (d_{i,0}, \ldots, d_{i,k-1})$, where $k \leqslant n$, 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, $\mathcal{D}$ 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 $\mathcal{P}$. Let $g_{j,i}$ be non null elements chosen uniformly at random in $GF(q)$, for $i = 1, \ldots, n$ and $i \neq j$. $\mathcal{D}$ 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.

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, participant 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 participant 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.

3.7.1 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 how ever 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 $p_1(i), \ldots, p_{r-1}, p_{r+1}(i), \ldots, p_n(i)$ 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.

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

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.

Chapter 4

Simple and Efficient Secret Sharing Schemes

4.1 Introduction

Secret sharing is a new alternative for outsourcing data in a secure way. It avoids the need for time consuming encryption decryption process and also the complexity involved in key management. The data must also be protected from untrusted cloud service providers. Secret sharing based solution provides secure information dispersal by making shares of the original data and distribute them among different servers. Data from the threshold number of servers can be used to reconstruct the original data. It is often impractical to distribute data among large number of servers. We have to achieve a trade off between security and efficiency. An optimal choice is to use a $(2, 3)$ or $(2, 4)$ threshold secret sharing scheme, where the data are distributed as shares among three or four servers and shares

Some results of this chapter are included in the following paper.
Binu V P, Sreekumar A : "Simple and Efficient Secret Sharing Schemes for Sharing Data and Image." arXiv preprint arXiv:1502.07475 (2015).

from any two can be used to construct the original data. This provides both security, reliability and efficiency. We propose some efficient and easy to implement secret sharing schemes in this regard based on number theory and bitwise XOR. These schemes are also suitable for secure sharing of images. Secret image sharing based on Shamir's schemes are lossy and involves complicated Lagrange interpolation. So the proposed scheme can also be effectively utilized for lossless sharing of secret images.

Confidentiality, reliability and efficiency are the major concerns in secure storage of data. The idea of secret sharing for the information dispersal is suggested by Krawczyk et al [124] in 1994. He proposed a computationally secure secret sharing scheme for the distributed storage using Rabin's [175] information dispersal algorithm and Shamir's secret sharing scheme. However the data is encrypted using a symmetric key encryption and the shares of the key are distributed along with the data shares. The share size is less than the secret in this case compromising the information theoretic security. Abhishek Parak et al [163] in 2010 proposed a space efficient secret sharing scheme for the implicit data security. They incorporated $k-1$ secrets in n shares and any k shares can be used to reconstruct the original secret. A recursive construction using Shamir's scheme is applied in which computational overhead is more. Recursive methods of secret sharing is also mentioned in [82] [162]. Computational secret sharing schemes are proposed for the space efficiency in [8] [179] [214].

Secret sharing based solution provides information theoretical security on confidentiality with out encryption and hence avoid the complexities associated with encryption and key management. It also provides the guarantee on availability of data. Perfect secret sharing needs large amount of computational overhead. We propose specially designed secret sharing schemes using XOR and number theoretic technique to reduce the computation overhead. Unanimous consent schemes are easy to implement

using XOR. But the implementation of a general (t, n) threshold scheme is difficult. Wang et al [215] proposed a scheme based on boolean operation which is used for secret image sharing in 2007. Kurihara et al [125] [126] proposed a $(3, n)$ and a generalized (t, n) secret sharing scheme based on simple XOR operations. Efficient and ideal threshold scheme based on XOR is proposed by Lv et al [140] in 2010. Secret sharing using number theoretic schemes are also developed based on Chinese Reminder Theorem (CRT) [2] [105] [146]. They are not widely used because of the computational complexity. The proposed scheme make use of simple number theoretic concept and the extended Euclid's algorithm [129].

4.2 Proposed Secret Sharing Schemes

The proposed system suggests a method of storing and retrieving private data in a secure and effective manner. The private data include personal information, sensitive information or unique identification etc. The data storage may be a public information storage such as cloud storage server. We propose number theoretic and XOR based scheme for efficient implementation of secret sharing schemes. It can be used for secure storage and retrieval of secret information. Since it does not involve any encryption, the PKI needed for key management can be avoided. Section 4.2.1 contains the detailed description of the secret sharing algorithm using number theoretic concept. Section 4.2.2 explains the XOR based schemes. The algorithms mentioned below are designed to share a file one byte at a time. The scheme can be used to share both textual data and images.

4.2.1 Schemes Based On Number Theory

In this section, the proposed secret sharing schemes which are based on number theoretic concepts are explained in detail. Two threshold secret sharing schemes of order $(2, 3)$ and $(2, 4)$ are proposed. The Algorithm 4.1 is the $(2, 3)$ secret sharing scheme. The retrieval algorithms depends on which shares are used for the reconstruction and are given in Algorithms 4.2, 4.3, 4.4. A $(2, 4)$ secret sharing scheme is mentioned in Algorithm 4.5. The secret revealing algorithms corresponds to different combination of shares are given in Algorithms 4.6, 4.7, 4.8, 4.10 and 4.11. These algorithms use simple number theory concept. In order to find the inverse of a number, extended Euclid's algorithm [39] can be used. The share generation and the secret revealing can be done with a complexity of $O(n)$, where n is the number of bytes to share. Table lookup can be used for faster performance.

Algorithm 4.1: $(2, 3)$ Secret Sharing: Number Theory

Input: Input file S to share.
Output: Three Shares S_1, S_2, S_3 of same size as the original file.

Choose a field Z_p where $p = 257$.
while *not at end of the input file* **do**
    /* read a byte or pixel */
    s=read_byte(S)
    if $s == 0$ **then**
        $s = 256$
    end
    /* find cube root of s */
    $a = s^{171} \pmod{p}$
    r=random(257) /* r is a random number between 1-256 */
    /* generate s_1 ,the share1 pixel */
    $s_1 = r \times a \pmod{p}$
    if $s_1 == 256$ **then**
        $s_1 = 0$
    end
    /* generate s_2,the share2 pixel */
    $s_2 = r^2 \times a \pmod{p}$
    if $s_2 == 256$ **then**
        $s_2 = 0$
    end
    /* generate s_3, the share3 pixel */
    $s_3 = r^4 \times a \pmod{p}$
    if $s_3 == 256$ **then**
        $s_3 = 0$
    end
end

Algorithm 4.2: $(2,3)$ Secret Revealing:Number Theory S_1S_2

Input: Shares S_1 and S_2
Output: The original secret file S which is shared

1 Choose a field Z_p where $p = 257$.
2 **while** *not at end of the input files* **do**
    /* read a byte or pixel from S_1 */
3 s_1=read_byte(S_1)
    /* read a byte or pixel from S_2 */
4 s_2=read_byte(S_2)
5 **if** $s_1 == 0$ **then**
6 $s_1 = 256$
7 **end**
8 **if** $s_2 == 0$ **then**
9 $s_2 = 256$
10 **end**
11 $a = s_1^2 \times s_2^{-1} \pmod{p}$
12 $s = a^3 \pmod{p}$
    /* s is the secret data byte or pixel */
13 **if** $s == 256$ **then**
14 $s = 0$
15 **end**
16 **end**

Algorithm 4.3: $(2, 3)$ Secret Revealing:Number Theory S_1S_3

Input: Shares S_1 and S_3
Output: The original secret file S which is shared

Choose a field Z_p where $p = 257$.
while *not at end of the input files* **do**
    /* read a byte or pixel from S_1 */
    s_1=read_byte(S_1)
    /* read a byte or pixel from S_3 */
    s_3=read_byte(S_3)
    if $s_1 == 0$ **then**
        $s_1 = 256$
    end
    if $s_3 == 0$ **then**
        $s_3 = 256$
    end
    $s = s_1^4 \times s_3^{-1} \pmod{p}$
    /* s is the secret data byte or pixel */
    if $s == 256$ **then**
        $s = 0$
    end
end

Algorithm 4.4: $(2, 3)$ Secret Revealing:Number Theory S_2S_3

Input: Shares S_2 and S_3
Output: The original secret file S which is shared

1 Choose a field Z_p where $p = 257$.
2 **while** *not at end of the input files* **do**
    /* read a byte or pixel from S_2 */
3   s_2=read_byte(S_2)
    /* read a byte or pixel from S_3 */
4   s_3=read_byte(S_3)
5   **if** $s_2 == 0$ **then**
6     $s_2 = 256$
7   **end**
8   **if** $s_3 == 0$ **then**
9     $s_3 = 256$
10 **end**
11 $a = s_2^2 \times s_3^{-1} \pmod{p}$
    /* s is the secret data byte or pixel */
12 $s = a^3 \pmod{p}$
13 **if** $s == 256$ **then**
14    $s = 0$
15 **end**
16 **end**

Algorithm 4.5: $(2, 4)$ Secret Sharing:Number Theory

Input: Input file S to share.
Output: Four Shares S_1, S_2, S_3, S_4 of same size as the original file.

Choose a field Z_p where $p = 257$.
while *not at end of the input file* **do**
    s=read_byte(S) `/* read a byte or pixel */`
    if $s == 0$ **then**
        $s = 256$
    end
    r=random(257) `/* r is a random number between 1-256 */`
    `/*` s_1 `is the share1 pixel */`
    $s_1 = r$
    if $s_1 == 256$ **then**
        $s_1 = 0$
    end
    `/*` s_2 `is the share2 pixel */`
    $s_2 = r \times s \pmod{p}$
    if $s2 == 256$ **then**
        $s2 = 0$
    end
    `/*` s_3 `is the share3 pixel */`
    $s_3 = r^2 \times s \pmod{p}$
    if $s_3 == 256$ **then**
        $s_3 = 0$
    end
    `/*` s_4 `is the share4 pixel */`
    $s_4 = r^3 \times s \pmod{p}$
    if $s_4 == 256$ **then**
        $s_4 = 0$
    end
end

Algorithm 4.6: $(2, 4)$ Secret Revealing:Number Theory S_1S_2

Input: Shares S_1 and S_2
Output: The original secret file S which is shared

```
1  Choose a field Z_p where p = 257.
2  while not at end of the input files do
       /* read a byte or pixel from S_1                    */
3      s_1=read_byte(S_1)
       /* read a byte or pixel from S_2                    */
4      s_2=read_byte(S_2)
5      if s_1 == 0 then
6          s_1 = 256
7      end
8      if s_2 == 0 then
9          s_2 = 256
10     end
11     s = s_1^{-1} × s_2 (mod p)
12     if s == 256 then
13         s = 0
14     end
15 end
```

Algorithm 4.7: $(2, 4)$ Secret Revealing:Number Theory S_1S_3

Input: Shares S_1 and S_3
Output: The original secret file S which is shared

```
1  Choose a field Z_p where p = 257.
2  while not at end of the input files do
       /* read a byte or pixel from S_1                    */
3      s_1=read_byte(S_1)
       /* read a byte or pixel from S_3                    */
4      s_3=read_byte(S_3)
5      if s_1 == 0 then
6          s_1 = 256
7      end
8      if s_3 == 0 then
9          s_3 = 256
10     end
11     s = (s_1^2)^{-1} × s_3 (mod p)
12     if s == 256 then
13         s = 0
14     end
15 end
```

Algorithm 4.8: $(2, 4)$ Secret Revealing:Number Theory S_1S_4

Input: Shares S_1 and S_4
Output: The original secret file S which is shared

Choose a field Z_p where $p = 257$.
while *not at end of the input files* **do**
    `/* read a byte or pixel from` S_1 `*/`
    s_1=read_byte(S_1)
    `/* read a byte or pixel from` S_4 `*/`
    s_4=read_byte(S_4)
    if $s_1 == 0$ **then**
        $s_1 = 256$
    end
    if $s_4 == 0$ **then**
        $s_4 = 256$
    end
    $s = (s_1^3)^{-1} \times s_4 \pmod{p}$
    `/*` s `is the secret byte or pixel */`
    if $s == 256$ **then**
        $s = 0$
    end
end

Algorithm 4.9: $(2, 3)$ Secret Revealing:Number Theory S_2S_3

Input: Shares S_2 and S_3
Output: The original secret file S which is shared

```
1  Choose a field Z_p where p = 257.
2  while not at end of the input files do
3      s_2=read_byte(S_2) /* read a byte or pixel from S_2      */
4      s_3=read_byte(S_3) /* read a byte or pixel from S_3      */
5      if s_2 == 0 then
6          s_2 = 256
7      end
8      if s_3 == 0 then
9          s_3 = 256
10     end
11     s = s_2^2 × s_3^{-1}  (mod p)
12     if s == 256 then
13         s = 0
14     end
15 end
```

Algorithm 4.10: $(2, 4)$ Secret Revealing:Number Theory S_2S_4

Input: Shares S_2 and S_4
Output: The original secret file S which is shared

```
1  Choose a field Z_p where p = 257.
2  while not at end of the input files do
       /* read a byte or pixel from S_2                    */
3      s_2=read_byte(S_2)
       /* read a byte or pixel from S_4                    */
4      s_4=read_byte(S_4)
5      if s_2 == 0 then
6          s_2 = 256
7      end
8      if s_4 == 0 then
9          s_4 = 256
10     end
11     s = sqrt(s_2^3 × s_4^{-1} (mod p))
       /* s is the secret byte or pixel                    */
12     if s == 256 then
13         s = 0
14     end
15 end
```

Algorithm 4.11: $(2, 4)$ Secret Revealing:Number Theory S_3S_4

Input: Shares S_3 and S_4
Output: The original secret file S which is shared

```
1  Choose a field Z_p where p = 257.
2  while not at end of the input files do
3      s_3=read_byte(S_3) /* read a byte or pixel from S_3      */
4      s_4=read_byte(S_4) /* read a byte or pixel from S_4      */
5      if s_3 == 0 then
6          s_3 = 256
7      end
8      if s_4 == 0 then
9          s_4 = 256
10     end
11     s = s_3^3 × (s_4^2)^{-1}  (mod p)
12     if s == 256 then
13         s = 0
14     end
15 end
```

4.2.2 Schemes based on XOR

An (n, n) scheme using XOR can easily be setup by creating $n - 1$ random shares of same size as the secret and the n^{th} share as the XOR of these $n - 1$ shares and the secret k. The secret can be revealed by simply XOR ing of all these shares. In this, we propose two schemes based on XOR. An ideal $(2, 3)$ scheme where the size of the share is same as that of the secret is mentioned in Algorithm 4.16 and a non ideal scheme which is also not perfect is mentioned in Algorithm 4.12. In this the size of the share is reduced to half. The scheme can be used when the storage become a constraint. The secret sharing and revealing can be done in time $O(n)$, where n is the number of bytes to share. The secret reconstruction corresponds to different combination of shares in the non ideal scheme are

mentioned in Algorithms 4.13, 4.14 and 4.15. The ideal schemes are mentioned in Algorithms 4.17, 4.18 and 4.19.

Algorithm 4.12: $(2, 3)$ XOR secret sharing-non ideal

Input: Secret file S to share.
Output: Three shares S_1, S_2 and S_3 of half the size of S.

while *not at end of the input files* **do**
    s=read_byte(S) `/* read a byte or pixel from` S `*/`
    bs=binary(s) `/*` bs `is the binary representation of` s `*/`
    `/* odd bits of` bs `taken as share1 data nibble` s_1 `*/`
    s_1=odd_bits(bs)
    `/* even bits of` bs `taken as share2 data nibble` s_2 `*/`
    s_2=even_bits(bs)
    `/* share3 nibble is formed by xoring` s_1 `and` s_2 `*/`
    $s_3 = s_1 \oplus s_2$
end

Algorithm 4.13: $(2, 3)$ XOR secret revealing S_1S_2-non ideal

Input: Share S_1 and S_2
Output: The original secret file S which is shared.

while *not at end of the input files* **do**
    s_1=read_byte(S_1) `/* read a byte or pixel from` S_1 `*/`
    s_2=read_byte(S_2) `/* read a byte or pixel from` S_2 `*/`
    s = intermix(s_1, s_2) `/* intermix the bits of` s_1 `and` s_2 `to construct the secret byte */`
end

Algorithm 4.14: $(2, 3)$ XOR secret revealing S_1S_3-non ideal

Input: Share S_1 and S_3
Output: The original secret file S which is shared.

```
1 while not at end of the input files do
2     s1=read_byte(S1) /* read a byte or pixel from S1      */
3     s3=read_byte(S3) /* read a byte or pixel from S3      */
4     s2 = s1 ⊕ s3
      /* intermix the bits of s1 and s2 to construct the
         secret byte                                        */
5     s = intermix(s1, s2)
6 end
```

Algorithm 4.15: $(2, 3)$ XOR secret revealing S_2S_3-non ideal

Input: Share S_2 and S_3
Output: The original secret file S which is shared.

```
1 while not at end of the input files do
2     s2=read_byte(S2) /* read a byte or pixel from S2      */
3     s3=read_byte(S3) /* read a byte or pixel from S3      */
4     s1 = s2 ⊕ s3
5     s = intermix(s1, s2) /* intermix the bits of s1 and s2 to
         construct the secret byte                          */
6 end
```

Algorithm 4.16: $(2, 3)$ XOR Ideal Secret Sharing

Input: Input file S to share.
Output: Three Shares SH_1, SH_2, SH_3 of same size as the original file.

while *not at end of the input file* **do**
    s=read_byte(S) /* read a byte or pixel */
    r=random(257) /* r is a random number between 0-256 */
    s_1, s_2=split_two(s) /* split s into 2 nibbles */
    r_1, r_2=split_two(r) /* split r into 2 nibbles */
    $s0 = 0000$ /* a dummy variable initialized to zero */
    $sh_1 = s0 \oplus r_1 || s_2 \oplus r_2$
    /* sh_1 is the share1 pixel and '||' is concatenation operation */
    $sh_2 = s_1 \oplus r_1 || s_0 \oplus r_2$
    /* sh_2 is the share2 pixel and '||' is concatenation operation */
    $sh_3 = s_2 \oplus r_1 || s_1 \oplus r_2$
    /* sh_3 is the share3 pixel and '||' is concatenation operation */
end

Algorithm 4.17: $(2, 3)$ XOR Ideal Secret Recovery SH_1, SH_2

Input: Shares SH_1 and SH_2
Output: Original secret S that is shared

while *not at end of the input files* **do**
    sh_1=read_byte(SH_1) `/* read a byte or pixel */`
    sh_2=read_byte(SH_2)
    x_1, y_1=split_two(sh_1) `/* split a byte into 2 nibbles */`
    x_2, y_2=split_two(sh_2)
    $s_1 = x_1 \oplus x_2$
    $s_2 = y_1 \oplus y_2$
    $s = s_1 || s_2$
end

Algorithm 4.18: XOR Ideal Secret Recovery SH_1, SH_3

Input: Shares SH_1 and SH_3
Output: Original secret S that is shared

while *not at end of the input files* **do**
    sh_1=read_byte(SH_1) `/* read a byte or pixel */`
    sh_3=read_byte(SH_3)
    x_1, y_1=split_two(sh_1) `/* split a byte into 2 nibbles */`
    x_3, y_3=split_two(sh_3)
    $s_2 = x_1 \oplus x_3$
    $s_1 = y_1 \oplus y_3 \oplus s_2$
    $s = s_1 || s_2$
end

Algorithm 4.19: $(2,3)$ XOR Ideal Secret Recovery SH_2, SH_3

Input: Shares SH_2 and SH_3
Output: Original secret S that is shared

while *not at end of the input files* **do**
    sh_2=read_byte(SH_2) `/* read a byte or pixel */`
    sh_3=read_byte(SH_3)
    x_2, y_2=split_two(sh_2)`/* split a byte into 2 nibbles */`
    x_3, y_3=split_two(sh_3)
    $s_1 = y_2 \oplus y_3$
    $s_2 = x_2 \oplus x_3 \oplus s_1$
    $s = s_1 || s_2$
end

4.3 Conclusion

The confidentiality, availability and performance requirement of storage system is addressed in this chapter. Secret sharing based solutions provides information theoretic security and also provides trust and reliability. We developed simple XOR and number theory based schemes which are easy to implement. This will greatly improve the performance of the system. The storage requirement can also be reduced if we use scheme where the share size is only half the size of the original secret. The schemes mentioned in this chapter are simple and easy to implement when sharing data with third party servers. A $(2,3)$ or $(2,4)$ threshold secret sharing schemes are the best choices. The cost factor can also be reduced by using the non ideal XOR based scheme, where the share size is reduced to half but the information theoretic security is compromised. A secret vector which indicates the share number that each server stores can be kept secret. A simple substitution or transposition cipher can also be used as a preprocessing step before

sharing the file for additional security. The use of these schemes can be further explored in other areas where the threshold required is as specified in the algorithm. We have used this schemes for efficient sharing of secret images also.

Chapter 5

POB and Generalized Secret Sharing

5.1 Introduction

In the previous chapters we have seen the secret sharing schemes having threshold and generalized access structures. This chapter explores the construction of an efficient secret sharing scheme realizing the general access structure. An efficient (n, n) threshold secret sharing scheme is proposed by Sreekumar et al in [202] using a specially designed number system called Permutation Ordered Binary (POB) number system. We are combining this scheme with the concept of cumulative arrays proposed by Ito et al in [107], to build secret sharing scheme with more generalized access structure.

In this chapter we give a brief introduction about general access

Some results of this chapter are included in the following paper.
Binu V P, Sreekumar A : "Generalized Secret Sharing using Permutation Ordered Binary System",Sapience'14 - International Conference on Security and Authentication ISBN: 978-93-83459-32-2

structure based secret sharing schemes, more specifically cumulative arrays are dealt in detail. The POB construction and (n, n) secret sharing using POB is explained next. The chapter ends with the algorithm of the proposed secret sharing scheme using POB and cumulative arrays.

There exist several monotone access structure for which there is no threshold scheme possible. Benaloh and Leichter had proven in [15] that, there are access structures that cannot be realized using threshold scheme. So secret sharing based on arbitrary monotone increasing access structure was a challenge. Several researchers address this problem and introduced secret sharing schemes realizing the general access structure. The most efficient and easy to implement scheme was Ito, Saito, Nishizeki's [107] construction. It is based on Shamir's scheme. The idea is to distribute shares to each authorized set of participants using multiple assignment scheme where more than one share is assigned to a participant if he belongs to more than one minimal authorized subset.

The disadvantage with multiple share assignment scheme is that the share size depends on the number of authorized set that contain P_j. A simple optimization is to share the secret S only for minimal authorized sets. Still this scheme is inefficient. Benaloh and Leichter [15] developed a secret sharing scheme for an access structure based on monotone formula. This generalizes the multiple assignment scheme of Ito, Saito and Nishizeki [107]. The idea is to translate the monotone access structure into a monotone formula. Each variable in the formula is associated with a trustee in $\mathcal{P}$ and the value of the formula is *true* if and only if the set of variables which are true corresponds to a subset of $\mathcal{P}$ which is in the access structure. This formula is then used as a template to describe how a secret is to be divided into shares.

Brickell [36] developed some ideal schemes for general access structure based secret sharing using vector spaces. Stinson [204] introduced a monotone circuit construction based on monotone formula and also the

construction based on public distribution rules. Benaloh's scheme was generalized by Karchmer and Wigderson [116]. They showed that if an access structure can be described by a small Monotone Span Program (MSP), then it has an efficient Linear Secret Sharing Scheme (LSSS). The proposed generalized secret sharing scheme make use of cumulative arrays for the generalized secret sharing which is given in the next section.

5.2 Cumulative Secret Sharing Scheme

Cumulative secret sharing schemes provide a secret sharing capability using an arbitrary access structure. Cumulative schemes were first introduced by Ito et al [107] and then used by several authors to construct a general scheme for arbitrary access structures. Simmons [199] proposed cumulative map, Jackson [109] proposed the notion of cumulative array. Ghodosi et al [80] introduced simple and more efficient scheme. The scheme also having the capabilities to detect cheaters. Generalized cumulative arrays in secret sharing is introduced by Long [139].

Definition 5.2.1. Let $\mathcal{A}$ be a monotone authorized access structure on a set of participants $\mathcal{P}$. A cumulative scheme for the access structure $\mathcal{A}$ is map $\alpha : \mathcal{P} \longrightarrow 2^S$, where S is some set such that for any $\mathcal{A} \subseteq P$,

$$\bigcup_{P_i \in \mathcal{A}} \alpha(P_i) = S$$

The scheme is represented using $|\mathcal{P}| \times |S|$ array $M = [m_{ij}]$, where row i of the matrix M is indexed by $p_i \in P$ and column j of the matrix M is indexed by an element $s_j \in S$, such that $m_{ij} = 1$ if and only if P_i is given s_j, otherwise $m_{ij} = 0$.

Definition 5.2.2. Let $\mathcal{A}$ be an access structure over the set of participants $\mathcal{P} = \{P_1 \ldots, P_n\}$ and $\mathcal{A}_{min} = \{\mathcal{A}_1, \ldots, \mathcal{A}_l\}$ is the set of all

minimal set of $\mathcal{A}$. Then the **incident array** of $\mathcal{A}$ is a $l \times n$ Boolean matrix $I_{\mathcal{A}} = [a_{ij}]$ defined by,

$$a_{ij} = \begin{cases} 1 & \text{if} \quad P_j \in \mathcal{A}_i \\ 0 & \text{if} \quad P_j \notin \mathcal{A}_i \end{cases}$$

for $1 \leqslant j \leqslant n$ and $1 \leqslant i \leqslant l$

Definition 5.2.3. Let $\mathcal{A}^c_{max} = \{B_1, \ldots, B_m\}$ be the set of all maximal unauthorized sets. The **cumulative array** $C_{\mathcal{A}}$ for $\mathcal{A}$ is an $n \times m$ matrix $C_{\mathcal{A}} = [b_{ij}]$, where each row of the matrix is indexed by a participant $P_i \in \mathcal{P}$ and each column is indexed by a maximal unauthorized set $B_j \in \mathcal{A}^c_{max}$, such that the entries b_{ij} satisfy the following:

$$b_{ij} = \begin{cases} 0 & \text{if} \quad P_i \in B_j \\ 1 & \text{if} \quad P_i \notin B_j \end{cases}$$

for $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$.

It is noted that following theorem is true and proved in [80].

Theorem 5.2.1. *If α_i is the i^{th} row of the cumulative array $C_{\mathcal{A}}$, then $\alpha_{i1} + \cdots + \alpha_{it} = \overrightarrow{1}$ if and only if $\{P_{i1}, \ldots, P_{it}\} \in \mathcal{A}$*

cumulative scheme of [107] and [46] uses Shamir's threshold [190] scheme, where as Blakley's scheme is used in [200] and [109]. A simple scheme using cumulative array and Karnin-Greene-Hellman threshold scheme [117], proposed by Ghodosi et al [80] is given below.

The Scheme

Let $\mathcal{A}_{min} = \mathcal{A}_1 + \cdots + \mathcal{A}_\ell$ be a monotone access structure over the set of participants $\mathcal{P} = P_1, \ldots, P_n$. Let $\mathcal{A}^c_{max} = B_1 + \cdots + B_m$ be the set of

maximal unauthorized subsets. The share distribution and reconstruction phases are given below.

Share Distribution Phase

1. The dealer $\mathcal{D}$ constructs the $n \times m$ cumulative array $C_{\mathcal{A}} = [b_{ij}]$, where n is the number of participants and m is the cardinality of $\mathcal{A}^c_{max}$.
2. $\mathcal{D}$ used Karnin-Greene-Hellman (m, m) threshold scheme [117] to generate m shares $S_j, 1 \leqslant j \leqslant m$.
3. $\mathcal{D}$ gives shares S_j privately to participant P_i if and only if $b_{ij} = 1$.

Secret Reconstruction Phase

1. The secret can be recovered by every set of participants in the access structure using the modular addition over $\mathbb{Z}_q$.

Example 5.2.1. Let $n = 4$ and $\mathcal{A}_{min} = \{\{1, 2\}, \{3, 4\}\}$. In this case, we obtain that $\mathcal{A}^c_{max} = \{\{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}\}$ and $m = 4$.

The cumulative array for the access structure $\mathcal{A}$ is,

$$\mathcal{C}_{\mathcal{A}} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

The Dealer then construct a $(4, 4)$ threshold scheme and generate four shares s_1, s_2, s_3, s_4 such that the secret $S = s_1 + s_2 + s_3 + s_4$. The share s_1 is then assigned to P_2 and P_4. The share s_2 is then given to P_2 and P_3. The share s_3 is given to P_1 and P_4. Finally the share s_4 is given to P_1 and P_3. Let S_1, S_2, S_3 and S_4 be the shares of each participant P_1, P_2, P_3

and P_4 respectively. Then $S_1 = \{s_3, s_4\}$, $S_2 = \{s_1, s_2\}$, $S_3 = \{s_2, s_4\}$ and $S_4 = \{s_1, s_3\}$. it is noted that participant P_1 and P_2 can together have all the 4 shares to reconstruct the secret S. Same is the case for the participant P_3 and P_4. This shows that only the authorized set of participants can obtain the shared secret S. The scheme is also perfect.

5.3 Permutation Ordered Binary(POB) System

The POB system is developed by Sreekumar et al [202] for the efficient distributed storage and retrieval of secret data using secret sharing technique. The share generation and reconstruction involves simple XOR operations. The share generation algorithm is linear and depends on the size of the secret. The shares generated are 1 bit less than the secret, but still provides the same level of security and hence a reduction in storage space can be achieved. The POB system can be used to implement an (n, n) threshold secret sharing scheme very efficiently. In this section we will give a brief introduction about the POB system construction and the (n, n) secret sharing scheme using POB. More details about the POB system can be found in [201].

5.3.1 POB system construction

The POB number system is very special in which all the numbers in the range $0, \ldots, \begin{pmatrix} n \\ r \end{pmatrix} - 1$, are represented by a binary string $B = b_{n-1}b_{n-2}\ldots b_0$, of length n and having exactly r 1s and is represented as $POB(n, r)$, where n and r are positive integers and $n \geqslant r$.

Each POB number B is associated with a value $V(B)$ and is computed as a sum of the positional values of each bit b_j in the number. The positional values of each bit is computed as

$$b_j.\begin{pmatrix} j \\ p_j \end{pmatrix}, \quad where, \quad p_j = \sum_{i=0}^{j} b_i \ ,$$

and the value $V(B)$ represented by the POB-number B is,

$$V(B) = \sum_{j=0}^{n-1} b_j.\begin{pmatrix} j \\ p_j \end{pmatrix} \tag{5.1}$$

The POB representation is unique and it can be proved that, since exactly $\begin{pmatrix} n \\ r \end{pmatrix}$ such binary strings exist, each number will have a distinct representation. Each POB number represented using binary $POB(n, r)$ will have a unique value $V(B)$ computed as per equation 5.1. The POB number is denoted by the suffix 'p', in order to distinguish it from a binary number.

Example 5.3.1. $POB(9, 4) = 100101010_p$ be a POB number having 9 bits and 4 ones. Its value $V(B)$ is computed as follows

$$V(B) = \begin{pmatrix} 8 \\ 4 \end{pmatrix} + \begin{pmatrix} 5 \\ 3 \end{pmatrix} + \begin{pmatrix} 3 \\ 2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$V(B) = 70 + 10 + 3 + 1 = 84$$

There exist efficient algorithm to convert POB value into corresponding POB number. The following algorithm will generate the POB number from the given POB value.

Algorithm 5.1: Convert POB value to POB number

Input: n, r and $V(B)$-The value of the POB number B
Output: The $POB(n, r)$ number $B = b_{n-1}b_{n-2}\ldots b_0$ corresponds to the value $V(B)$

```
1 let j = n and temp = V(B)
2 for k = r down to 1 do
3     repeat
4         j = j − 1
5         p = ( j )
              ( k )
            if temp ⩾ p then
              temp = temp − p
              b_j = 1
            else
              b_j = 0
            end if
6     until b_j == 1;
7 end
```

The POB number system developed have great potential for secret sharing. Each POB number is considered as a balanced string which contains same number of ones and zeros. These balanced strings are useful for sharing images where each pixel will have uniform code as share. The POB system can be used to develop an (n, n) secret sharing scheme very efficiently, which is explored in the next section.

5.3.2 (n, n) secret sharing scheme using POB

It is noted that efficient (n, n) schemes are the building blocks of secret sharing schemes having more generalized monotone access structure. There are several proposals for generalized access structure based secret sharing using (n, n) threshold secret sharing scheme. Karnin et al [117] developed an unanimous consent scheme which is used in the Benaloh's and Leichter scheme [15]. Ito et al [107] used Shamir's (n, n) threshold scheme. POB system can be used for developing an efficient (n, n) scheme which is secure and reliable. The scheme is perfect and also the POB numbers are balanced strings. They always contains same number of ones and zeros.

The secret sharing algorithm takes a secret of size 8 bits and expands it to 9 bits by adding an extra bit at random position r. Algorithm 5.3 explains this procedure. The secret sharing scheme uses $POB(9, 4)$ scheme and produces POB numbers of size 9 bits with 4 ones in it. The value $V(B)$ corresponds to each POB number B is computed and is used as the share value, which is only 7 bits in size. The details of share generation is given in the Algorithm 5.2. The secret reconstruction technique is mentioned in Algorithm 5.4. The input POB values are converted to POB numbers using the Algorithm 5.1. The original secret K can be easily reconstructed using simple XOR operation and also the extra bit at position r can be easily removed. This random location is computed based on one of the POB number selected i.e., A_2. The random location r is computed as, $r = \left\lceil \frac{V(A_2)+1}{14} \right\rceil$. The POB value $V(A_2)$ will have value in the range [1..125]. So r will take values in the range [1..9].

Algorithm 5.2: (n, n) Secret Sharing using POB

Input: A single byte string $K = K_1K_2K_3 \ldots K_8$.
Output: n shares $S_1, S_2, \ldots, S_n$ of length 7 bits each

Choose $n-2, POB(9,4)$-numbers randomly $A_i, 2 \leqslant i \leqslant n-1$.
Let $r = \left\lceil \frac{V(A_2)+1}{14} \right\rceil$
/* The input string K is expanded to 9 bits by inserting an extra bit at position r using expand algorithm. */
$T = expand(K)$
Let $W = T \oplus A_2 \oplus A_3 \oplus \ldots \oplus A_{n-1}$
/* Compute the bits of A_1 using W */
$noOfOne = 0$
for *i = 1 to* 9 **do**
    if $W_i == 1$ **then**
        $noOfOne = noOfOne + 1$
        if $noOfOne$ is odd **then**
            $A_1[i] = 1$
        else
            $A_1[i] = 0$
        end if
    end if
end
Randomly assign the remaining null bits of A_1 to 0 or 1
/* Finally A_1 consists of four 1s and five 0s */
$A_n = W \oplus A_1$
/* generate the n shares */
for $i = 1$ *to* n **do**
    $S_i = V(A_i)$.
end

Algorithm 5.3: expand algorithm

Input: binary string of 8 bits-K
Output: string of 9 bits-T

Compute the binary string $T = T_1T_2 \dots T_9$
for *i=0 to 8* **do**

$$T_i = \begin{cases} K_i, & \text{if } i < r \\ K_{i-1}, & \text{if } i > r \\ 0, & \text{if i = r and K is even parity} \\ 1, & \text{if i = r and K is odd parity} \end{cases}$$

end
return T

Algorithm 5.4: (n, n) POB secret recovery

Input: n shares S_1, S_2, ...,S_n of length 7 bits each.
Output: The secret $K = K_1K_2K_3 \dots K_8$.

Let $A_1, A_2, \dots A_n$ be the POB-numbers corresponding to the shares $S_1, S_2, \dots, S_n$ respectively.
$r = \left\lceil \frac{S_2+1}{14} \right\rceil$
Compute $T = A_1 \oplus A_2 \oplus A_3 \oplus \dots \oplus A_n$
Let $T = T_1T_2 \dots T_9$
for *i= 1 to 8* **do**
  if $i \geqslant r$ **then**
    $j = i + 1$
  else
    $j = i$
  end if
  $K_i = T_j$
end
The recovered secret is $K = K_1K_2K_3 \dots K_8$

Example 5.3.2. Let us consider a $(4, 4)$ threshold secret sharing scheme. The secret to be shared is $K = 10110110$.

Randomly choose two $POB(9, 4)$ numbers $\{A_2, A_3\}$.

$$\begin{aligned} A_2 &= 101100010 \; and \\ A_3 &= 010101001 \end{aligned}$$

Let the random number $r = \left\lceil \frac{V(A_2)+1}{14} \right\rceil = \lceil \frac{102}{14} \rceil = 8$.

The secret K is expanded to 9 bits string T as per the Algorithm 5.3

$$T = 101101110$$

$$W = T \oplus A_2 \oplus A_3$$

$$W = 10110111 \oplus 101100010 \oplus 010101001$$

$$W = 010100101$$

Now we will compute A_1 using the step 6 of the Algorithm 5.2

$$A_1 = *1 * 0 * *1 * 0$$

randomly fill rest of the bits so that there will be 4 ones and 5 zeros

$$A_1 = 110010100$$

Now we will compute A_4

$$A_4 = W \oplus A_1$$

$$A_4 = 010100101 \oplus 110010100 = 100110001$$

The shares are $V(A_1), V(A_2), V(A_3)$ and $V(A_4)$

$$S_1 = 113 = 1110001$$

$$S_2 = 101 = 1100101$$

$$S_3 = 48 = 0110000$$

$$S_4 = 86 = 1010110$$

Secret Recovery

The secret can be recovered by using simple XOR operation. From the shares, the POB value can be found which is then converted into POB numbers using the Algorithm 5.1.

Compute

$$T = A_1 \oplus A_2 \oplus A_3 \oplus A_4$$

$$T = 110010100 \oplus 101100010 \oplus 010101001 \oplus 100110001$$

$$T = 101101110$$

Deleting the 8^{th} bit, we get secret as

$$K = 10110110$$

5.4 Proposed Generalized Secret Sharing Scheme

The proposed scheme make use of (n, n) scheme using POB and cumulative arrays to efficiently share a secret according to a generalized access structure. The detailed algorithm for secret sharing according to the generalized access structure is given in 5.5. The secret reconstruction is mentioned in the Algorithm 5.6.

Algorithm 5.5: Generalized Secret Sharing using POB

Input: Access structure corresponds to a secret sharing scheme.
Output: Shares for each participants corresponds to the given access structure.

1 Find the maximal unauthorized set $\mathcal{A}^c_{max}$ corresponds to the given access structure.
2 The dealer $\mathcal{D}$, constructs the $n \times m$ cumulative array $C_{\mathcal{A}} = [b_{ij}]$, where n is the number of participants and m is the cardinality of $\mathcal{A}^c_{max}$.
3 $\mathcal{D}$ uses (m, m) POB scheme to generate m shares $S_j, 1 \leqslant j \leqslant m$.
4 $\mathcal{D}$ gives shares S_j privately to participant P_i if and only if $b_{ij} = 1$.

Algorithm 5.6: Secret Reconstruction using POB

Input: Shares corresponds to the participants.
Output: Shared secret corresponds to the authorized set or error.

1 From the shares generate the POB number.
2 The secret can be reconstructed by XORing the shares corresponds to an authorized set of participant.
3 For an unauthorized set the algorithm gives an error else the shared secret $K$is returned.

One of the issue with general access structure based secret sharing scheme is the number of shares each participant has to maintain. The storage become a major constraint here. For every eight bytes of secret one byte of storage requirement is saved by using the POB number system based threshold secret sharing. The secret reconstruction needs only simple XOR operation. The operations performed in the POB, that is bit expansion, conversion of POB number to POB value and vice versa

can be performed in time proportional to number of bytes in the secret values. This makes the scheme more suitable for cumulative array based generalized secret sharing compared with Benaloh's and Leichter scheme [15] and Ito et al [107] scheme.

5.5 Concluding Remarks

In this chapter we have considered a secret sharing scheme realizing the general access structure. The share size is a major concern in the design of generalized secret sharing scheme. The share size grows exponentially in many cases. We have proposed a scheme with cumulative arrays and a (n, n) threshold scheme using POB. The POB system has a great potential for secret sharing. The representation is unique and also efficient. In the POB based threshold scheme the share size is small and also the secret generation and reconstruction can be easily done by simple XOR operation. An 8 bit secret can be shared with a share of 7 bit size. The scheme is not ideal but the probability of guessing the share reduces as the size of the secret to be shared increases. For sharing a key of size 64 bits only 56 bits are used. Combining this with the cumulative array scheme is a good choice for secret sharing based on generalized access structure. There is a possibility of 126 shares corresponds to a single byte secret. Thus the probability of guessing the share is 1/126. As the size of the secret grows, this probability reduces. For a k byte secret the probability reduces to $(1/126)^k$.

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.

Binu V P, Sreekumar A : "An Epitome of Multi Secret Sharing Schemes for General Access Structure.", International Journal of Information Processing, 8(2), 13-28, 2014.ISSN : 0973-8215.

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 Γ 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 ($t \leqslant k$). 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 kn 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 $\mathbf{G} =< G, + >$. 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 $\mathcal{T} = \{T_X | X \in \Gamma_0\}$ 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 $X \in \Gamma_0^h$, the Dealer computes

$$T_X^h = K^h - f_h(\sum_{x:P_x \in X} S_x)$$

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

In order ro 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_X^h from the bulletin board and compute $K^h = T_X^h + f_h(V_X)$.

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 with out 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 $f : M \implies G$ 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

$$V_X = g_X^{\prod_{i=1}^{t} S_i} = g_X^{\prod_{x \in X} S_x}$$

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^{rS_1} to P_2. The participant P_t will pass $g_X^{rS_1 \cdots S_t}$ 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

$$V_X = g_X^{\prod_{i=1}^{t} S_i} = g_X^{\prod_{x \in X} S_x}$$

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 α 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 $g_X^{V_X}$ corresponds to the every authorized set X in the initialization step by the Dealer. Every participants $x \in X$ can verify whether $g_X^{V_X} = g_X^{V'_X}$, 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 $C = \sum_{x \in X} g_x^{S_x}$. 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 $g_x^{S_i}$ and broadcasts it to all participants in the set X. Thus every participant can computes C 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 Γ. The participants of any qualified subset γ_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 Γ_0, the Dealer computes public information T_j as follows.

- Compute
  $H_j = K \oplus (U_{j1}^{S_0} \bmod N) \oplus (U_{j2}^{S_0} \bmod N) \oplus, \ldots, \oplus (U_{jd}^{S_0} \bmod N)$.

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

$$f(x) = H_j \times \prod_{k=1}^{d} (X - ID_{jk})/(-ID_{jk}) + \sum_{l=1}^{d} [(P_{jl}^{S_0} \bmod N) \times (X/ID_{jl}) \times \prod_{\substack{k=1 \\ k \neq l}}^{d} (X - ID_{jk})/(ID_{jl} - ID_{jk})] \pmod{Q}$$

  where d is the number of participants in qualified subset γ_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_{ij}, computes and provides $S_{ji}{}' = U_0^{S_{ji}} \pmod N$, where $S_{ji}{}'$ is the pseudo share of P_{ji}. If $S_{ji}^{'h} \pmod N = U_{ji}$, then $S_{ji}^{'}$ is the true shadow else it is false and the participant P_{ji} is the cheater.

- 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 $f(X)$:

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

- 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_0^i and publishes the corresponding h_i, U_{0i}. The Dealer also generate and publishes the information T_{ij} for each qualified subset γ_{ij} in minimal access structure Γ_i. The participants of each qualified subset γ_{ij} in Γ_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 $S_i, \ldots, S_n$ and send them to the participants through a secret channel.

2. For every shared secret $K^{[h]}$ and for every minimal qualified subset $X \in \Gamma_0^{[h]}$, 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 $H^{[h]} = \{(R_X^{[h]}, T_X^{[h]}) | X \in \Gamma_0^{[h]}\}$ 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 $X = \{P1, \ldots, P_t\}$.

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

3. P_t receives $f(R_X^{[h]} + S_i)$, for $1 \leqslant i \leqslant t-1$ 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 $C_{X,i}^{[h]} = f(f(R_X^{[h]} + S_i))$ for every secret K^h, every authorized set X and for every participant P_i. 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 $\mathcal{D}$ 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_α, 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 $S_{ij}, H(S_i), H^2(x_\alpha \parallel i_l \parallel j_m)$.

In the combining phase, the participants of an authorized subset A_{ij} of Γ_{S_i} 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.

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

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 [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.

$$f_q^{S_i}(x) = S_i + d_1^{i_q} x + d_2^{i_q} x^2 + \ldots + d_{mi_q-1}^{i_q} x^{m_{i_q}-1}$$

For each participant $P_b^{i_q} \in A_q^{S_i}$ in Γ_{S_i}, the Dealer compute pseudo share $U_{P_b}^{i_q} = h(x_{P_b^{i_q}}) \parallel i_l \parallel q_m$, 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 $B_{P_b}^{i_q} = f_q^{S_i}(ID_b^{i_q})$. Finally the shift values are computed and published corresponds to each secret and each authorized subset $M_{P_b}^{i_q} = B_{P_b}^{i_q} - U_{P_b}^{i_q}$.

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

$$S_i = \sum_{b \in \{1,2,\ldots,m_{i_q}\}} B_{P_b}^{i_q} \prod_{r \in \{1,2,\ldots,m_{i_q}\} r \neq b} \frac{-ID_{P_r^{i_q}}}{ID_{P_b^{i_q}} - ID_{P_r^{i_q}}}$$

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 send 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.

6.7.1 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 $ps_i = g^{s_i} \pmod{n}$.
- 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 $s0_i$ from $[2, n]$ such that $s0_i$ is relatively prime to $\phi(n)$ and compute $ps0_i = g^{s0_i} \pmod{n}$ corresponds to each secret K_i.
- Find $h0_i$ such that $s0_i \times h0_i \equiv 1 \pmod{\phi(n)}$.

- 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 Γ_i.
- Compute $f_i(1)$ and for each subset $\gamma_{ij} = \{P_{1j}, P_{2j}, \ldots, P_{lj}\}$ compute

$$H_{ij} = f_i(d_{ij}) \bigoplus ps_1^{s0_i} \pmod{n} \bigoplus ps_2^{s0_i} \pmod{n} \bigoplus \ldots \bigoplus ps_l^{s0_i} \pmod{n}$$

- The Dealer then publish

  $ps0_i, h0_i, f_i(1), H_{i1}, H_{i2}, \ldots, H_{it}, d_{i1}, d_{i2}, \ldots, d_{it}$

  corresponds to each secret K_i and the access structure Γ_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 (Γ_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 $x_{kij} = ps0_i^{s_k}, k = 1, \ldots, l$. These values are then delivered to the designated combiner.
- The combiner computes

$$f_i(dij)\prime = H_{ij} \bigoplus x_{1ij} \bigoplus x_{2ij} \bigoplus \ldots \bigoplus x_{lij}$$

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

$$f_i(x) = f_i(1) \times \frac{(x - d_{ij})}{1 - d_{ij}} + f_i(d_{ij})\prime \times \frac{(x-1)}{d_{ij} - 1}$$
$$= \frac{x \times f_i(1) - d_{ij} \times f_i(1) - x \times f(d_{ij})\prime + f(d_{ij})\prime}{1 - d_{ij}}$$

- 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 $x_i^{h0_i} = ps_i \pmod m$. That is $x_i^{h0_i} = (ps0_i)^{s_i})^{h0_i} = (g^{s0_i h0_i})^{s_i} = g^{s_i} = ps_i \pmod m$. 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

Properties	Cachin [40]	Pinch [169]	RJH CCH [101]	Sun [207]	Adhikari [59]	Roy [180]	Proposed Scheme
share size same as secret	Yes	Yes	Yes	Yes	Yes	Yes	Yes
use of one way function	Yes	Yes	No	Yes	Yes	Yes	Yes
use of discrete logarithm	No	Yes	Yes	No	No	No	Yes
use of interpolation	No	No	Yes	No	No	Yes	Yes
shares remain secret during reconstruction	No	Yes	Yes	Yes	Yes	Yes	Yes
dealer knows the share	Yes	Yes	No	Yes	Yes	Yes	No
shares can be reused	No	Yes	Yes	Yes	Yes	Yes	Yes
dynamic	No	Yes	Yes	Yes	Yes	Yes	Yes
verifiability	No	No	Yes	Yes	Yes	Yes	Yes

In the reconstruction phase, each participant P_{ij} in γ_{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

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

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 with out modifying the participants secret shadow. The scheme is also computationally efficient and can be implemented easily.

Chapter 7

Elliptic Curve and Pairing

7.1 Introduction

Over the past few decades Elliptic curve have found useful applications. The curve offers rich and insightful structure, especially those defined over a finite field. These curves are suitable for wide variety of applications in cryptography. There is not much work done in the area of secret sharing, where elliptic curve can be effectively utilized. In this chapter we explore the fundamentals of elliptic curve and then an important construct called Bilinear pairing, which can be effectively utilized to build secret sharing schemes with several extended capabilities. Our aim in this section is to summarize just enough of the basic theory of elliptic curve needed for secret sharing applications. The readers may refer to books and survey articles to learn the theory of elliptic curve in detail [23] [88] [119] [121] [147] [195] [196]. Elliptic curve pairing and their applications are reviewed by Dutta et al [67].

7.2 Elliptic Curves

An elliptic curve is the set of solutions to an equation of the form

$$Y^2 = X^3 + AX + B$$

Equations of this type are called *Weierstrass equations* after the mathematician who studied them extensively. Two elliptic curves E_1 and E_2 defined by the equations

$$E_1 : Y^2 = X^3 - 3X + 3 \qquad \text{and} \qquad E_2 : Y^2 = X^3 - 6X + 3$$

The plot of these curves are given in 7.1 and 7.2.

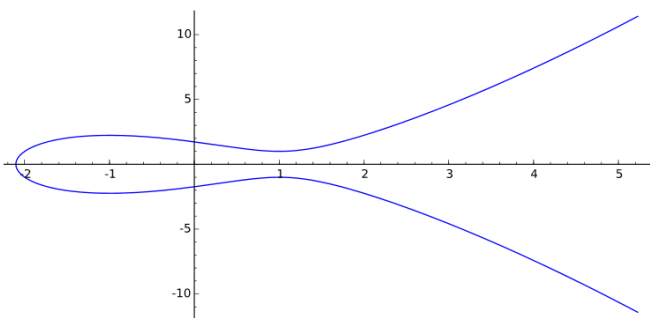

Figure 7.1: Elliptic Curve E1

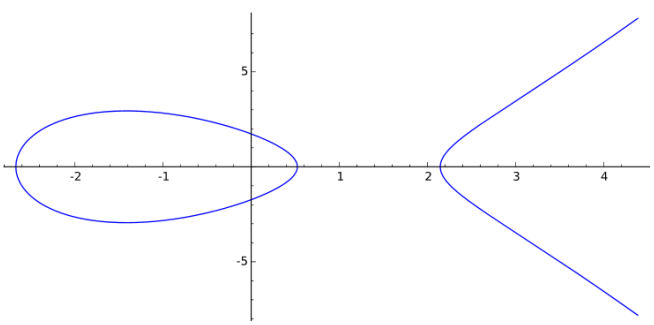

Figure 7.2: Elliptic Curve E2

An amazing feature of elliptic curve is that, we can take two points on an elliptic curve and add them to produce a third point in a natural way. The addition operation is visualized geometrically. If we connect two points P and Q on the curve with a line L, it will intersect the curve at a third point R. The reflection R' of it will be the sum $P+Q$. The Figure 7.3 shows elliptic curve point addition geometrically. If we add the point $P = (a, b)$ to its reflexion about the X-axis $P' = (a, -b)$, the line L through these two points will be a vertical line $x = a$ and this line will intersect the curve at two points. There is no third point of intersection. The solution for this is to consider a point $\mathcal{O}$ that does not exist in the XY-plane, but we assume that it lies on every vertical line. So

$$P + P' = \mathcal{O}$$

.

It is also noted that, if we add the point P to $\mathcal{O}$, we get back to P.i.e.,

$$P + \mathcal{O} = P$$

.

So $\mathcal{O}$ acts like zero for elliptic curve addition, which is the identity element in elliptic curve additive group.

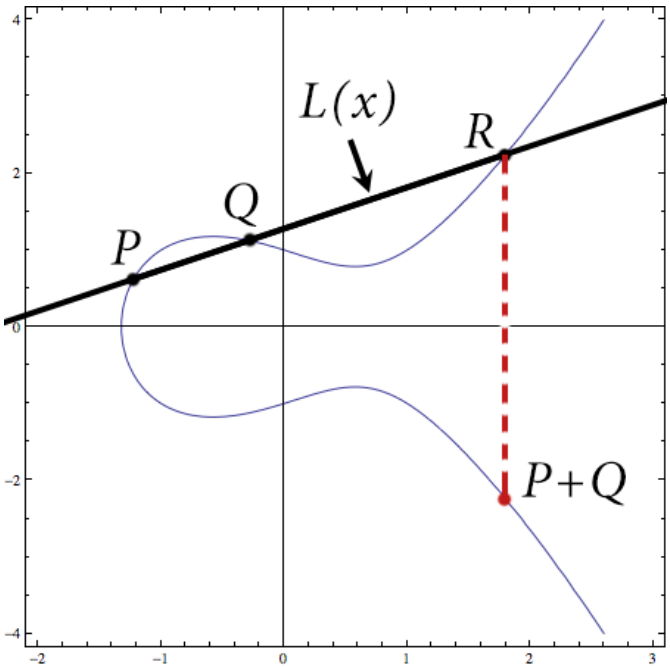

Figure 7.3: Elliptic Curve Point Addition

Definition 7.2.1. An elliptic curve E is the set of solutions to a Weierstrass equation

$$E : Y^2 = X^3 + AX + B$$

with an extra point $\mathcal{O}$. The constant A and B must satisfy

$$4A^3 + 27B^2 \neq 0$$

The *addition law* on E is defined in the following way. Let P and Q be two points on E and let L be the line connecting P and Q. If $P = Q$ then L is a tangent to E at P. The intersection of E and L consist of three points P, Q, R, counted with appropriate multiplicities and the assumption

that $\mathcal{O}$ lies on every vertical line. If $R = (a, b)$, then the sum of P and Q denoted by $P + Q$, is the reflection $R' = (a, -b)$ of R across X-axis.

If $P = (a, b)$, then $-P = (a, -b)$, which is the reflected point. Repeated addition is represented as multiplication of a point by an integer. i.e.;

$$[k]P = \underbrace{P + P + P + \cdots + P}_{\text{k points}}$$

Similarly

$$[-k]P = \underbrace{-P - P - P - \cdots - P}_{\text{k points}}$$

The quantity $\Delta_E = 4A^3 + 27B^2$ is called the *discriminant* of E. When $\Delta_E \neq 0$, the cubic polynomial $X^3 + AX + B$ does not have any repeated roots or we say that the curve E is smooth. i.e., E can be factored as

$$X^3 + AX + B = (X - a_1)(X - a_2)(X - a_3)$$

where a_1, a_2, a_3 are distinct.

Theorem 7.2.1. *Let E be an elliptic curve. Then the addition law on E has the following properties.*

1. *$P + \mathcal{O} = \mathcal{O} + P = P$ for all $P \in E$. (Identity)*
2. *$P + (-P) = \mathcal{O}$ for all $P \in E$. (Inverse)*
3. *$(P + Q) + R = P + (Q + R)$ for all $P, Q, R \in E$. (Associative)*
4. *$P + Q = Q + P$ for all $P, Q \in E$. (Commutative)*

It is noted that, the addition law makes the points of E into an abelian group. The proof of these laws can be found in [131] [195].

We can find explicit formulas for easy addition and subtraction of points on an elliptic curve.

Theorem 7.2.2. *Let E be an elliptic curve*

$$E : Y^2 = X^3 + AX + B$$

and suppose we want to add two distinct points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. Let the line connecting P_1 and P_2 to be

$$L : Y = \lambda X + C$$

The slope and y intercept of the line is given by

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{if } P_1 \neq P_2. \\ \frac{3x_1^2 + A}{2y_1}, & \text{if } P_1 = P_2. \end{cases}$$

$$C = y_1 - \lambda x_1.$$

and let

$$x_3 = \lambda^2 - x_1 - x_2. \qquad \text{and} \qquad y_3 = \lambda(x_1 - x_3) - y_1.$$

Then $P_1 + P_2 = (x_3, y_3)$

- *if $P_1 = \mathcal{O}$, then $P_1 + P_2 = P_2$.*
- *if $P_2 = \mathcal{O}$, then $P_1 + P_2 = P_1$.*
- *if $x_1 = x_2$ and $y_1 = -y_2$, then $P_1 + P_2 = \mathcal{O}$.*

Proof. The line L intersect the curve in three points. Let $P_3 = (x_3, y_3')$ be the third zero of L. Now substitute $Y = \lambda X + C$ into the equation of E to obtain

$$(\lambda X + C)^2 = X^3 + AX + B$$

Expanding this will give

$$f(X) = X^3 - \lambda^2 X^2 + (A - 2C\lambda)X + B - C^2 = 0$$

x_1, x_2, x_3 must be the roots of this equation, since they are the x coordinates of P_1, P_2 and P_3, which satisfy both the equation of the elliptic curve and $L = 0$. Thus $f(X) : (X - x_1)(X - x_2)(X - x_3)$. Comparing the coefficients of the X^2 terms gives $x_1+x_2+x_3 = \lambda^2$. Hence $x_3 = \lambda^2-x_1-x_2$. We can write $y_3' = \lambda x_3 + C$. So $y_3' = \lambda x_3 + y_1 - \lambda x_1$. By taking the inverse of y_3', i.e., $-y_3'$, we will obtain $y_3 = \lambda(x_1 - x_3) - y_1$. □

7.3 Elliptic Curves Over Finite Fields

Elliptic curves whose points have coordinates in a finite field $\mathbb{F}_p$ are the best candidates for cryptography. We can define an elliptic curve over $\mathbb{F}_p$ by the equation

$$E : Y^2 = X^3 + AX + B$$

where $A, B \in \mathbb{F}_p$ with $4A^3 + 27B^2 \neq 0$ and $p \geqslant 3$. We then look for points $(x, y) \in \mathbb{F}_p$ satisfying the elliptic curve equation. i.e.,

$$E(\mathbb{F}_p) = \{(x, y) : x, y \in \mathbb{F}_p \quad \text{which satisfy} \quad y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}\}$$

.

Example 7.3.1. Let us consider an elliptic curve

$$E : Y^2 = X^3 + 3X + 8 \quad \text{over the field} \quad \mathbb{F}_{13}.$$

The points on this curve can be found out by putting in all possible values of X and then check whether Y is quadratic residue or not. Corresponding to each X value, there will be two possible values for Y. So there can be a maximum of $2p + 1$ points on the curve including $\mathcal{O}$. In the example given, $E(\mathbb{F}_{13})$ consist of nine points.

$$E(\mathbb{F}_{13}) = \{\mathcal{O}, (1, 5), (1, 8), (2, 3), (2, 10), (9, 6), (9, 7), (12, 2), (12, 11)\}$$

.

The theory of geometry developed for the field $\mathbb{R}$ can be used for $\mathbb{F}_p$ using the algebraic geometry. If $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are the two points then the sum $R = P + Q = (x_3, y_3)$, can be obtained by applying the formulas mentioned in addition theorem 7.2.2. The division operation can be done by finding the inverse of the number in the field. All operations are done in $\mathbb{F}_p$.

Theorem 7.3.1. *[99] Let E be an elliptic curve over $\mathbb{F}_p$ and let P and Q be points on $E(\mathbb{F}_p)$.*

(a) The elliptic curve addition algorithm applied to P and Q yields a point $R = P + Q$ in $E(\mathbb{F}_p)$.

(b) The addition law on $E(\mathbb{F}_p)$ satisfies all the properties listed in the theorem 7.2.1. It is noted that the addition law makes $E(\mathbb{F}_p)$ into a finite group.

The congruence $X^3 + AX + B \equiv 0 \pmod{p}$ is quadratic residue 50% of the time or else it is a non residue or 0 (happens only once). Hasse's theorem provides a bound on the number of points on an elliptic curve.

Theorem 7.3.2. *(Hasse) Let N be the number of points in an elliptic curve ($\#E(\mathbb{F}_p)$) defined over $\mathbb{F}_p$. Then*

$$N - (p + 1) \leqslant t_p \text{ with } t_p \text{ satisfying } |t_p| \leqslant 2\sqrt{p}$$

The quantity t_p is called the trace of Frobenius for $E(\mathbb{F}_p)$. Hasse's theorem gives a bound for number of points on the elliptic curve. However it will not provide a method for calculating the number of points. The method of substituting each value for X and checking $X^3 + AX + B$ is a square or not against a table take $O(p)$ time, so is very inefficient. An algorithm developed by Schoof [188] can be used to find $\#E(F_p)$ in $O((logp)^6)$ time. Elkies and Atkin improved this algorithm later and is

known as *SEA algorithm* [187] [188]. Satoh [185] developed a reasonably efficient algorithm for counting points on curve in the field of form p^e, where p is a small prime and e is moderately large.

7.4 Elliptic Curve Discrete Logarithm Problem

Elliptic Curve Discrete Logarithm Problem (ECDLP) is computationally harder than the Discrete Logarithm Problem (DLP) in $\mathbb{F}_p^*$. Let $E(\mathbb{F}_p)$ be an elliptic curve defined over $\mathbb{F}_p$. Given two points Q and P, the attacker has to find out n such that $Q = nP$. i.e., the attacker has to find out how many times P must be added to itself in order to get Q.

$$Q = \underbrace{P + P + P + \cdots + P}_{\text{n additions on E}} = nP$$

Definition 7.4.1. Let E be an elliptic curve over the finite field $\mathbb{F}_p$ and let P and Q be the points in the $E(\mathbb{F}_p)$. Then the Elliptic Curve Discrete Logarithm Problem (ECDLP) is the problem of finding an integer n such that $Q = nP$. The integer n is represented as

$$n = log_P(Q)$$

We refer n, the elliptic curve logarithm of Q with respect to P.

There are situations, where n is not defined i.e., Q is not a multiple of P. But in the cryptographic applications, we choose P and compute $Q = nP$. So $n = log_P(Q)$ always exist. If s is the order of P such that $sP = \mathcal{O}$. The value of n will be in $\mathbb{Z}/s\mathbb{Z}$. The ECDL satisfies

$$log_P(Q_1 + Q_2) = log_P(Q_1) + log_P(Q_2)$$

.

7.5 Hardness of ECDLP

The collision algorithms take approximately $\sqrt{N}$ steps in order to find a collision among N objects. But they require creation of one or more lists of size approximately $\sqrt{N}$. The baby step-giant step algorithm is a collision algorithm that is used to solve the discrete logarithm problem for the field $\mathbb{F}_p$ in $\sqrt{p}$ time. The index calculus method solves the DLP in $\mathbb{F}_P$ much more rapidly. But for elliptic curve groups, collision algorithm is the fastest known method. There is no index calculus algorithm known for ECDLP and indeed, there are no general algorithms known that solve ECDLP in less than $O(p)$ steps. This is the reason elliptic curves groups are found useful at present.

In order to solve the ECDLP, the attacker can build two lists of points by randomly choosing integers $a_1, a_2, \ldots, a_r$ and $b_1, b_2, \ldots, b_r$ between 1 and p.

$$List - 1 : a_1P, a_2P, a_3P, \ldots, a_rP$$
$$List - 2 : b_1P + Q, b_2P + Q, b_3P + Q, \ldots, b_rP + Q$$

As soon as a collision or match is found between two lists then ECDLP can be solved. If some $a_iP = b_jP + Q$, then $Q = (a_i - b_j)P$, which provides the solution. If r is larger than $\sqrt{p}$ i.e.; $r \approx 3\sqrt{p}$, then there is a very good chance of collision. The collision algorithm usually requires a lot of storage for the two lists. Pollard's ρ method can be used for storage-free collision algorithm with the same running time.

There are some primes p for which the DLP in $\mathbb{F}_p$ is comparatively easy. For example if $p-1$ is a product of small primes, then the Pohling-Hellman algorithm [170] gives a quick solution to DLP in $\mathbb{F}_p{}^*$. In a similar way, there are some elliptic curve and some primes for which ECDLP in $E(\mathbb{F}_p)$ is comparatively easy. These curves must be avoided while building secure crypto systems.

7.6 Computing nP, Double and Add Algorithm

In cryptographic application, we need to compute nP. If n is large then computing nP by $P, 2P, 3P, 4P, \ldots, nP$ is not practical. The most efficient way to compute nP is similar to computing g^e using *square and multiply algorithm*. Since the operation on elliptic curve involves point addition, we call it as *square and add*. The underlying technique is as follows.

First write n in binary form

$$n = n_0 + n_1.2 + n_2.2^2 + n_3.2^3 + \cdots + n_r.2^r \quad with\ n_0, n_1, \ldots, n_r \in \{0, 1\}$$

Algorithm 7.1: Double and Add Algorithm for Elliptic Curve.

Input: Point $P \in E(\mathbb{F}_p)$ and $n \geqslant 1$

Output: The new point $R = nP \in E(\mathbb{F}_p)$

1 Set $Q = P$ and $R = \mathcal{O}$
2 **while** $n > 0$ **do**
3 if $n \equiv 1(\mod 2)$ set $R = R + Q$
4 set $Q = 2Q$ and $\lfloor n = n/2 \rfloor$
5 **end**
6 Return the point R, which equals nP.

Next we compute

$$Q_0 = P, \quad Q_1 = 2Q_0, \quad Q_2 = 2Q_1, \ldots, Q_r = 2Q_{r-1}$$

Each Q_i is twice the previous Q_{i-1}

$$Q_i = 2^i P$$

Computing each Q_i needs a doubling and a total of r doubling. Finally the computation of nP needs additional r additions also.

$$nP = n_0 Q_0 + n_1 Q_1 + n_2 Q_2 + \cdots + n_r Q_r$$

If we consider a point addition as a point operation in $E(\mathbb{F}_p)$. Then computing nP needs $2r$ point operations in $E(\mathbb{F}_p)$. Since $n \geqslant 2^r$, the computation takes not more than $2log_2(n)$ point operation to compute nP. The double and add algorithm is given in Algorithm 7.1.

7.7 Elliptic Curve Over $\mathbb{F}_{p^k}$

The binary is the most suitable language for computers. Using an elliptic curve modulo 2 is the preferred one. But it is noted that $E(\mathbb{F}_2)$ contains at most five points, so $E(\mathbb{F}_2)$ is not suitable for the security applications. Field containing 2^k elements is a preferred choice. It is noted that for every prime power p^k, there exist a field $\mathbb{F}_{p^k}$ with p^k elements. We can consider an elliptic curve whose Weierstrass equation has coefficients in a field $\mathbb{F}_{p^k}$, and then consider the points having coordinates in $\mathbb{F}_{p^k}$. Hasse's theorem is applicable in this more generalized settings also.

Theorem 7.7.1. *(Hasse). Let E be an elliptic curve over $\mathbb{F}_{p^k}$. Then*

$$\#E(\mathbb{F}_{p^k}) = p^k + 1 - t_{p^k} \quad with \quad t_{p^k} \quad satisfying \quad |t_{p^k}| \leqslant 2p^{k/2}$$

This shows that elliptic curve over $\mathbb{F}_{2^k}$ is a suitable choice for cryptographic application. But the discriminant $\Delta = -16(4A^3 + 27B^2)$ is always zero. The solution is to use a more generalized Weierstrass equations to define the elliptic curve.

Definition 7.7.1. An Elliptic curve E is the set of solutions to a generalized Weierstrass equation

$$E : Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$$

together with the point $\mathcal{O}$. The coefficients $a_1, a_2, \ldots, a_6$ should satisfy $\Delta \neq 0$, to ensure that the curve is non singular. Δ is defined as

$$\Delta = -b_2^2 b_8 - 8b_4^3 - 27b_6^2 + 9b_2 b_4 b_6$$

where

$$\begin{aligned} b_2 &= a_1^2 + 4a_2 \\ b_4 &= 2a_4 + a_1 a_3 \\ b_6 &= a_3^2 + 4a_6 \\ b_8 &= a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2 \end{aligned}$$

The addition law can be applied here also. But the reflection map $(x, y) \to (x, -y)$ is replaced by

$$(x, y) \to (x, -y - a_1 x - a_3)$$

If $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are the two points with $P_1 \neq \pm P_2$. Then the sum $P_3 = (x_3, y_3)$, where

$$x_3 = \lambda^2 + a_1\lambda - a_2 - x_1 - x_2 \quad with \quad \lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

If $P_1 = P_2$ then the x coordinates of P_3 is

$$x_3 = \frac{x^4 - b_4 x^2 - 2b_6 x - b_8}{4x^3 + b_2 x^2 + 4b_4 x + b_6}$$

There are lot of computational advantage when working with elliptic curves defined over $\mathbb{F}_{2^k}$. For security reasons k should be prime. If k is composite then for $j|k$, there exist a subfield $\mathbb{F}_{p^j}$, which can be used to speed up computations by compromising security. The use of an elliptic curve in $\mathbb{F}_2$, with coordinates of the points chosen from $\mathbb{F}_{2^k}$ allows to use Frobenius map instead of doubling map, which provides significant gain in efficiency.

Definition 7.7.2. The Frobenius map τ is the map from the field $\mathbb{F}_{p^k}$ to itself defined by the rule

$$\tau : \mathbb{F}_{p^k} \to \mathbb{F}_{p^k}, \qquad \alpha \to \alpha^p$$

It is noted that the Forbenius map preserves addition and multiplication.

$$\tau(\alpha + \beta) = \tau(\alpha) + \tau(\beta)$$

and

$$\tau(\alpha.\beta) = \tau(\alpha).\tau(\beta)$$

The multiplication rule is straight forward

$$\tau(\alpha.\beta) = (\alpha.\beta)^p = \alpha^p.\beta^p = \tau(\alpha).\tau(\beta)$$

For $p = 2$ the addition law is easy

$$\tau(\alpha + \beta) = (\alpha + \beta)^2 = \alpha^2 + 2.\alpha.\beta + \beta^2 = \alpha^2 + \beta^2 = \tau(\alpha) + \tau(\beta)$$

Let $P = (x, y) \in E(\mathbb{F}_{2^k})$ be a point on E, with coordinates in some larger field $\mathbb{F}_{2^k}$. The Forbenius map is defined by applying τ to each coordinate.

$$\tau(P) = (\tau(x), \tau(y)) \in E(\mathbb{F}_{2^k})$$

If P and Q are the elements of $E(\mathbb{F}_{2^k})$,then

$$\tau(P + Q) = \tau(P) + \tau(Q)$$

This shows that Forbenius map is a group homomorphism of $E(\mathbb{F}_{2^k})$ to itself. The computation of nP mentioned in section 7.6 needs approximately $\log n$ doubling and $\frac{1}{2}\log n$ additions. A refinement which uses negative powers of 2 reduces the time to approximately *logn*

doubling and $\frac{1}{3}\log n$ additions. In both the cases, the number of doubling remains $\log n$. Kolbitz suggested an idea to replace doubling map with the Forbenius map. This lead to large saving in time because the computation of $\tau(P)$ takes comparatively less time than computing $2P$.

7.8 Points of Finite Order on Elliptic Curves

The points of finite order on an elliptic curve are called *torsion points*. They play a major role in forming the elliptic curve groups.

Definition 7.8.1. Let $m \geqslant 1$ be an integer. A point $P \in E$ satisfying $mP = \mathcal{O}$ is called point of order m in the group E. The set of points of order m is denoted by

$$E[m] = \{P \in E : [m]P = \mathcal{O}\}$$

These points are called *points of finite order* or *torsion points.*

It is noted that $E[m]$ forms an additive subgroup of E. If the coordinates of points are chosen from a particular field K. Then we will represent it as $E(K)[m]$. If we add two points P and Q in $E[m]$ then $P + Q$ is also in $E[m]$. Similarly $-P$ is also in $E[m]$. The group of points of order m has a simple structure if the coordinates of the points are chosen from a large field.

Proposition 7.1. Let $m \geqslant 1$ be an integer.

(a) Let E be an elliptic curve over $\mathbb{Q}$ or $\mathbb{R}$ or $\mathbb{C}$. Then

$$E(\mathbb{C})[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$$

is a product of two cyclic group of order m.

(b) Let E be an elliptic curve over $\mathbb{F}_p$, and if p does not divide m. Then there exist a value for k such that

$$E(\mathbb{F}_{p^{jk}})[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}, \quad for\ all \quad j \geqslant 1$$

Remark 7.8.1. If l is prime and K is any field. Then

$$E(K)[l] = \mathbb{Z}/l\mathbb{Z} \times \mathbb{Z}/l\mathbb{Z}$$

even if m is not prime

$$E(K)[m] = \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$$

We have to consider $E[m]$ as a 2-dimensional vector space over the field having basis P_1, P_2. Every point P in $E[m]$ can be represented as a linear combination of P_1 and P_2. i.e., $P = aP_1 + bP_2$, for a unique choice of coefficients $a, b \in \mathbb{Z}/m\mathbb{Z}$. Finding a and b given P_1, P_2 and P is as hard as ECDLP.

7.9 Rational Functions and Divisors on Elliptic Curves

Rational functions on an elliptic curve is related to its zeros and poles. Consider a rational function of a single variable. A rational function is a ratio of polynomials.

$$f(X) = \frac{a_0 + a_1X + a_2X^2 + \cdots + a_nX^n}{b_0 + b_1X + b_2X^2 + \cdots + b_nX^n}$$

The polynomial can be factored completely if we allow complex numbers. So the rational function $f(X)$ can be factored as

$$f(X) = \frac{a(X-\alpha_1)^{e_1}(X-\alpha_2)^{e_2}\cdots(X-\alpha_r)^{e_r}}{b(X-\beta_1)^{d_1}(X-\beta_2)^{d_2}\cdots(X-\beta_s)^{d_s}}$$

The numbers $\alpha_1, \alpha_2, \ldots, \alpha_r$ are called *zeros* of $f(X)$ and the numbers $\beta_1, \beta_2, \ldots, \beta_s$ are called the *poles* of $f(X)$. The exponents $e_1, e_2, \ldots, e_r$ are the associated multiplicities of zeros and the exponents $d_1, d_2, \ldots, d_s$ are the associated multiplicities of poles. We can keep track of the zeros and poles of $f(X)$ and their multiplicities by defining the *divisor* of $f(X)$, which is the formal sum.

$$div(f(X)) = e_1[\alpha_1] + e_2[\alpha_2] + \cdots + e_r[\alpha_r] - d_1[\beta_1] - d_2[\beta_2] - \cdots - d_r[\beta_r]$$

In a similar fashion, if E is an elliptic curve

$$E : Y^2 = X^3 + AX + B$$

and if $f(X, Y)$ is a rational function of two variables, then there are points on E where the numerator of f vanishes and also there are points where the denominator of f vanishes. That is f has zeros and poles on E. We can assign multiplicities to the zeros and poles, so f has a divisor.

$$div(f) = \sum_{P \in E} n_p[P]$$

The coefficients n_P are integers and only finitely many of the n_P are non zero, so $div(f)$ is a finite sum.

Example 7.9.1. Suppose E defined by the cubic equation factors as

$$X^3 + AX + B = (X - \alpha_1)(X - \alpha_2)(X - \alpha_3)$$

Then the points $P_1 = (\alpha_1, 0), P_2 = (\alpha_2, 0)$ and $P_3 = (\alpha_3, 0)$ are points of order 2. i.e., $2P_1 = 2P_2 = 2P_3 = \mathcal{O}$. The divisor of Y is equal to

$$div(Y) = [P_1] + [P_2] + [P_3] - 3[\mathcal{O}]$$

In general , we can define *divisor* on E to be any formal sum

$$D = \sum_{P \in E} n_P[P] \quad with\ n_P \in \mathbb{Z}\ and\ n_P = 0 \text{ for all but finetely many } P$$

The *degree of a divisor* is the sum of its coefficients

$$Deg(D) = \sum_{P \in E} n_P$$

The sum of the divisor is

$$Sum(D) = \sum_{P \in E} n_P P$$

The following theorem says which divisors are divisors of functions and to what extent the divisor of a function determines the function.

Theorem 7.9.1. *[99] Let E be an elliptic curve*

1. *Let f and f' be rational functions on E. If div(f)=div(f'), then there is a non zero constant c such that $f = cf'$*
2. *Let $D = \sum_{P \in E} n_P[P]$ be a divisor on E. Then D is the divisor of a rational function on E, if and only if*

$$Deg(D) = 0 \quad and \quad Sum(D) = \mathcal{O}$$

In particular, if a rational function on E has no zeros or no poles, then it is a constant.

It is noted that, if $P \in E[m]$ is a point of order m. Then $m[P] = \mathcal{O}$, so the divisor

$$m[P] - m[\mathcal{O}]$$

satisfies the conditions of the above theorem. Hence there is a rational function $f_P(X, Y)$ on E satisfying

$$div(f_P) = m[P] - m[\mathcal{O}]$$

Example 7.9.2. Consider the case when $m = 2$. The points of order 2 has Y coordinate 0. If $P = (\alpha, 0) \in E[2]$, then the function $f_P = X - \alpha$ satisfies

$$div(X - \alpha) = 2[P] - 2[\mathcal{O}]$$

7.10 Bilinear Pairing on Elliptic Curve

Bilinear Pairing on elliptic curve is an important construct which provides solution to several security problems. There are lot of examples of bilinear pairing in linear algebra. The dot product is a bilinear pairing on the vector space $\mathbb{R}^n$. The pairing takes two vectors v and w from $\mathbb{R}^n$ and return a number.

$$\beta(v, w) = v.w = v_1w_1 + v_2w_2 + \cdots + v_nw_n$$

It is linear in the sense that for any vectors v_1, v_2, w_1, w_2 and any real numbers a_1, a_2, b_1, b_2, it satisfies the relation

$$\beta(a_1v_1 + a_2v_2, w) = a_1\beta(v_1, w) + a_2\beta(v_2, w)$$
$$\beta(v, b_1w_1 + b_2w_2) = b_1\beta(v, w_1) + b_2\beta(v, w_2)$$

Another bilinear pairing is the determinant map on $\mathbb{R}^2$. Thus if $v = (v_1, v_2)$ and $w = (w_1, w_2)$, then

$$\delta(v, w) = det \begin{pmatrix} v_1 & v_2 \\ w_1 & w_2 \end{pmatrix} = v_1w_2 - v_2w_1$$

is a bilinear map. The determinant map is alternating, which means that, if we switch the vectors, the value changes sign. This also implies that $\delta(v, v) = 0$.

$$\delta(v, w) = -\delta(w, v)$$

The bilinear pairing on elliptic curve is similar to this. They take two points on an elliptic curve as input and gives as output a number. It satisfies the bilinear property also

$$\beta(a_1P_1 + a_2P_2, Q) = a_1\beta(P_1, Q).a_2\beta(P_2, Q)$$
$$\beta(Q, b_1P_1 + b_2P_2) = b_1\beta(Q, P_1).b_2\beta(Q, P_2)$$

where P_1, P_2 and Q are the points on elliptic curve and a_1, a_2, b_1, b_2 are the elements of the field selected.

Bilinear pairing on elliptic curve have number of important cryptographic applications in practice. Most of these applications need finite fields of prime power order $\mathbb{F}_{p^k}$.

7.11 The Weil Pairing

The Weil pairing (e_m) takes as input a pair of points $P, Q \in E[m]$ and gives as output an m^{th} root of unity. i.e., $e_m(P, Q)^m = 1$. The bilinearity of the Weil pairing is represented by the equations

$$e_m(P_1 + P_2, Q) = e_m(P_1, Q)e_m(P_2, Q)$$
$$e_m(P, Q_1 + Q_2) = e_m(P, Q_1)e_m(P, Q_2)$$

Definition 7.11.1. Let P, Q be points of order m in E i.e., $P, Q \in E[m]$. Let f_P and f_Q be rational functions on E satisfying

$$div(f_P) = m[P] - m[\mathcal{O}] \quad and \quad div(f_Q) = m[Q] - m[\mathcal{O}]$$

The Weil Pairing of P and Q is the quantity

$$e_m(P, Q) = \frac{f_P(Q + S)}{f_P(S)} \Big/ \frac{f_Q(P - S)}{f_Q(-S)}$$

where S is any point in E and $S \notin \{\mathcal{O}, P, -Q, P-Q\}$. This ensures that $e_m(P,Q)$ is always defined and is non zero. The value of $e_m(P,Q)$ does not depend on the choice of f_P, f_Q and S

The Weil pairing has many useful properties and are useful for number of cryptographic applications.

Theorem 7.11.2.

1. *The Weil pairing will return a value which is the m^{th} root of unity*
$$e_m(P,Q)^m = 1 \quad for\ all\ P, Q \in E[m]$$
2. *The Weil pairing will satisfy the bilinear property. i.e., for all $P, P_1, P_2, Q, Q_1, Q_2 \in E[m]$*
$$e_m(P_1 + P_2, Q) = e_m(P_1, Q)e_m(P_2, Q)$$
$$e_m(P, Q_1 + Q_2) = e_m(P, Q_1)e_m(P, Q_2)$$
3. *The Weil pairing is alternating, which means that*
$$e_m(P,Q) = e_m(Q,P)^{-1}\ and\ e_m(P,P) = 1\ for\ all\ P, Q \in E[m]$$
4. *The Weil pairing is non degenerate, which means that*
$$e_m(P,Q) = 1 \quad for\ all\ Q \in E[m],\ then\ P = \mathcal{O}$$

If we allow coordinates of points in a sufficiently large field, then $E[m]$ is like a 2-dimensional vector space over the field $\mathbb{Z}/m\mathbb{Z}$. If we choose $P_1, P_2 \in E[m]$ be the basis, then any point P can be written as a linear combination of these basis

$$P = a_P P_1 + b_P P_2 \quad for\ unique\ a_P, b_P \in \mathbb{Z}/m\mathbb{Z}$$

The glory of Weil pairing is that it can be computed very efficiently without expressing P and Q in term of the basis for $E[m]$. Expressing a point in terms of the basis is complicated than solving ECDLP.

7.12 Miller Algorithm to Compute Weil Pairing

Victor Miller [148] developed an algorithm using double-and-add method to efficiently compute the Weil Pairing. The key idea is to rapidly evaluate certain functions with specified divisors.

Theorem 7.12.1. *Let E be an elliptic curve and $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ are the non zero points on the curve.*

(a) Let λ be the slope of the line connecting P and Q. If $P = Q$, it is the slope of the tangent at P. If the line is vertical $\lambda = \infty$. A function $g_{P,Q}$ on E is defined as follows:

$$g_{P,Q} = \begin{cases} \frac{y - y_p - \lambda(x - x_P)}{x + x_P + x_Q - \lambda^2} & if\ \lambda \neq \infty, \\ x - x_P & if\ \lambda = \infty. \end{cases}$$

Then

$$div(g_{P,Q}) = [P] + [Q] - [P + Q] - [\mathcal{O}]$$

(b) Miller's Algorithm.
Let $m \geqslant 1$ and write the binary expansion of m as

$$m = m_0 + m_1 2 + m_2 2^2 + \ldots + m_{n-1} 2^{n-1}$$

with $m_i \in \{0, 1\}$ and $m_{n-1} \neq 0$. The algorithm returns a function f_P whose divisor satisfies

$$div(f_P) = m[P] - [mP] - (m - 1)[\mathcal{O}]$$

Algorithm 7.2: Miller's Algorithm

Input: *An Elliptic Curve point P of order m*

Output: *A function f_P with $div(f_P) = m[P] - m[\mathcal{O}]$*

Set $T = P$ and $f = 1$
for $i = n - 2 : 0$ **do**
    Set $f = f^2.g_{T,T}$
    Set $T = 2T$
    if $m_i = 1$ **then**
        set $f = f.g_{T,P}$
        set $T = T + P$
    end
end
Return the value f

Proof. a) Let $y = \lambda x + v$ be the line through P and Q or the tangent line at P if $P = Q$. The line intersect E at the three points P, Q and $-(P + Q)$, so

$$div(y - \lambda x - v) = [P] + [Q] + [-P - Q] - 3[\mathcal{O}]$$

vertical lines intersect E at points and their negatives, so

$$div(x - x_{P+Q}) = [P + Q] + [-P - Q] - 2[\mathcal{O}]$$

It is noted that

$$g_{P,Q} = \frac{y - \lambda x - v}{x - x_{P+Q}} \tag{7.1}$$

has the divisor

$$[P] + [Q] - [P + Q] - [\mathcal{O}]$$

According to the Addition theorem $x_{P+Q} = \lambda^2 - x_P - x_Q$. Let $y_P = \lambda x_P + v$ so $v = y_P - \lambda x_P$. Replacing the values of v and x_{P+Q} in

equation 7.1 will result

$$g_{P,Q} = \frac{y - \lambda x - y_P + \lambda x_P}{x - \lambda^2 + x_P + x_Q}$$

$$g_{P,Q} = \frac{y - y_P - \lambda(x - x_P)}{x + x_P + x_Q - \lambda^2}$$

If $\lambda = \infty$, then $P + Q = \mathcal{O}$, so g_{P+Q} have divisor $[P] + [-P] - 2[\mathcal{O}]$.

b) The Millers algorithm is similar to double-and-add algorithm. The function $g_{T,T}$ in step 3 and $g_{T,P}$ in step 6 have divisors

$div(g_{T,T}) = 2[T] - [2T] - [\mathcal{O}]$ and $div(g_{T,P}) = [T] + [P] - [T + P] - [\mathcal{O}]$

Using induction on this relation, it can be proved that f_P is a function with divisor $m[P] - m[\mathcal{O}]$

□

Let $P \in E[m]$, then the Miller algorithm can be used to compute a function f_P with divisor $m[P] - m[\mathcal{O}]$. If R is any point on E, then we can compute $f_P(R)$ by evaluating the functions $g_{T,T}(R)$ and $g_{T,P}(R)$ during the execution of the algorithm. It is noted that for computing Weil pairing, we have to evaluate the function at each of the specified point in the given formula

$$e_m(P, Q) = \frac{f_P(Q + S)}{f_P(S)} \Big/ \frac{f_Q(P - S)}{f_Q(-S)}$$

One can compute $f_P(Q + S)$ and $f_P(S)$ simultaneously for efficiency, and similarly for $f_Q(P - S)$ and $f_Q(-S)$. Further savings in computations are available using the Tate pairing, which is a variant of the Weil pairing that we discuss next.

7.13 The Tate Pairing

The Weil pairing on elliptic curve is defined over any field. For elliptic curves over finite fields there is another efficient pairing is defined called Tate pairing. It is computationally more efficient than Weil pairing.

Definition 7.13.1. Let E be an elliptic curve over $\mathbb{F}_q$ and let l be a prime. If P and Q are the two points on $E(\mathbb{F}_q)$ such that $P \in E(\mathbb{F}_q)[l]$ and $Q \in E(\mathbb{F}_q)$. Choose a rational function f_P on E with

$$div(f_P) = l[P] - l[\mathcal{O}]$$

Then the Tate pairing of P and Q is the quantity

$$\tau(P, Q) = \frac{f_P(Q+S)}{f_P(S)} \in \mathbb{F}_q^*$$

where S is any point in $E(\mathbb{F}_q)$ such that $f_P(Q+S)$ and $f_P(S)$ are defined and is non zero. If $q \equiv 1 \pmod{l}$, then the modified Tate pairing of P and Q to be

$$\hat{\tau}(P, Q) = \tau(P, Q)^{(q-1/l)} = \left(\frac{f_P(Q+S)}{f_P(S)}\right)^{(q-1/l)} \in \mathbb{F}_q^*$$

Theorem 7.13.2. *Let E be an elliptic curve over $\mathbb{F}_q$ and l be a prime such that*

$$q \equiv 1 \pmod{l} \quad \textit{and} \quad E(\mathbb{F}_q)[l] \cong \mathbb{Z}/l\mathbb{Z}$$

Then the modified Tate pairing gives a well-defined map

$$\hat{\tau} : E(\mathbb{F}_q)[l] \times E(\mathbb{F}_q)[l] \to \mathbb{F}_q^*$$

The Tate pairing satisfies the following properties:
Bilinearity

$$\hat{\tau}(P_1 + P_2, Q) = \hat{\tau}(P_1, Q)\hat{\tau}(P_2, Q) \textit{ and}$$
$$\hat{\tau}(P, Q_1 + Q_2) = \hat{\tau}(P, Q_1)\hat{\tau}(P, Q_2)$$

Nondegeneracy

$\hat{\tau}(P, P)$ is a primitive l^{th} root of unity for all nonzero $P \in E(\mathbb{F}_q)[l]$

(if x is the primitive l^{th} root of unity, then $x^l = 1$)

Miller's algorithm can be used to compute the function f_P and the Tate pairing efficiently. An efficient implementation of Tate pairing is given in [76].

7.14 MOV Algorithm

The Menezes, Okamoto and Vanstone (MOV) algorithm [145] reduces the ECDLP in $E(\mathbb{F}_p)$ to DLP problem in $\mathbb{F}^*_{p^k}$. Let E be an elliptic curve over $\mathbb{F}_p$, and let $m \geqslant 1$ be an integer such that $p \nmid m$. The curve has m^2 points of order m, but their coordinates may lie in a larger field. We can define the term embedding degree as follows.

Definition 7.14.1. Let E be an elliptic curve over $\mathbb{F}_p$ and let $m \geqslant 1$ be an integer with $p \nmid m$. The embedding degree of E with respect to m is the smallest value of k such that

$$E(\mathbb{F}_{p^k})[m] \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$$

If m is a large prime, then the embedding degree have following characterization, which is suitable for cryptographic applications.

Proposition 7.2. Let E be an elliptic curve over $\mathbb{F}_p$ and let $l \neq p$ be a prime. If $E(\mathbb{F}_p)$ contains a point of order l, then the embedding degree of E with respect to l is given by:

(i) The embedding degree of E is 1.(This cannot happen if $l > \sqrt{p} + 1$).

(ii) If $p \equiv 1 \pmod{l}$, then the embedding degree is l.

(iii) If $p \not\equiv 1 \pmod{l}$, then the embedding degree is the smallest value of $k \geqslant 2$ such that

$$p^k \equiv 1 \pmod{l}$$

The significance of the embedding degree k is that, Weil pairing can be used to embed ECDLP in $E(\mathbb{F}_p)$ into the DLP in the field $\mathbb{F}_{p^k}$.

Let E be an elliptic curve over $\mathbb{F}_p$. If $l > \sqrt{p} + 1$ be a large prime. Let k be the embedding degree and the DLP in $\mathbb{F}_{p^k}^*$ is solvable. Then if $P, Q \in E(\mathbb{F}_p)$ such that $Q = nP$, the MOV algorithm can be used to solve ECDLP and find n.

Algorithm 7.3: MOV Algorithm

Input: Elliptic Curve points P and Q such that $Q = nP$.
Output: The sultion of ECDLP i.e., n.

1 Compute the number of points $N = \#E(\mathbb{F}_{p^k})$.
2 Choose a random point $T \in E(\mathbb{F}_{p^k})$ and $T \notin E(\mathbb{F}_p)$.
3 Compute $T' = (N/l)T$. If $T' = \mathcal{O}$, go back to step 2, else T' is a point of order l and proceed to the next step 4.
4 Compute the Weil pairing values

$$\alpha = e_l(P, T') \in \mathbb{F}_{p^k}^* \text{ and } \beta = e_l(Q, T') \in \mathbb{F}_{p^k}^*$$

Solve the DLP for α and β in $\mathbb{F}_{p^k}^*$, i.e., find an exponent n such that $\beta = \alpha^n$
5 Since $Q = nP$, the ECDLP is also solved.
6 Return n.

Remark 7.14.1. There exist polynomial time algorithm to compute the number of points, if k is not so large. The Weil pairing computation in step 4 can be done quite efficiency using Miller's algorithm in time proportional to $log(p^k)$. The DLP can be solved using the index calculus method which is a sub exponential algorithm and is considerably faster than the collision algorithms such as Pollard's ρ method.

The point T' constructed is independent of P and they form a basis of

$$E[l] \cong \mathbb{Z}/l\mathbb{Z} \times \mathbb{Z}/l\mathbb{Z}$$

$e_l(P, T')$ is the non trivial l^{th} root of unity in $\mathbb{F}^*_{p^k}$. The linearity of Weil pairing implies that

$$e_l(P, T') = e_l(nP, T') = e_l(P, T')^n = e_l(Q, T')$$

So n solves the ECDLP for P and Q.

The practicality of MOV algorithm depends on the size of k. If k is large, say $k > (\ln p)^2$, then the MOV algorithm is completely infeasible. However there are certain special curves whose embedding degree is small. An important class of such curves satisfying the property that

$$\#E(\mathbb{F}_p) = p + 1$$

These curves are called *super singular curves* [75]. They have the embedding degree $k = 2$ and in any case $k \leqslant 6$.

For example the curve $E : y^2 = x^3 + x$ is super singular for any prime $p \equiv 3 \pmod 4$ and it has an embedding degree 2 for any $l > \sqrt{p}+1$. Solving ECDLP in $E(\mathbb{F}_p)$ is no harder than solving DLP in $\mathbb{F}^*_{p^2}$. This means that, it is a poor choice for the applications in cryptography.

There exist another class of elliptic curves over $\mathbb{F}_p$ called *anomalous*. They have the property $\#E(\mathbb{F}_p) = p$. There exist fast linear time algorithm to solve ECDLP on these curves [184]. So the use of these curves must also be avoided.

The ECDLP is also easy in elliptic curves defined over $\mathbb{F}_{2^m}$, when m is composite. The idea is to transfer the ECDLP in $\mathbb{F}_{2^m}$ to an hyperelliptic curve over a smaller field $\mathbb{F}_{2^k}$, where k divides m.

7.15 Modified Weil Pairing and Distortion Maps

In cryptographic application, we want to evaluate $e_m(P, P)$ or $e_m(aP, bP)$. But the Weil pairing is alternating, which means that $e_m(P, P) = 1$, for all P. So the direct use of Weil pairing is not useful. Let $P_1 = aP$ and $P_2 = bP$.

$$e_m(P_1, P_2) = e_m(aP, bP) = e_m(P, P)^{ab} = 1^{ab} = 1$$

One way to get around this is to use an elliptic curve that has a map $\phi : E \longrightarrow E$, with the property that P and $\phi(P)$ are independent in $E[m]$. Hence we can evaluate

$$e_m(P_1, P_2) = e_m(P_1, \phi(P_2)) = e_m(aP, b\phi(P)) = e_m(P, \phi(P))^{ab}$$

For cryptographic application, we choose m to be prime.

Definition 7.15.1. Let E be an elliptic curve and $l \geqslant 3$ be prime. Let $P \in E[l]$ be a point of order l and let $\phi : E \to E$ be a map from E to itself. Then the map ϕ is called l distortion map, if it has the following properties.

(i) $\phi(nP) = n\phi(P)$ for all $n \geqslant 1$.

(ii) The pairing $e_l(P, \phi(P))$ is a primitive l^{th} root of unity. i.e.,

$$e_l(P, \phi(P))^r = 1 \quad \text{if and only if} \quad l|r.$$

The modified Weil pairing is defined in the following way.

Definition 7.15.2. Let E be an elliptic curve. Let $P \in E[l]$ and let ϕ be an l distortion map for P. The *modified Weil pairing* $\hat{e}_l$ on $E[l]$ is defined by

$$\hat{e}_l(Q, Q') = e_l(Q, \phi(Q'))$$

For cryptographic applications, the Weil pairing is evaluated at points that are multiple of P. The important property of modified Weil pairing is its non degeneracy. If Q and Q' are the multiples of P, then

$$\hat{e}_l(Q, Q') = 1 \quad \text{if and only if} \quad Q = \mathcal{O} \; or \; Q' = \mathcal{O}$$

Example 7.15.1. Lets choose an elliptic curve $E : y^2 = x^3 + x$ over the field $\mathbb{F}_p$ with $p \equiv 3 \pmod 4$.
Let $\alpha \in \mathbb{F}_{p^2}$ satisfying $\alpha^2 = -1$. The map is defined by

$$\phi(x, y) = (-x, \alpha y) \quad and \quad \phi(\mathcal{O}) = \mathcal{O}$$

Let $l \geqslant 3$ be a prime and there exist a non zero point $P \in E(\mathbb{F}_p)[l]$. Then ϕ is a l distortion map for P

$$\hat{e}_l(P, P) = e_l(P, \phi(P))$$

is the primitive l^{th} root of unity. Since $\phi(P) \; in \; E(\mathbb{F}_{p^2})$, it is a self map.
The map ϕ respect the addition law on E.

$$\phi(P_1 + P_2) = \phi(P_1) + \phi(P_2) \; for \; all \; P_1, P_2 \in E(\mathbb{F}_{p^2})$$

In particular $\phi(nP) = n\phi(P)$ for all $n \geqslant 1$.

7.16 Concluding Remarks

In this section we explored just enough theory of elliptic curve and pairing. These are the building blocks of several secret sharing constructions based on elliptic curve. The secret sharing schemes with enhanced capabilities can be build using elliptic curve and bilinear pairing. Several advantages are also achieved by the use of elliptic curves for building secret sharing techniques. Share verification, cheater identification and cheater detection are the major achievements with less

computational complexity. Cheater detection and identification can be easily achieved with pairing based techniques. The hardness of ECDLP helps in maintaining the security of shares when building multi secret sharing techniques. The security it offers is comparatively high.

Chapter 8

Generalized Multi-secret Sharing based on Elliptic Curve and Pairing

8.1 Introduction

Use of elliptic curve and pairing in secret sharing is gaining more importance. The use of elliptic curve helps to improve the security and also the computational complexity is reduced. In this chapter we propose a multi secret sharing scheme with monotone generalized access structure. The scheme makes use of Shamir's scheme and Elliptic Curve pairing for the implementation. The shares are chosen by the participant itself, so the consistency of the shares are ensured in this scheme. The participant shares remain secret during the reconstruction phase and this provides

Some results of this chapter are included in the following paper.
Binu V. P., Sreekumar A.,"Secure and Efficient Secret Sharing Scheme with General Access Structures based on Elliptic Curve and Pairing", Wireless Personal Communications-Springer, ISSN: 0929-6212.DOI 10.1007/s11277-016-3619-8.

multi use facility where same share can be used for the reconstruction of multiple secrets. The shared secret, access structure or the participant set can be modified without updating the secret shadow of each participant. This provides dynamism and adds more flexibility to the scheme. The combiner can also verify the shares of the other participants during the reconstruction phase in order to identify the cheaters. The cheating detection and cheater identification is done by using bilinear pairing. This scheme is simple and easy to implement compared with other generalized multi secret sharing scheme with extended capabilities using pairing.The important properties of this proposed scheme are

- Generalized access structure.
- Multi secret sharing.
- Multi use, where each participant has to keep only a single share and can be reused.
- Dynamic, participant or access structure can be modified.
- No secure channel is required.
- Cheater Identification facility.
- Consistency of the shares can be verified.

The use of elliptic curve and pairing have found applications in secret sharing schemes very recently. Several schemes based on threshold and generalized secret sharing is proposed and they have found useful applications. Pairing can be used to introduce verifiability and cheating detection in secret sharing scheme with more security. Chen Wei et al [218] in 2007 proposed a dynamic threshold secret sharing scheme based on bilinear maps. A threshold multi secret sharing scheme based on elliptic curve discrete logarithm is proposed by Runhua Shi et al [194] in

2007. Sharing multiple secrets which are represented as points on elliptic curve using self pairing [133] is proposed by Liu et al [137] in 2008. Wang et al [216] proposed a verifiable threshold multi secret sharing scheme in 2009. In Wang's et al scheme, the number of secrets must be less than or equal to the threshold and also more public values must be changed when the secret need to be updated. Eslami et al [69] in 2010 proposed a modified scheme which avoids these problems. Several publicly verifiable secret sharing schemes are proposed based on pairing. But most of them are single secret sharing schemes [212] [220] [223]. An efficient One Stage Multi Secret Sharing(OSMSS) is proposed recently in 2014 by Fatemi et al [70]. Generalized secret sharing with monotone access structure is also proposed using Elliptic Curve and Bilinear Pairing with capability to detect cheating [100] [226].

8.2 Pairing and Secret Sharing

Pairing on elliptic curves have a number of important cryptographic applications. While first used for cryptanalysis, pairings have since been used to construct many cryptographic systems for which no other efficient implementation is known, such as identity based encryption, attribute based encryption [67] etc. The mapping allows development of new cryptographic schemes based on the reduction of one problem in one group to a different, usually easier problem in the other group. The first group is usually called GAP Diffie-Hellman Group, where the Decisional Diffie Hellman problem (DDHP) [29] is easy. But the Computational Diffie Hellman (CDHP) problem remains hard.

Let G be a cyclic additive group generated by P whose order is prime q. For all $a, b, c \in \mathbb{Z}_q^*$. The CDHP is, given P, aP, bP, compute abP. DDHP is defined as, given P, aP, bP, cP, decide whether $c = ab$ in $\mathbb{Z}_q^*$.

The important pairing based construct is the bilinear pairing. A mapping from $(G_1 =< P >, +)$ to $(G2, .)$, two groups of the same prime order q and the DLP is hard in both the groups is called Bilinear Maps, if the following conditions are satisfied.

1. **Bilinearity**: $\forall P, Q \in G_1, \forall a, b \in Z_q^*$

$$e(aP, bQ) = e(P, Q)^{ab}$$

2. **Non-Degeneracy**: If everything maps to the identity, that's obviously not interesting

$$\forall P \in G_1, P \neq 0 \Rightarrow < e(P, P) >= G_2 \ (e(P, P) \text{ generates } G_2)$$

   In other words:

$$P \neq 0 \Rightarrow e(P, P) \neq 1$$

3. **Computability**: e is efficiently computable. i.e., there is a polynomial time algorithm to compute $e(P, Q) \in G_2$, for all $P, Q \in G_1$.

We can find G_1 and G_2 where these properties hold. The Weil and Tate pairings prove the existence of such constructions [7] [30]. These pairing have found numerous cryptographic applications [115]. Typically, G_1 is an elliptic curve group and G_2 is a finite field.

8.3 Proposed Secret Sharing Scheme

The proposed scheme makes use of Shamir's scheme and also Elliptic Curve Pairing for the implementation. The scheme can be used to share multiple secrets with out changing the participant share. The process of sharing a single secret is mentioned below. The scheme can be extended

to share multiple secrets $K_1, K_2, \ldots, K_m$. Let $P_1, P_2, \ldots, P_m$ be the set of participants involved. A monotone access structure with minimal qualified set $\mathcal{A}^0 = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_t\}$ is considered. The scheme also uses a public notice board where every user have the access, but only the Dealer can write or modify the data. The participant select their shares and are kept secret. The problem with Dealer sending inconsistent shares can thus be avoided. The scheme is also multi use i.e., the same share can be used for sharing several secret. The dynamic nature of the scheme allows the participants set or the access structure to be changed without changing the existing participant's share. This adds more flexibility to the scheme. The use of Elliptic Curve makes the scheme more robust and secure.

The Secret Sharing Scheme consist of four important phases.

1. Initialization.
2. Share generation.
3. Secret Distribution.
4. Verification and Secret Reconstruction.

8.3.1 Initialization

The initialization phase need to be executed only once for a particular secret sharing scheme. It is assumed that the Dealer is a trusted authority and there are n authenticated participants $P_1, P_2, \ldots, P_n$. In the initialization phase some public parameters are posted on the public bulletin called notice board which can be accessed by every participant.

1. The Dealer($\mathbf{D}$) chooses an elliptic curve E over $GF(q)$, where $q = p^r$ and p is a large prime such that DLP and ECDLP in GF(q) is hard. Let G_1 and G_2 be two cyclic group of order q for some large prime p. G_1 is an additive group of points of an elliptic curve over $\mathbb{F}_p$ and G_2

is multiplicative sub group of an extension of finite field $\mathbb{F}^*_{p^2}$. Elliptic curve pairing can be used to map elements from group G_1 to G_2. Modified Weil pairing (ê) is used for the implementation.

2. **D** chooses a generator G of G_1 and also defines a hash function H which maps $H : G_1 \longmapsto \{0,1\}^l$, where l is the bit length of the field.

3. **D** then publish $\{E, G_1, G_2, q, \hat{e}, G, H\}$ in the notice board.

8.3.2 Share Generation

In this phase shares of secrets are generated. The shares are not generated by the Dealer instead they are selected by the participants and send to the Dealer. Dealer will verify the shares and assign the shares corresponds to each participants and publish them in the public notice board.

1. Each participant P_i select a random number $X_i \in \mathbb{Z}^*_q$ and compute $Y_i = X_i G$. The participant will keep X_i secret and send Y_i to the Dealer.

   The Dealer needs to ensure that these Y_i's are distinct to make sure that each participant is using different shares. If $Y_i = Y_j$ for some $P_i \neq P_j$, then the Dealer will ask for new share. It is noted that only the pseudo shares are send to the Dealer. An intruder or the Dealer cannot obtain any information about the secret share because ECDLP in the field is hard to solve.

2. The Dealer then publish the pseudo shares (p_i, Y_i) corresponds to each participants, where $p_i \in \mathbb{Z}_q{}^*$ is the public identity corresponds to each participant $P_1, P_2, \ldots, P_n$ chosen randomly.

8.3.3 Secret Distribution

In this phase the Dealer will share the secret by publishing informations on a public notice board corresponds to the secret to be shared. Each participant can make use of these public values. These public information together with the secret share of each participant can be used for the retrieval of shared secret. Sharing of a single secret value is mentioned here. The same steps can be followed to share multiple secrets.

1. Let K be the secret to be shared. Dealer will set up a polynomial $f(x)$ of degree 1.
   i.e., $f(x) = K + bx$, where $b \in {\mathbb{Z}_q}^*$.

2. For each minimal qualified subset in $\mathcal{A}^0$, an integer $a_1, a_2, \ldots, a_t \in {\mathbb{Z}_q}^*$ is chosen to represent the t qualified subsets.

3. Choose a random number $X_0 \in {\mathbb{Z}_q}^*$ and compute $Y_0 = X_0 G$ also $Y_i' = X_0 Y_i$, for $i = 1, 2, \ldots, n$.

4. Compute $f(1)$ and for each qualified subset $\mathcal{A}_j = \{P_{1j}, P_{2j}, \ldots, P_{dj}\}$ in $\mathcal{A}^0$, compute

   $A_j = f(a_j) \oplus H(Y_{1j}') \oplus H(Y_{2j}') \oplus \cdots \oplus H(Y_{dj}')$, $1 \leqslant j \leqslant t$.

5. Publish $Y_0, f(1), (a_1, A_1), (a_2, A_2), \ldots, (a_t, A_t)$ on the public bulletin.

8.3.4 Verification and Secret Reconstruction

The participants in each qualified subset $\mathcal{A}_j = \{P_{1j}, P_{2j}, \ldots, P_{dj}\}$, $1 \leqslant j \leqslant t$ can reconstruct the secret using the secret share and also the public values in the bulletin board. Each user contribute his pseudo share for the reconstruction of secret. The pseudo share is computed from his secret share and the public informations. The designated combiner can also identify the cheaters during the reconstruction phase using pairing.

1. Each participant P_{ij} in the qualified subset $\mathcal{A}_j$ get the public information Y_0 from the bulletin board and computes $Y'_{ij} = Y_0 X_{ij}$, using the secret share X_{ij}. The participant then delivers Y'_{ij} to the designated combiner.

2. Combiner checks $\hat{e}(G, Y'_{ij}) = \hat{e}(Y_0, Y_{ij})$. If it is not true then the share send by the participant P_{ij} is invalid and corrective measures have to be taken in this case.

3. Once all the valid shares are received, the combiner can retrieve $f(a_j) = A_j \oplus H(Y'_{1j}) \oplus H(Y'_{2j}) \oplus \cdots \oplus H(Y'_{dj})$.

4. Using $f(1)$ and $f(a_j)$, the polynomial can be reconstructed using Lagrange Interpolation [120].

$$f(x) = f(1).\frac{x - a_j}{1 - a_j} + f(a_j).\frac{x - 1}{a_j - 1} \tag{8.1}$$

5. The shared secret K is $f(0)$.

8.4 Security Analysis

One of the major requirement of a secret sharing scheme is the secure distribution of the shares to the participants by the Dealer. An untrustable Dealer may send inconsistent shares to the participant. The verifiable secret sharing ensures that the shares are consistent i.e., the authorized set of shares when combined will generate the same secret. Here the secret shares are chosen by the participant itself and send to the Dealer during the share generation. The pseudo shares are also used during the reconstruction. So the Dealer or any other participant does not have any idea about the secret share chosen by the participant. Combiner can also verify the shares send by the participants using this pseudo shares. Finding X_i from Y_i or finding X_0 from Y_0 is hard as solving the ECDLP. Hence the security of the secret

share depends on the hardness of solving ECDLP. It is noted that there is no efficient algorithm exist for solving the ECDLP. The birthday paradox method can be used to solve ECDLP which is having a running time of $O(n)$, where n is the order of the group. This attack can be broken by choosing a field of size at least 160 bits. In order to overcome the attack of the sub exponential algorithm used for solving discrete logarithm problem, we need a field of at least 1024 bit size. This shows that elliptic curve group provides more security with less number of bits. Thus the use of elliptic curve field instead of finite field results in savings of both time and space.

Cheating detection and Cheater identification is a major security requirement. It is done very efficiently using elliptic curve pairing. An efficient algorithm is proposed by Victor Miller [148] for computing the Weil pairing, which is having polynomial time complexity.

Theorem 8.4.1. *The probability that the participant distribute invalid shares during the reconstruction is negligible.*

proof. The designated combiner can verify the shares send by the participants by checking $\hat{e}(G, Y'_{ij}) = \hat{e}(Y_0, Y_{ij})$. If there is a mismatch, the participant is a cheater and we are able to detect and identify the cheater.

From the properties of bilinear pairing

$$\begin{aligned}
\hat{e}(G, Y'_{ij}) &= \hat{e}(Y_0, Y_{ij}) \\
\hat{e}(G, Y_0 X_{ij}) &= \hat{e}(GX_0, GX_{ij}) \\
\hat{e}(G, Y_0 X_{ij}) &= \hat{e}(GX_0, GX_{ij}) \\
\hat{e}(G, GX_0 X_{ij}) &= \hat{e}(GX_0, GX_{ij}) \\
\hat{e}(G, G)^{X_0 X_{ij}} &= \hat{e}(G, G)^{X_0 X_{ij}}
\end{aligned}$$

If this equation doesn't hold then the participant is a cheater.

During the reconstruction phase each participant submit $Y'_{ij} = X_{ij}Y_0$. The participant does not have to disclose his secret share X_{ij}. An attacker has to solve the ECDLP to find the secret share from the pseudo share which is computationally hard. The secret share can be reused with out compromising the security of the scheme. This provides more flexibility unlike other secret sharing schemes.

The list of participants in the authorized access structure can only reconstruct the secret. This is achieved with Shamir's secret sharing scheme. It is noted that Shamir's scheme provides information theoretic security. The security does not depends on the assumptions about any hard mathematical problem. The polynomial used in the scheme is having only degree one which makes the scheme computationally efficient. In order to reconstruct the degree one polynomial, two points are necessary. $f(1)$ is published in the bulletin board. The other value $f(a_j)$ can only be computed by the list of authorized participant in each authorized subset($\mathcal{A}_j$) mentioned in the minimal qualified set ($\mathcal{A}^0$). The participants not mentioned in the authorized subset cannot obtain $f(a_j)$ and hence cannot reconstruct the polynomial. No information about the secret $f(0)$ is thus revealed. Lagrange Interpolation can be done efficiently in $O(nlog^2n)$ time. However the reconstruction of the degree 1 polynomial from two points $(1, f(1))$ and $(a_j, f(a_j))$ can be done with four multiplication and an inverse computation. The addition and subtraction does not cost much.

The scheme can be easily extended to share multiple secrets. The Dealer has to construct polynomials of degree 1 corresponds to each secret $K_1, K_2, \ldots, K_m$ to be shared. The Dealer then publish $f(1)_i, Y_{0i}$ and A_{ji}, $1 \leqslant i \leqslant m$ corresponds to each secret along with other public parameters. It is noted that only public parameters need to be added to share more secrets. However participant shares remain same for the multi secret sharing.

The proposed scheme is dynamic in nature. When the Dealer wants to share a new secret, he just modify the public parameters. The participant does not need to alter their secret shadows. When a participant wants to change his secret shadow in case of leakage, he can do so. He will send the pseudo share to the Dealer after choosing a new secret share. The Dealer then modify the public parameters accordingly. This process does not affect the secret share of other participants. When a new participant joins the system and the access structure changes, then only the public parameters need to be changed. Other participants need not renew their secret shadows unlike other secret sharing schemes.

The verifiability is implicit in the scheme. Since the secret shadows are chosen by the participant itself, an adversary or the Dealer have no idea about the secret shadows. This also provides multi use capability where same shadows can be used for the sharing and reconstruction of several secrets.

It is noted that there is no secret communication exist between the Dealer and the participants. So the scheme avoids the need of a secure channel. The scheme is also efficient because of the low computational cost. Let us define the following terms to represent the time taken for each operation.

T_{ECM} – The time taken for computing nX, where n is a scalar and X is an elliptic curve point.

T_P – The time taken for Pairing.

T_H – The time taken for executing the Hash function H.

T_L – Time for polynomial reconstruction.

n – Total number of participants.

d – Number of participants in each qualified access set.

In the system initialization phase, each entity including the participants and the Dealer has to compute a public share from his secret share. This needs $(n+1)$ point multiplication. The cost is $(n+1)T_{ECM}$. Also the hash function has to be computed for each d participants set in the access structure. This need a computational cost of dT_H. The cost of XOR and the polynomial evaluation does not cost much. So the total computational cost in the initialization and the secret distribution phase is $O((n+1)T_{ECM} + dT_H)$. During the verification and secret reconstruction phase, each participant has to do a point multiplication. The combiner has to do two pairing operation for verification. The hash operations has to be done for each participant share involved in the secret reconstruction. The final secret is obtained by Lagrange Interpolation. It is noted that the polynomial used is of degree one. So the interpolation doesn't take too much computational cost. The computational cost involved in XOR operations is also negligible. Thus the total computational cost involved in the secret reconstruction and verification phase is $dT_{ECM} + 2T_P + dT_H + T_L$. It is noted that the overall computational cost depends mainly on the point multiplication by a constant. If X is an elliptic curve point then nX can be done efficiently by double-and-add method. Suppose $n = 2^k - 1$ then in the worst case it needs $2k$ point operations i.e., k additions and k multiplications. If we use ternary expansion then computing nX never requires more than $\frac{3}{2}k + 1$ point operations i.e., $k+1$ doubling and $\frac{1}{2}k$ additions.

8.5 Concluding Remarks

We have proposed a novel generalized multi secret sharing scheme based on elliptic curve and bilinear pairing in this chapter. The scheme is computationally efficient and provides more security. The scheme is multi use and dynamic in nature. Participants can be added, the access

structure can be changed or more secrets can be shared with out changing participants shadow. The shares are chosen by the participant itself. It avoids the verifiability problem and also the need for a secure channel. The pairing helps to identify the cheating and also to detect the cheaters. Unlike other multi secret sharing scheme, it is simple and easy to implement. The number of public parameters are also less. When a participant leaves the system, the access structure changes. We have to remove the participant entry from all the sets where the participant belongs and then modify the public parameters according to the new access structure. We have also done a detailed analysis of the proposed algorithm and mentioned the complexities involved in terms of the complexity of the elliptic curve point operations, pairing, time taken for hash function and Lagrange interpolation. One disadvantage with this scheme is that it cannot reconstruct the shared multiple secrets simultaneously.

Chapter 9

Threshold Multi-secret Sharing using Elliptic Curve and Pairing

9.1 Introduction

Elliptic curve and pairing are good candidates for developing secret sharing schemes with several extended capabilities. In the previous chapter we had seen a generalized scheme based on ECDLP and bilinear pairing. In this chapter we propose a threshold multi secret sharing scheme, where more than one secret can be shared according to the specified threshold access structure. The scheme make use of elliptic curve bilinear pairing and self pairing. Verification of share by the participants, shares consistency checking, detection and identification of cheaters are the extended capabilities achieved. Unlike the multi stage secret sharing scheme, all the shared secrets are retrieved in a single stage here. The participants can be added very easily. The scheme is efficient and the number of public values are also less compared with the existing threshold

multi secret sharing scheme based on the elliptic curve. The Dealer can modify the secret or add additional secret by changing the public parameters of the scheme. This is the first proposal of a threshold multi secret sharing scheme with extended capabilities using self pairing.

Elliptic curves were found numerous applications in cryptography [147]. Developed as a public key crypto system, it has found more secure with small key size compared with other public key crypto system. Elliptic Curve Discrete Logarithm Problem (ECDLP) is much harder compared with the Discrete Logarithm Problem(DLP). So the computational cost can be reduced while maintaining the same level of security with small key size. In 1993 Meneze's et al [145] introduced pairing. Pairing is introduced to show an attack on elliptic curve discrete logarithm problem and later found useful applications. Pairing on elliptic curve have found useful applications in key exchange, identity based encryption etc [67]. The use of elliptic curve and pairing have found applications in secret sharing schemes very recently. Several schemes based on threshold and generalized secret sharing is proposed and they have found useful applications. Pairing can be used to introduce verifiability in secret sharing scheme with more security.

Chen Wei et al [218] in 2007 proposed a dynamic threshold secret sharing scheme based on bilinear maps. Each participant holds a permanent private key. The threshold is realized by adjusting the number of linear equations. The scheme also having cheating detection capability. But it is a single secret sharing scheme. A threshold multi secret sharing scheme based on elliptic curve discrete logarithm is proposed by Runhua Shi et al [194] in 2007. A fast multi-scalar multiplication scheme is also introduced. Sharing multiple secrets which are represented as points on elliptic curve using self pairing is proposed by Liu et al [137] in 2008. The proposed scheme is based on Liu et al scheme. Chen's scheme is modified to share multiple secrets by S. J. Wang et al [216] in 2009. In Wang's et al

scheme, number of secrets must be less than or equal to the threshold and also more public values must be changed when the secret need to be updated. Eslami et al [69] in 2010 proposed a modified scheme which avoids these problems.

Several publicly verifiable secret sharing schemes are proposed based on pairing. But most of them are single secret sharing schemes. A practical publicly verifiable secret sharing scheme based on pairing is proposed by Youliang Tian et al [212] in 2008. A pairing based publicly verifiable secret sharing is introduced by Wu and Tseng [220] in 2011. An efficient verifiable secret sharing scheme is proposed by Jie Zhang et al [223]. Tian et al [211] proposed a distributed publicly verifiable secret sharing scheme. An efficient One Stage Multi Secret Sharing(OSMSS) is proposed recently in 2014 by Fatemi et al [70]. The scheme makes use of bilinear pairing. The number of public values are reduced and the scheme is more efficient compared with the previous schemes.

9.2 Elliptic Curve and Self Pairing

A self pairing and its applications are proposed by Lee [133] in 2004. The pairing which map $e : G \times G \implies G$ is called self pairing.

Let K be a field with characteristic zero or a prime p and $E = E(\bar{K})$ be an elliptic curve over $\bar{K}$, where $\bar{K}$ is an algebraic closure of K. Consider the set of all torsion points of order l that is $lP = \mathcal{O}$. These points forms a subgroup $E_K[l]$ of $E(K)$ where $l \neq 0$. $E[l]$ can be represented as a direct sum of two cyclic groups. $E[l] \cong Z_l \oplus Z_l$. That is any point in $E[l]$ can be represented as a linear combination of two generating pair G and H of $E[l]$. Consider the points $P = r_1G + s_1H$ and $Q = r_2G + s_2H$ in $E[l]$, where r_1, r_2, s_1 and s_2 are integers in $[0, l-1]$. We can define pairing for some fixed integers, $\alpha, \beta \in [0, l-1]$, in the following way

$$e_{\alpha,\beta} : E[l] \times E[l] \implies E[l]$$

$$e_{\alpha,\beta} : (r_1 s_2 - r_2 s_1)(\alpha G + \beta H)$$

When α, β are zero $e_{\alpha,\beta}$ is trivial. This case is not considered.

Proposition 9.1. The pairing $e_{\alpha,\beta} : E[l] \times E[l] \to E[l]$ will satisfy following properties

(i) Identity: For all $A \in E[l]$, $e_{\alpha,\beta}(A, A) = \mathcal{O}$

(ii) Bilinearity: For all $A, B, C \in E[l]$

$$\begin{aligned} e_{\alpha,\beta}(A + B, C) &= e_{\alpha,\beta}(A, C) + e_{\alpha,\beta}(B, C) \\ e_{\alpha,\beta}(A, B + C) &= e_{\alpha,\beta}(A, B) + e_{\alpha,\beta}(A, C) \end{aligned}$$

(iii) Anti-symmetry: For all $A, B \in E[l]$, $\quad e_{\alpha,\beta}(A, B) = -e_{\alpha,\beta}(B, A)$

(iv) Non-degeneracy: If $A \in E[l]$, $e_{\alpha,\beta}(A, \mathcal{O}) = \mathcal{O}$. Moreover if $e_{\alpha,\beta}(A, B) = \mathcal{O}$, for all $B \in E[l]$ then $A = \mathcal{O}$.

9.3 Liu et al Scheme

Liu et al [137] proposed a point sharing method in elliptic curve using self pairing. In this scheme multiple secrets $K_1, K_2, \ldots, K_m$ are shared, which is represented as points on the Elliptic curve. The scheme consist of four main steps.

1. Initialization.
2. Share Distribution.
3. Secret Sharing.
4. Secret Reconstruction.

The initialization and share distribution phase need to be done only once for a particular (t, n) threshold secret sharing scheme, where t is the threshold. Secret can be dynamically changed or more secret can be added with out modifying the participants secret share. This is achieved with a public notice board, where every user have the access but only the Dealer can modify the data.

9.3.1 Initialization

It is assumed that the Dealer is a trusted authority and the participants $U_1, U_2, \ldots, U_n$ are honest. In the initialization phase some public parameters are posted on the public bulletin called notice board which can be accessed by every participant.

1. The Dealer (**D**) chooses an elliptic curve E over $GF(q), q = p^r$, where p is a large prime such that DLP and ECDLP in GF(q) is hard. **D** then choose $E[l] \subseteq E(GF(q^k))$, a torsion subgroup of large prime order l.

2. **D** chooses a generating pair $\{G, H\} \in E[l]$, α and β, which are used for pairing. Dealer then compute $W = \alpha G + \beta H$.

3. **D** then publish $\{E, q, l, k, W\}$ in the notice board.

9.3.2 Share Distribution

In this phase, shares needed to reconstruct the secret are distributed to n participants and any t of them can reconstruct the secret. These shares are independent of the secret to be distributed.

1. **D** generate a matrix A of size $n \times t$ as:
$$A = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 2^2 & \dots & 2^{t-1} \\ 1 & 3 & 3^2 & \dots & 3^{t-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & n & n^2 & \dots & n^{t-1} \end{pmatrix}$$

2. **D** randomly choose t pairs of numbers $a_i\prime, b_i\prime \in [1, l-1], 1 \leqslant i \leqslant t$ and computes

$$\begin{aligned} (a_1, a_2, \dots, a_n)^T &= A.(a_1\prime, a_2\prime, \dots, a_t\prime)^T \\ (b_1, b_2, \dots, b_n)^T &= A.(b_1\prime, b_2\prime, \dots, b_t\prime)^T \end{aligned}$$

3. **D** sends $P_j = \{a_j, b_j\}$ as a secret share to each user $U_j, 1 \leqslant j \leqslant n$ through a secure channel.

9.3.3 Secret Sharing

After distributing the secret shares, the Dealer will share multi-secret by publishing informations on a public notice board corresponding to each secret to be shared. Each participant can make use of these public values. These public information together with the secret share of each participant can be used for the retrieval of shared secrets. The number of secrets m must be less than or equal to the threshold t for the scheme to work.

1. The secrets to be shared i.e.; $K_1, K_2, \dots, K_m$ is mapped into m points on the Elliptic curve $M_1, M_2, \dots, M_m$.

2. **D** chooses $\{c_i, d_i\} \in [0, l-1]$ randomly and computes $Q_i = c_iG + d_iH$ and $R_i = e_{\alpha,\beta}(Q_i, P_i\prime) + M_i$, for all $1 \leqslant i \leqslant m$.

3. **D** then publish $\{c_i, d_i, R_i\}$,$1 \leqslant i \leqslant m$, on the public bulletin.

9.3.4 Secret Reconstruction

Let t users $U_1, U_2, \ldots, U_t$ wants to reconstruct m secrets. Each user contribute his pseudo share for the reconstruction of secret. The pseudo share is computed from his secret share and the public informations.

For each secret K_i from 1 to m and for each user U_j from 1 to t

1. Each U_j download the pair $\{c_i, d_i\}$ from the public bulletin and compute pseudo share $S_{ij} = e_{\alpha,\beta}(Q_i, P_j)$, where $P_j = a_jG + b_jH$ and $Q_i = c_iG + d_iH$, $1 \leqslant i \leqslant t$ and $1 \leqslant j \leqslant t$.

2. Each user U_j multi-casts the pseudo share S_{ij} to other $t-1$ participants.

3. Each user then computes $T_i = \sum_{k=1}^{t} y_k S_{ij}$, where $y_k = \prod_{j=1, j\neq k}^{t} (k-j)^{-1}$.

4. Each user can download the point R_i from the public bulletin and recovers $M_i = R_i - T_i$.

9.4 Proposed Multi-secret Sharing Scheme

The major difficulties with Liu's scheme is that the secrets are represented as points on the elliptic curve. The mapping of secret to the elliptic curve point is very difficult. The number of secrets that can be shared also depends on the threshold t. We cannot share more than t secrets. The Dealer is assumed to be a trusted authority. There is no provision for the verification of the shares distributed by the Dealer and also the participants cannot verify the shares distributed by the other participant during the reconstruction. The proposed scheme overcome these difficulties. We make use of self pairing and bilinear pairing for the efficient construction of the scheme.

The proposed scheme is a threshold multi secret sharing scheme where any number of secrets can be shared and threshold number of users can reconstruct the multi secrets. The multi secrets can be reconstructed in single stage unlike the multi stage secret sharing scheme where the secrets are reconstructed stage by stage. Each user can verify the shares during the secret share distribution phase by the Dealer. The participants can also verify the shares send by other participants during the reconstruction phase to identify the cheaters.

The proposed scheme consist of the following three phases

1. Initialization and Secret Sharing.
2. Secret Reconstruction.
3. Verification.

9.4.1 Initialization and Secret sharing

In the initialization phase some public parameters are posted on the public bulletin called notice board which can be accessed by every participant. Let $U_1, U_2 \ldots U_n$ be the n users involved in the secret sharing phase and let $K_1, K_2, \ldots, K_m$ be the m secrets to be shared.

1. The Dealer(**D**) chooses an Elliptic curve E over $GF(q)$, where $q = p^r$ and p is a large prime such that DLP and ECDLP in GF(q) is hard. **D** then choose $E[l] \subseteq E(GF(q^k))$, a torsion subgroup of large prime order l.
2. **D** chooses a generating pair $\{G, H\} \in E[l]$, α and β, which are used for pairing and then compute $W = \alpha G + \beta H$.
3. **D** then publish $\{E, q, l, k, G, H, W\}$ in the notice board.
4. A secret point P_0 is chosen where $P_0 = a_0 G + b_0 H$.

5. The Dealer then construct a polynomial of degree $t-1$.

$$f(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{t-2}x^{t-2} + b_0x^{t-1}$$

where a_0 and b_0 corresponds to the point P_0 and the other coefficient values are chosen from ${\mathbb{Z}_l}^*$.

6. **D** compute shares $S_i = f(x_i) \pmod{l}$ and send the shares $P_i = (x_i, S_i)$ secretly to the users U_i, for $i = 1, \ldots, n$.

7. The Dealer also publishes the values c, d corresponds to a point $Q = cG + dH$ and the verification point $V_i = e_{\alpha,\beta}(P_i, Q)$ for $i = 1, \ldots, n$ corresponds to each share and also $V_0 = e_{\alpha,\beta}(P_0, Q)$.

8. Publish the recovery code $R_i = K_i - e(P_0, iP_0)$ for $i = 1, \ldots, m$ corresponds to each secret that is to be shared.

9.4.2 Secret Reconstruction

1. When the threshold number of users want to reconstruct the secret, they pool the shares and reconstruct the polynomial $f(x)$ using Lagrange Interpolation.

$$f(x) = \sum_{j=1}^{t} S_j \prod_{1 \leqslant i \leqslant t, i \neq j} \frac{x - x_i}{x_j - x_i}$$

2. From the reconstructed polynomial, a_0 and b_0 can be obtained and hence P_0 can be obtained.

3. Using the published recovery codes, the m secrets can be recovered by

$$K_i = R_i + e(P_0, iP_0)$$

for $i = 1, \ldots, m$.

9.4.3 Verification

Each user can verify the shares using the share verification point V_i.

1. Each user compute $v_i = e_{\alpha,\beta}(P_i, Q)$ using his share and the public value Q

2. If the computed value $v_i = V_i$, the share send by the Dealer is valid.

The validity of the shares send by each user can be verified by using the same technique mentioned above during the reconstruction stage. The consistency of the shares are also verified by checking V_0. The polynomial is reconstructed after all the shares are pooled. Using a_0, b_0 and the generator G, H, the value of P_0 can be obtained. The participant can verify the consistency of the shares by checking $V_0 = v_0$, where $v_0 = e_{\alpha,\beta}(P_0, Q)$ and V_0 is the corresponding published value.

9.5 Security Analysis

One of the major requirement of the secret sharing scheme is the consistency of the shares.

Theorem 9.5.1. *The probability that the Dealer distribute inconsistent shares to the participant is negligible.*

proof. The coefficients of the polynomial $f(x)$ are chosen by the Dealer. The values of x_i are chosen randomly and are send along with the evaluated polynomial value to the participant as shares. However the Dealer cannot send invalid shares to the participants because the share (x_i, S_i) can be verified by each participant using self pairing with the public value Q. The consistency of the shares can also be verified by checking $e_{\alpha,\beta}(P_0, Q) = V_0$. If the shares are inconsistent then the computed value will not match with V_0.

Theorem 9.5.2. *The probability that the participant distribute invalid shares during the reconstruction is negligible.*

proof. Each participant can verify the shares send by the other participants by checking $e_{\alpha,\beta}(P_i, Q) = V_i$. If there is a mismatch, the participant is a cheater and we are able to detect and identify the cheater.

Theorem 9.5.3. *Adversary cannot derive any information about the secret from the public values.*

proof. The polynomial $f(x)$ is of degree $t-1$, so less than t participant cannot derive any useful information about the secret by pooling their shares. The verification code $Vi = e_{\alpha,\beta}(P_i, Q)$ cannot reveal any info about the share P_i. The point V_i is a linear combination of the generator (G, H) and the coefficients can be any value from the field $\mathbb{Z}_l^*$. The public values R_i also cannot give any information about the shared secret. The non degeneracy of pairing ensures that $e(P_0, P_0)$ is a primitive l^{th} root of unity for all non zero $P_0 \in E(F_q)[l]$.

Theorem 9.5.4. *Finding P_0 is as difficult as guessing the secret.*

proof. The security of the system depends on finding P_0 and is again depends on a_0 and b_0. These values can be retrieved only by reconstructing the $t-1$ degree polynomial by the t users involved in secret reconstruction. The shares have the same size as these parameters and are elements of the same field. This provides information theoretical security. Less than t participant cannot derive any useful information because of the security of Shamir's scheme. Hence the adversary can only guess the values. The probability is $1/l$. When l is large this probability is very less. Finding P_0 without knowing a_0 and b_0 is again like trying all linear combinations of the generators G and H, which is again a more complex process. Without

knowing P_0 finding the secret from the public parameter R_i is as complex as guessing the secret.

9.6 Experimental Results

SAGE and Python were used for implementing the scheme. A simple example showing sharing of two secrets K_1, K_2 according to $(2, 3)$ threshold scheme is mentioned here. We considered field with smaller prime power for easy understanding.

9.6.1 Initialization

1. Elliptic Curve defined by $E : y^2 = x^3 + 4.x$ over Finite Field in i of size 47^6 is chosen for secret sharing. The order of E is $10779422976(2^8.3^4.7^2.103^2)$. Additive Abelian group isomorphic to $Z/103824 + Z/103824$ is embedded in Abelian group of points on the curve.

2. The Dealer **D** chooses a torsion subgroup $E[103] \subseteq E(GF(47^6))$. Two randomly chosen generator pairs G, H of $E[103]$ are

$$\begin{aligned} G = \quad & (19i^5 + 38i^4 + 26i^3 + 28i^2 + 45i + 6 : \\ & 20i^5 + 18i^4 + 12i^3 + 32i^2 + 12i + 43 : 1) \\ H = \quad & (5i^5 + 8i^4 + 41i^3 + 46i^2 + 39i + 34 : \\ & 32i^5 + 7i^4 + 18i^3 + 34i^2 + 8i + 32 : 1) \end{aligned}$$

   Let $\alpha = 51, \beta = 35$, compute $W = \alpha G + \beta H$

$$\begin{aligned} W = \quad & (25i^5 + 3i^4 + 11i^3 + 15i^2 + 39i + 19 : \\ & 40i^5 + 41i^4 + 9i^3 + 44i^2 + 22i + 1 : 1) \end{aligned}$$

3. $E : y^2 = x^3 + 4x, q = 47^6, l = 103, k = 1, \alpha = 51, \beta = 35, W = (25i^5+3i^4+11i^3+15i^2+39i+19 : 40i^5+41i^4+9i^3+44i^2+22i+1 : 1)$ are made public.

9.6.2 Share Distribution

1. The matrix A for a $(2, 3)$ scheme is of size 3×2

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix}$$

2. Dealer chooses two pairs of random numbers from $[1, 102]$ $a_1\prime = 11, a_2\prime = 25, b_1\prime = 15, b_2\prime = 33$ and compute

$$\begin{aligned} A \times [a_1\prime, a_2\prime] &= [36, 61, 86] &= [a_1, a_2, a_3] \\ A \times [b_1\prime, b_2\prime] &= [48, 81, 114] &= [b_1, b_2, b_3] \end{aligned}$$

3. The three users U_1, U_2, U_3 will get shares $P_1 = (36, 48), P_2 = (61, 81), P_3 = (86, 114)$.

9.6.3 Secret Sharing

1. We consider two secrets K_1 and K_2 to be shared. These secrets are mapped into elliptic curve points M_1 and M_2.

$$\begin{aligned} M_1 = \quad & (19i^5 + 38i^4 + 26i^3 + 28i^2 + 45i + 6 : \\ & 20i^5 + 18i^4 + 12i^3 + 32i^2 + 12i + 43 : 1) \\ M_2 = \quad & (5i^5 + 8i^4 + 41i^3 + 46i^2 + 39i + 34 : \\ & 32i^5 + 7i^4 + 18i^3 + 34i^2 + 8i + 32 : 1) \end{aligned}$$

2. In order to share the two secrets, **D** chooses two random pairs of numbers $(c_1, d_1), (c_2, d_2) \in [0, l-1]$ and compute $Q_i = c_i G + d_i H$ for the two secrets. Dealer computes pairing $R_i = e_{\alpha,\beta}(Q_i, P_i\prime) + M_i$ corresponds to each secret, where $P_i\prime = a_i\prime G + b_i\prime H$.

   Let $c_1 = 15, d_1 = 11, c_2 = 23, d_2 = 39$ and $a_1\prime = 11, b_1\prime = 15, a_2\prime = 25, b_2\prime = 33$.

$$
\begin{aligned}
R_1 &= e_{51,35}(Q_1, P_1\prime) \\
R_1 &= (c_1 b_1\prime - d_1 a_1\prime)(\alpha G + \beta H) + M_1 \\
R_1 &= (15.15 - 11.11)W + M_1 \\
R_1 &= (i^5 + 27i^4 + 5i^3 + 7i^2 + 35i + 38 : \\
&\quad 21i^5 + 44i^4 + 28i^3 + 15i^2 + 9i + 16 : 1)
\end{aligned}
$$

$$
\begin{aligned}
R_2 &= e_{51,35}(Q_2, P_2\prime) \\
R_2 &= (c_2 b_2\prime - d_2 a_2\prime)(\alpha G + \beta H) + M_2 \\
R_2 &= (23.33 - 25.39).W + M_2 \\
R_2 &= (33i^5 + 43i^4 + 20i^3 + 17i^2 + 39i + 33 : \\
&\quad 45i^5 + 5i^4 + 43i^3 + 24i^2 + 41i + 38 : 1)
\end{aligned}
$$

3. Publish

$$\begin{aligned}
\{c_1, d_1, R_1\} &= \{15, 11, (i^5 + 27i^4 + 5i^3 + 7i^2 \\
&\quad +35i + 38 : \\
&\quad 21i^5 + 44i^4 + 28i^3 + 15i^2 \\
&\quad +9i + 16 : 1)\} \\
\{c_2, d_2, R_2\} &= \{23, 39, (33i^5 + 43i^4 + 20i^3 + 17i^2 \\
&\quad +39i + 33 : \\
&\quad 45i^5 + 5i^4 + 43i^3 + 24i^2 + \\
&\quad 41i + 38 : 1)\}
\end{aligned}$$

9.6.4 Secret Reconstruction

Assume that participants P_1 and P_2 want to reconstruct the secrets K_1, K_2.

1. Each participant compute his share contribution for the reconstruction of each secret as

$$S_{ij} = e_{\alpha,\beta}(Q_i, P_i)$$

$$
\begin{aligned}
S_{11} &= e_{51,35}(Q_1, P_1) \\
S_{11} &= (c_1.b_1 - d_1.a_1).W \\
S_{11} &= (15.48 - 11.36).W \\
S_{11} &= (8i^5 + 5i^4 + 2i^3 + 29i^2 + 13i + 7 : \\
&\quad 4i^5 + 29i^4 + 44i^3 + 43i^2 + 6i + 20 : 1) \\
S_{12} &= e_{51,35}(Q_1, P_2) \\
S_{12} &= (15.81 - 11.61).W \\
S_{12} &= (37i^5 + 28i^4 + 31i^3 + 15i^2 + 40i + 44 : \\
&\quad 38i^5 + 5i^4 + 3i^3 + 39i^2 + 26i + 8 : 1) \\
S_{21} &= (5i^5 + 36i^4 + 31i^3 + 34i^2 + 5i + 38 : \\
&\quad 5i^5 + 22i^4 + 13i^3 + 4i^2 + 39i + 17 : 1) \\
S_{22} &= (25i^5 + 3i^4 + 11.i^3 + 15i^2 + 39i + 19 : \\
&\quad 7i^5 + 6i^4 + 38i^3 + 3i^2 + 25i + 46 : 1)
\end{aligned}
$$

2. The shares generated are then multi-casted.

3. The inverse of the matrix A is $\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$. Each user then compute

$y_k S_{ij}$.

$$\begin{aligned} T_1 &= 2S_{11} - 1S_{12} \\ T_1 &= (25i^5 + 3i^4 + 11i^3 + 15i^2 + 39i + 19 : \\ &\quad 40i^5 + 41i^4 + 9i^3 + 44i^2 + 22i + 1 : 1) \\ T_2 &= -1S_{21} + 1S_{22} \\ T_2 &= (29i^5 + 43i^4 + 19i^3 + 20i^2 + 36i + 25 : \\ &\quad 45i^5 + 9i^4 + 29i^3 + 15i^2 + 9i + 31 : 1) \end{aligned}$$

4. Each participant can download R_i from the public bulletin and reconstruct $M_i = R_i - T_i$.

$$\begin{aligned} M_1 &= R_1 - T_1 \\ M_1 &= (19i^5 + 38i^4 + 26i^3 + 28i^2 + 45i + 6 : \\ &\quad 20i^5 + 18i^4 + 12i^3 + 32i^2 + 12i + 43 : 1) \\ M_2 &= R_2 - T_2 \\ M_2 &= (5i^5 + 8i^4 + 41i^3 + 46i^2 + 39i + 34 : \\ &\quad 32i^5 + 7i^4 + 18i^3 + 34i^2 + 8i + 32 : 1) \end{aligned}$$

From M_1, M_2, K_1 and K_2 can be obtained.

The comparison of various schemes which uses elliptic curve and bilinear pairing is studied and is shown in Table 9.1. We considered recent proposal of only the threshold multi secret sharing schemes based on elliptic curve and pairing. Lets consider a (t, n) threshold multi secret sharing scheme which can share m secrets. It is found that in Liu's [137]

Table 9.1: comparison of various schemes using elliptic curve and pairing

Scheme	Liu [137]	Chen [218]	Wang [216]	Eslami [69]	Proposed
single secret(ss) /multi secret(ms)	ms	ss	ms	ms	ms
secrets	t	1	t	m +n	$m \geqslant n$
public parameters	5 + 3m	8 +n-t	8+2n	8+n+m-t	7+n+m
single stage	Yes	Yes	Yes	Yes	Yes
verifiability	No	Yes	Yes	Yes	Yes
cheater detection	No	Yes	No	Yes	Yes
cheater identification	No	Yes	No	Yes	Yes

and Wang's [216] scheme, the number of secret that can be shared is proportional to t. So these schemes are not suitable for sharing more than t secrets. Wangs scheme is a modification of Chen's single secret sharing scheme. Eslami again modified the Wang's scheme. The advantage of our proposed scheme is that, large number of secret can be shared and also the public values used are less. Most of the schemes mentioned in the literature having the share verification property. But they use discrete logarithm problem. This verification code can reveal information about the secret if discrete logarithm problem is tractable. Hence the security of the shared secret also depends on the hardness of the DLP problem. This is avoided in our scheme. The verification code does not reveal any information about the secret and is more secure compared with the existing scheme.

9.7 Concluding Remarks

In this chapter, we have proposed a novel threshold multi-secret sharing scheme based on elliptic curve and bilinear pairing. Most of the schemes proposed in the literature use bilinear pairing for verification of shares, cheater detection and identification. We have used the method of point sharing and verification using self pairing. A non degenerate Tate pairing or modified Weil pairing can be used to share multiple secrets. The number of public parameters are greatly reduced and the security does not depend on the hard computational problem. The verification mechanism can prevent users from cheating. The consistency of the shares can be verified by the participants which avoids the need of a trusted Dealer. The proposed scheme is the first threshold multi secret sharing scheme based on self pairing with the extended capabilities of share verification and cheater identification. The use of elliptic curve and self pairing can be further explored to develop secret sharing schemes with more generalized access structures.

Chapter 10

Secret Sharing Applications

10.1 Introduction

Secret sharing have found several useful applications. Originally started as a solution to safeguard secret keys, it has found numerous applications in various security protocols. In this chapter we describe two such applications developed based on the secret sharing schemes. E-voting using secret sharing based Secure Multi-party Computation (SMC) and CTS (Cheque Truncation System) based on secret image sharing. Apart from this, there are several application areas where secret sharing can be effectively utilized. Secret sharing schemes have found numerous applications in designing several cryptographic protocols. Threshold cryptography [63], access control [153], secure multi-party computation [14] [48] [55], authenticated group key transfer protocol [92], broadcast

Some results of this chapter are included in the following paper.
Divya G. Nair, Binu V. P, G. Santhosh Kumar, "An Improved E-Voting Scheme using Secret Sharing based Secure Multi-Party Computation ", Eighth International Conference on Computer Communication Networks (ICCN 2014), Banglore, Elsevier, ISBN :9789351072539,P-17

encryption [19], attribute based encryption [21] [85], generalized oblivious transfer [191] [209], visual cryptography [152] etc. The secret sharing scheme can also be used for the secure distributed storage of data with out encryption. This adds trust and reliability. We have developed a simple application based on the scheme mentioned in Chapter 4 for the distributed data storage [151]. It is not included in the thesis because it is in the development stage. We are exploring more on this area in our future endeavor.

10.2 Secret Sharing Homomorphism and Secure E-voting

10.2.1 Introduction

Secure E-voting is a challenging protocol. Several approaches to e-voting, based on homomorphic crypto systems, mix-nets, blind signatures etc are proposed in the literature. But most of them need complicated homomorphic encryption which involves complicated encryption decryption process and key management which is not efficient. In this chapter we propose a secure and efficient E-voting scheme based on secret sharing homomorphism. Here E-voting is viewed as a special case of multi party computation, where several voters jointly compute the result without revealing his vote. Secret sharing schemes are good alternative for secure multi party computation. They are computationally efficient and secure compared with the cryptographic techniques. It is the first proposal, which makes use of the additive homomorphic property of the Shamir's secret sharing scheme and the encoding-decoding of votes to obtain the individual votes obtained by each candidates apart from the election result. We have achieved integrity and privacy while keeping the efficiency of the system.

10.2.2 E-voting

Voting is a distributed decision making process involving several people. Each participant called the voter casts a vote and the computations are performed on the vote casts by different voters to select the preferred candidate. Voting can be modeled as a secure multi party computation system because multiple parties submit input and obtain the result without knowing any details of other users input.

The process involved in traditional election is quite tedious, time and resource consuming. To overcome these difficulties E-voting system is introduced. The evolving new technologies made E-voting practical. But the research in this direction has to go a long way. The reliability and security are the major challenges. E-voting provides a lot of benefits compared with traditional voting. It avoids the requirement of geographical proximity of users. The cost can be greatly reduced because the resources can be reused. The use of E-voting must satisfy the security requirements such as authentication, voter privacy, confidentiality, integrity, etc. The security flaws make E-voting vulnerable than traditional system.

The first electronic election scheme was proposed by David Chaum [50] in 1981. Electronic voting systems catering to different requirements have been widely implemented and used. There have been several studies on using computer technologies to improve elections. In 1987 Benaloh [18] presents an election scheme based upon secret sharing and the prime residuosity assumption. Boyd et al [32] in 1990 proposed multiple key cipher without a trapdoor function and presents a voting scheme as an application of said cipher. Iverson and Kenneth [108] in 1992 made proposals for application of secret sharing technique and zero knowledge technique in secure election. Fujioka et al [73] suggested a practical secret voting scheme for large scale elections in 1993. In this voting scheme, voting is managed by an administrator who registers and authenticates

voters and a counter who tallies votes. Gritzalis et al [86] [87] mentioned the requirements of a secure E-voting system.

Confidentiality, Authenticity, Integrity and Verifiability are the major security requirements in E-voting scenario. Confidentiality ensures that nobody knows whom the voter is voted. Authentication is an important process where each voter must be identified as a person he claims to be and he should not be allowed to vote again. Integrity of the votes are also important. The system should ensure that the votes are valid and any modification must be detected. Verifiability means any one can verify at later time that the voting is properly performed or his vote was properly registered and has been taken into account in the final tally [73].

The are several proposal for efficient secret ballot elections based on mix-nets [111] [164] [183], homomorphic encryption [16] [18] [54] [56] [73] [157] [182] and blind signatures [224]. There are different methods addressing the security and reliability of the E-voting scheme. Most of the approaches are based on cryptography. The major objective is to protect the voters identity from the vote. Secure E-voting using Blind Signature is proposed in [102]. RSA [178] and Blind signatures are the major cryptographic algorithms involved [42] [47]. Homomorphic encryption techniques are used in several implementations [168]. E-voting scheme proposals using verifiable secret sharing schemes are also given in [16] [18]. Several modifications and use of homomorphic encryption and verifiable secret shuffle are mentioned in [98] [132] [154]. Malkhi et al [142] in 2003 gave constructions without cryptographic technique which uses secret sharing homomorphism. Iftene [105] in 2007 proposed a general secret sharing scheme for E-voting using Chinese Remainder Theorem. Pailliar's crypto system and its application to voting is proposed by Damagaard et al [57] in 2010. Discrete logarithm problem and secret sharing are used by Chen et al [51] in 2014. Scheme with enhanced confidentiality and privacy is suggested by Pan et al [160] in 2014.

Secret sharing and many variations of its form an important primitive in several security protocols and applications. In the proposed method we make use of Shamir's [190] secret sharing techniques and its additive homomorphism property for efficient implementation of E-voting and vote tallying. This avoids the complicated encryption decryption process and key management. Details of Shamir's secret sharing schemes are given in Chapter 1. The shares in this scheme are information theoretically secure and provides no information about the secret key. The scheme is also ideal.

Properties of polynomials give Shamir's scheme a $(+,+)$ homomorphic property. The secret domain and the share domain is same (integers modulo p). There are other schemes [3] [123] also having $(+,+)$ homomorphism property. We consider Shamir's scheme for the ease of implementation and also it is information theoretically secure. The homomorphism also provides verifiable secret sharing. It is very important in secure multi party computation. The first proposal of verifiable secret sharing was done by Chor et al [53]. In secret sharing not only the participant but also the Dealer may be malicious. So the participant must be able to verify whether the shares are consistent. A set of n shares is t consistent if every subset of t of the n shares defines the same secret. Publicly verifiable secret sharing scheme's are introduced by Stadler in 1996 [203]. Schoenmakers [186] in 1999 proposed a publicly verifiable secret sharing scheme (PVSS) with applications to E-voting. The scheme is better than the schemes mentioned in [54] [56]. The issue of homomorphic secret sharing for PVSS is also discussed. An efficient PVSS is needed for the secure implementation of E-voting.

10.2.3 Secret Sharing Homomorphism

Secret sharing homomorphism is introduced by Benaloh in 1987 [17]. It is noted that Shamir's scheme is additive homomorphic. He stated that

any t of the n agents can determine the super secret and no conspiracy of fewer than t agents can gain any information at all about any of the sub secrets. That is the sum of the shares of different sub secret when added up and then interpolate according to the threshold mentioned to obtain the master secret which is the sum of the sub secrets. He also mentioned the importance of secret sharing homomorphism to E-voting.

Shamir's secret sharing scheme has the $(+,+)$ homomorphism property. For example, assume there are two secrets K_1, K_2 and are shared using polynomials $g(X)$ and $f(X)$. If we add the shares $h(i) = g(i) + f(i)$, $1 \leqslant i \leqslant n$, then each of these $h(i)$ can be treated as the shares corresponds to the secret $K_1 + K_2$. The polynomial $h(X) = g(X) + f(X)$ and $h(0) = K_1 + K_2$. Additive homomorphism of Shamir's secret sharing can be used to build an e-voting scheme. But each voter choose 1 or 0 (vote or no vote). The shares are send to n tellers. Any t of them can collaborate to retrieve the result back.

In case of PVSS, two operations are defined. One on the shares $\oplus$ and the other operation $\otimes$ on the encrypted shares such that for all participants

$$E_i(s_i) \otimes E_i(s_i') = E(s_i \oplus s_i')$$

. If the underlying secret sharing scheme is homomorphic then by decrypting the combined encrypted shares, the recovered secret will be equal to $s_i \oplus s_i'$.

10.2.4 Proposed Scheme

The proposed system is a modification of the existing electronic voting scheme's used in India. Currently electronic voting machines are used in polling booth. These machines are costly and also not reliable. We propose an alternative solution for this using an online system which uses secret sharing homomorphism. This add trust and reliability to the existing voting scheme by incorporating secret sharing based techniques. The secrecy of

vote is an important issue. This needs to be addressed with ultimate care. In the current Electronic Voting System (EVS), when a vote is casted, the corresponding candidate data base entry is updated and it can be easily tracked. But in the proposed scheme, it is difficult to track the vote because the shares of the votes are added and updated in all the servers. We also add trust to the existing scheme by maintaining more than one server to keep the voting details. We are not considering on-line verification of the authenticity of the voter as in general e-voting scheme. Here we assume that the polling officers in each polling booth has to do it manually using the electoral role. The major components of the proposed e-voting schemes are

1. Voting Terminal
2. Share Generation
3. Collection Centers
4. Result Computation

We have considered the user authentication process which is done manually. The voting takes place in a Polling station. A voter is allowed to vote after his identity is verified. A polling station may contain many voting terminals. The user interface shows a voting panel which contains the list of all contesting candidates and their party symbols. Voting panel is setup and managed by the Chief Election Officer.

The share generation module is responsible for receiving the vote casted by each voter and make shares of it according to the threshold secret sharing scheme. The shares are generated according to the vote casted for each candidates. Each candidate vote is represented as an encoded binary code. So when a vote is casted, the shares of the decimal value corresponds to the encoded binary vote of each candidate is generated using the Shamir's

secret sharing scheme. The number of bits in the encoded binary code corresponds to each candidate vote depends on the number of contesting candidates and also total number of voters.

Let us assume that there are m contesting candidates $C_1, C_2, \ldots, C_m$ and n voters $V_1, V_2, \ldots, V_n$. Then the binary encoding of the vote corresponds to each candidate will consist of $(\lfloor log_2 n \rfloor + 1) \times m$ number of bits. Here we consider the fact that all voters may vote for the same candidate. So the number of bits required for the representation of votes for each candidate is equal to the number of bits required to represent the total number of voters which is $\lfloor log_2 n \rfloor + 1$.

The encoding of the vote corresponds to each contesting candidate is explained below with an example. Let us consider that there are three candidates and seven voters. So the total number of bits of each encoded vote will be nine. The bit pattern corresponds to the vote of each candidate is obtained by setting the corresponding bit C_i to 1 in the code $00C_300C_200C_1$ and other bits C_i to 0. For example the code corresponds to the vote of candidate C_3 is 001000000(64). So depending on the vote casted, it is encoded into a decimal code of 1, 8 or 64 respectively. This bit wise encoding helps in computing the total votes obtained by each candidate using the additive homomorphism.

The encoded vote is then shared using Shamir's threshold secret sharing scheme. The shares are then send to different Collection Centres(CC). The Collection Centres are responsible for receiving and summing up the shares corresponds to each vote casted. We can set up the threshold and also set number of collection centers required. If there are p collection centers $CC_1, CC_2, \ldots, CC_p$ and a threshold $t \leqslant p$ is set so that we can get back the result from any t collection centers. This provides trust and reliability. Based on the number of collection centers and threshold set up, Shamir's scheme can be used for a threshold (t, p) secret sharing. A $t-1$ degree polynomial $Q(x)$ is constructed with

constant term representing the encoded vote value in decimal. The other coefficients are chosen randomly from the field $\mathbb{Z}_q$, where q is a prime larger than the encoded vote values and the number of participants. The shares are generated by evaluating the polynomial $Q(x)$ at p different values $x_1, x_2, \ldots, x_p$. These $x_1, x_2, \ldots, x_p$ values represent different collection centers known only to the Chief Election Officer and are kept secret. These shares are then send securely to the p collection centers. Any t of them can be used for result evaluation and verification. The shares look totally random and the collection centres have no idea regarding which secret (vote) share it is. i.e., no information about the vote casted is obtained from the share value. The share size is also same as the secret size and hence it provides information theoretical security. Once all the collection centres receives the vote share, the voting terminal is intimated to receive the next vote or it is the confirmation that the vote is registered properly.

The collection centers are responsible for summing up the shares they receive for vote tallying. Here the shares are always valid. They are generated automatically from the terminal program embedded. So there is no need to check the consistency of the shares received by the collection centres. But proper measures must be taken for the secure and error free communication between the voting terminal and collection centres. Collection centres behave as group of authorized parties. In a real time voting scenario, a single machine can act as a collection centre by maintaining database which contains collection of shares. However in this case the collection centre must be trusted. We can maintain a hierarchy of collection centres for collecting vote shares according to the geographical location which compute the local sum of shares. The local sum is then send to the top level collection centres which further add the sums of shares received from local collection centres. A separate communication module can be incorporated for the efficient and secure communication of

shares. The collection centres can also keep the shares received from each polling booth or polling booths belong to the same area as a separate entity for the computation of region wise voting details. The strategy for share maintenance, number of collection centres etc can be determined based on the requirement. The implementation issues also depend on the hierarchal structure used.

The Result Computation module is responsible for computing and declaring the final result. The final result can be obtained using Lagrange Interpolation using the sum of shares stored on collection centres. If there are p collection centres and a (t, p) threshold secret secret sharing scheme is used, then the share sum from any t collection centres can be used for computing the final result. These t shares can be used to get back a $t-1$ degree polynomial $Q(x)$ and the encoded result will be $Q(0)$. The result is then decoded by converting $Q(0)$ into binary and then separating the bits corresponds to each candidates. The decimal equivalent of the separated bits represent the total vote obtained by each candidate. Based on this, the election result can be announced with votes secured by each contesting candidate.

It is noted that the result computation cannot be performed by a collection centre. They will just keep the share sum and a hash is computed, which is then signed by using the private keys of the collection centre. During the result computation, it can be verified for the integrity and authenticity. The result declaration module, is managed by higher officials and only they know the different x values used for each collection centre during the share generation. Any t of this x values and the corresponding share sum, which is the y values, the polynomial interpolation can be done. The result computation can be done with different combination of the share sum from t different collection centres which adds reliability. The trust is maintained by the Shamir's scheme because less than t collection centres cannot get any information about

the final result. At least t collection centres have to collate to get back the result.

10.2.5 E-voting Algorithms

The following algorithm only includes the core functionality required. Additional functionalities can be added depending on the implementation requirement. Suitable hash algorithm and signature algorithm must be chosen for maintaining the integrity and authenticity. When the voting is finished, the hash of final share sum of each collection centre SCC_j can be computed using SHA(Secure Hash Algorithm) [171] and is then digitally signed by the previously issued private keys of the collection centre. The election official can verify this for integrity and authenticity. The E-voting algorithm and Result computation are given in Algorithm 10.1 and Algorithm 10.2.

10.2.6 E-voting Example

Let us assume that three people Alice, Bob and Charles are contesting in an election and there are seven voters. So the maximum vote each contestant can get is seven. Three bits are hence required for the representation of votes gained by each candidate and a total of nine bits for the representation of encoded votes corresponds to three candidates.

Let $m = 3, n = 7, V = 9\ bits$.

A sample voting scenario is given below where six voters made the vote out of seven.

Algorithm 10.1: E-Voting

Input: Vote casted by the voters
Output: Sum of the shares of the votes at each CC

Let m denote the number of candidates and n denote number of voters.
Set V equals $(\lfloor log_2 n \rfloor + 1) \times m$ bits for encoding the votes.
Choose an appropriate field $\mathbb{Z}_q$.
for *each vote* $i = 1 : n$ **do**
    enc_vote = bin_decimal(set_bit (V))
    `/* V is set according to the vote casted */`
    `/* enc_vote is the encoded vote in decimal */`
    Pick $t-1$ random numbers $a_1, a_2, a_3, \ldots, a_{t-1}$ from $\mathbb{Z}_q$
    Construct the polynomial
    $Q(x)$=enc_vote $+a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$
    for $j = 1 : p$ **do**
        Generate share $V_{ij} = Q(j)$
        `/* where` V_{ij} `is the` j^{th} `share of` i^{th} `vote */`
        Send the share V_{ij} to C_j^{th} collection centre
        through a secure communication channel
    end
    for *each Collection Centre* $j = 1 : p$ **do**
        Sum of shares $SCC_j = SCC_j + V_{ij}$
    end
end

The votes are encoded as shown in Table 10.1 corresponds to each contesting candidate. Let us choose a field $\mathbb{Z}_{257}$. We have considered a $(2, 3)$ secret sharing scheme, where sum of the shares from any two collection centre can be used to reconstruct the secret result. Every time a vote is casted, a random polynomial $Q(x)$ of degree 1 are constructed with constant term as the encoded vote and the other coefficient are chosen randomly from $\mathbb{Z}_{257}$. Generate the three shares CC_1, CC_2 and CC_3 with x_i's as 1, 2 and 3 respectively. It is noted that the shares are random

Algorithm 10.2: Result Computation

Input: Share sum of the votes from collection centres

Output: Votes obtained by each candidate

1 **for** *each randomly chosen t Collection Centre* $j = 1 : t$ **do**
2 retrieve SCC_j
3 **end**
4 Interpolate using SCC_j and corresponding x_i values to obtain the polynomial Q(x)
5 Obtain the secret value $Q(0)$.
6 Decode $Q(0)$ and obtain the binary representation.
7 Each $(\lfloor log_2 n \rfloor + 1)$ bits will represent each candidates vote.
8 Publish the final results i.e., votes obtained by each candidates based on the encoded values.

irrespective of the encoded vote. So the collection centre cannot derive any information about the secret from the shares they receive. The collection centre also compute the share sum SCC_j from the shares they receive. Table 10.2 shows the random polynomials constructed, the corresponding shares generated and also the share sum in the sample run of the algorithm corresponds to $(2, 3)$ secret sharing scheme.

The election result can be computed from the sum of shares SCC_j maintained by each collection centre using Lagrange interpolation. The polynomial $Q(x)$ can be obtained using any two shares in the example using the Lagrange Interpolation formula as follows.

$$Q(x) = SCC_1.\frac{(x - x_2)}{(x_1 - x_2)} + SCC_2.\frac{(x - x_1)}{(x_2 - x_1)}$$

The final result depends on $Q(0)$ which is easily obtained by

$$Q(0) = SCC_1.\frac{(x_2)}{(x_2 - x_1)} + SCC_2.\frac{(x_1)}{(x_1 - x_2)}$$

Table 10.1: Example E-voting

Vote	Alice	Bob	Charles	Encoding Vote	
				Binary	Decimal
1	✓			001000000	64
2		✓		000001000	8
3		✓		000001000	8
4	✓			001000000	64
5			✓	000000001	1
6	✓			001000000	64

Table 10.2: Vote Sharing

Vote	enc _vote	q(x)	Shares		
			CC_1	CC_2	CC_3
1	64	233x+64	40	16	249
2	8	157x+8	165	65	222
3	8	78x+8	86	164	242
4	64	255x+64	62	60	58
5	1	217x+1	218	178	138
6	64	124.x+64	188	55	179
share sum SCC_j			245	24	60

Computation of results using different combination of shares $SCC_1 : SCC_2, SCC_1 : SCC_3$ and $SCC_2 : SCC_3$ are shown in equation 10.1,10.2 and 10.3. The operations are carried out in $\mathbb{Z}_{257}$. It is noted that the reconstructed values are consistent.

$$Q(0) = 245.\frac{2}{2-1} + 24.\frac{1}{1-2} = 209 \tag{10.1}$$

$$Q(0) = 245.\frac{3}{3-1} + 60.\frac{1}{1-3} = 209 \tag{10.2}$$

$$Q(0) = 24.\frac{3}{3-2} + 60.\frac{2}{2-3} = 209 \tag{10.3}$$

The final result can be obtained by decoding the reconstructed result

209 into binary. It is noted that 3 bits will represent vote secured by each candidate.

$$\text{decoded vote}: \ 011, 010, 001$$

The result can be published based on the decoded vote values which is shown in Table 10.3.

Table 10.3: E-voting Result

Candidate	**Votes Secured**
Alice	3
Bob	2
Charles	1

10.2.7 Implementation

We have done a preliminary implementation of this scheme using Java. The architecture of the developed system is shown in Figure 10.1. The proposed system focus on the generation of secret shares, secure distribution of shares and secure computation of votes obtained for each candidate. The detailed architecture is shown in Figure 10.2. Polling Station provides the interface for voting purpose. A polling station may contain many voting machines. It has a voting panel which contains the list of all candidates and their party symbols. Voting panel is loaded with this candidates information from a setup file which is managed by the Chief Election Commissioner. The vote casted by a voter is given to share generator module, which contains the encoding and Shamir share generator module. The Encoding module will encode the vote using the bitwise encoding algorithm as explained before. Share Generator uses Shamir's secret sharing scheme for generating shares of the encoded vote. The number of shares generated is based on number of collection centres. This provides both security and trust which

is implemented using Shamir's (t, n) threshold scheme in which any t shares of total n shares can be used for reconstructing the original vote casted.

The shares generated in the share generator module is sent to the collection centres through the Communication Server which manages the communication and coordination among all the other modules. This module handles Voting Machine Manager, Communication Manager and a Collection Centre Manager. Chief Election Commissioner module is working in an administrative role, which manages the other modules. The Voting Machine Handler manages a set up file containing the list of candidates and their party symbols. Any modification made in the set up file will be reflected in the voting panel interface. The Collection Centre Handler manages the collection centres. For reconstructing the sum of votes for each candidate, t collection centres need to be selected randomly based on (t, n) threshold scheme. Collection Centre Handler randomly selects any of the t collection centres during the reconstruction phase. Authentication of Collection centres is also managed by Collection Centre Handler. The Result Analyzer computes and declare the result using the share sum obtained from different collection centre.

Collection Centre(CC) manages all the shares and provides a local database for holding the shares. Usually a group of authorized parties behave as collection centres. Each collection centre will be having a local database which receives one share for every vote casted. Number of collection centres (n) depend on the number of shares generated for each vote, which in turn depends on the chosen threshold (t, n) scheme. Each collection centre CC_i gets i^{th} shares of all the votes. From the share of a vote casted, the collection centre cannot derive any useful information regarding the vote casted. The Computation Agent performs summation of all the shares it received in its local database and it is used as a partial sum in the multi party computation. When the Collection Centers are selected for the final result computation, the partial sum is passed to

Collection Center Handler module in the Chief Election Commissioner module. The Result Analyzer compute the result by reconstructing the encoded secret using Lagrange Interpolation. The decoding algorithm is performed then, which will reveal the individual sum of votes of each contesting candidate.

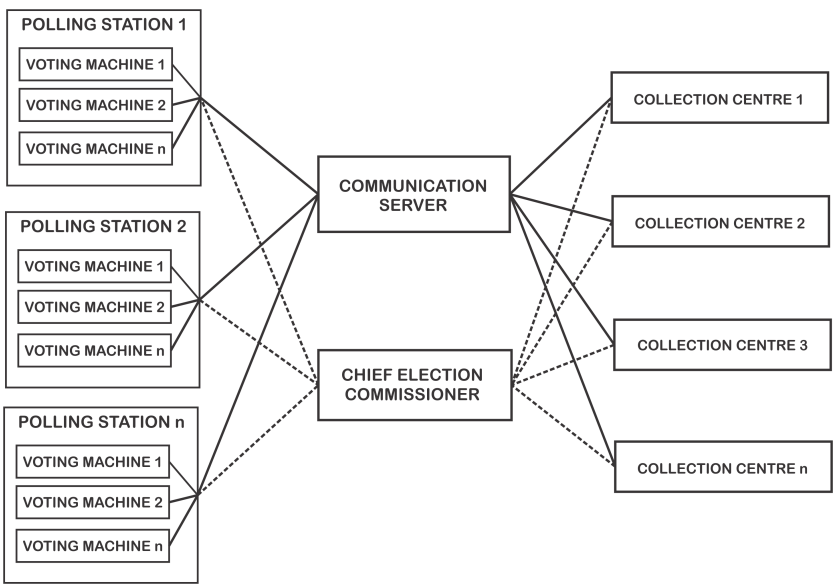

Figure 10.1: E-voting: System Architecture

Results based on 5 voters, 3 candidates and 5 collection centres are considered. The shares generated based on Shamir's $(3, 5)$ scheme are shown Table 10.4. CC_1, CC_2 and CC_3 are chosen for the computation of the result. The values of share sum SCC_1, SCC_2, SCC_3 obtained are 768, 1771 and 3284 respectively. The Result Analyzer uses these values for interpolation. The polynomial obtained is $275 + 238x^1 + 255x^2$. The constant term 275 represent the sum of votes. Decoding of this will result in 0001 0001 0011. Each 4 bit represent the individual votes obtained by candidates.

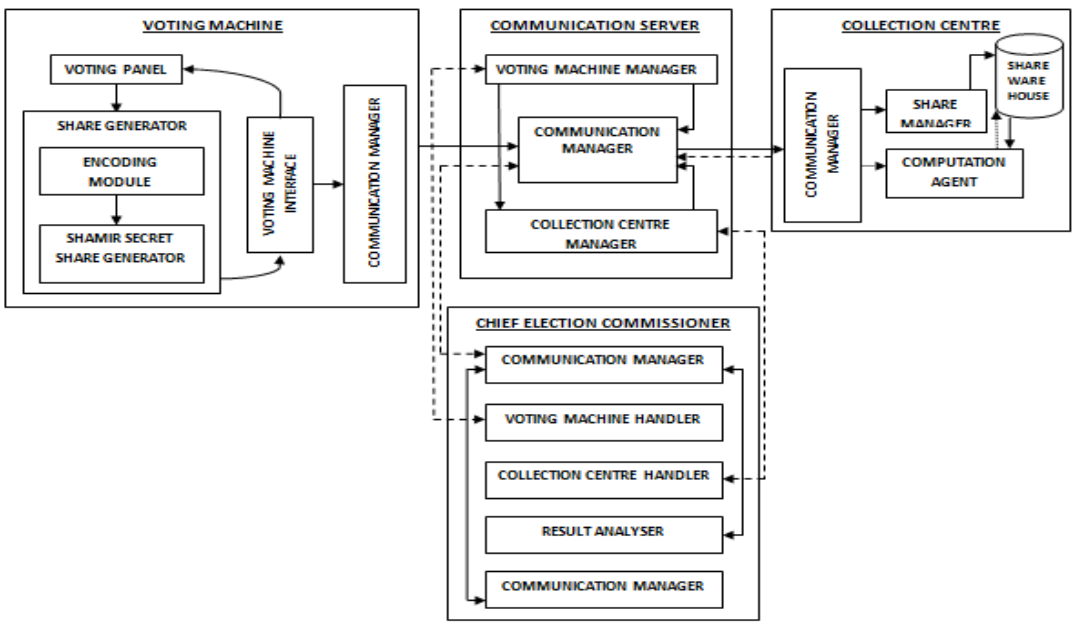

Figure 10.2: Detailed Architecture

Table 10.4: Voting System: Share Generation

Voters	CC_1	CC_2	CC_3	CC_4	CC_5
Voter1	(1,91)	(2,269)	(3,535)	(4,889)	(5,1331)
Voter2	(1,327)	(2,498)	(3,769)	(4,1140)	(5,1611)
Voter3	(1,70)	(2,251)	(3,544)	(4,949)	(5,1466)
Voter4	(1,113)	(2,278)	(3,511)	(4,812)	(5,1181)
Voter5	(1,167)	(2,475)	(3,925)	(4,1517)	(5,2251)

10.2.8 Analysis and Discussions

Security in on-line election is a challenging task. Authenticating the voter is a major challenge along with the privacy of the vote. We have considered manual authentication and proposed a modification to the existing voting

scheme which uses electronic voting machine. The voting machines are not reliable and also in certain situations more than one voting machine needs to be connected when the number of candidates are more. The proposed scheme is cost effective and also reliable.

It is noted that the proposed algorithm mentioned is simple and effective and provides privacy to the vote casted. The shares are generated by constructing a random polynomial and the share size is same as the encoded vote. The collection centres have no idea about how the votes are encoded, how many bits are used for encoding, which bits represents a particular candidate votes etc. The collection centres will receive a random value from the field $\mathbb{Z}_q$ from which no information about the secret vote can be obtained. The coalition of t untrusted collection centres can obtain the result. But they doesn't have any knowledge about the number of collection centres, the threshold used and also what is the x values assigned to each collection centre. In the example we have considered 1, 2 and 3 for simplicity, however different x values can be used and are kept secret.

Shamir's secret sharing scheme is information theoretically secure. It is perfect in the sense that no information can be obtained from less than the threshold number of participants. This adds trust to the existing E-voting scheme because the computation of the result need participation of t collection centres. The computation of the shares and the reconstruction of the final result using the share sum can be done using polynomial evaluation and interpolation. Efficient $O(n.log^2 n)$ algorithms for polynomial evaluation and interpolation are mentioned in [1] [138]. Simple quadratic algorithms are sufficient because the number of shares generated is not too large.

The encoding and decoding of the votes can also be done easily. The codes for each candidates and also the number of bits required to represent the votes depends on the number of voters and number of

contesting candidates. These setups are done by the election officials prior to the election process. The decoding of votes is a simple binary conversion which can also be done easily. The integrity of the share sum maintained by each centre is achieved by implementing a digital signature scheme. This can also be efficiently implemented using any digital signature scheme [4].

The algorithm is computationally efficient and the complexity involved depends on the share generation during the voting and the communication with the Collection Centres. The number of shares are usually small and hence the share generation using polynomial evaluation is simple. The secure communication between the voting terminal and the collection centre is more challenging. Separate communication module can be incorporated to do it efficiently. The collection centre must also be capable of handling requests from large number of voting terminals. Region wise collection centres can be incorporated to balance the load and update the top level collection centre data in a periodic manner. The result analysis needs the polynomial interpolation but is done only once and it doesn't add much complexity to the performance of the system.

In traditional elections most ideal security goals such as democracy, privacy, accuracy, fairness and verifiability are assured to a certain level given physical and administrative premises. The task of meeting the security goals is quite difficult in online elections. Another controversial pair of security properties in E-voting schemes are privacy and eligibility. It is difficult in online elections to unequivocally identify and check the credentials of a voter, while at the same time protecting the privacy of his/her vote. Computerized voting will never be used for general elections unless there is a protocol that both maintains individual privacy and prevents cheating.

A good voting system should satisfy number of generic voting principle. The authentication mechanism should ensure that only eligible

persons can vote and should not allow any one to vote more than once. The proposed method satisfies the fundamental requirement of a secure voting protocol. No one can determine for whom anyone else voted. Even the authorities will not be able to determine this because the information is not stored anywhere. For each vote casted the shares are send to all the collection centres and the partial sum is updated. The shares generated using Shamir's scheme is information theoretically secure and no information about the vote casted is obtained from the shares. The consistency of the result obtained can be verified with t different set of shares.

10.2.9 Concluding Remarks

The E-voting scheme using Shamir's secret sharing homomorphism and encoding and decoding of votes is the first proposal which helps to obtain not only the election result but also the votes gained by each candidate. The use of secret sharing homomorphism for E-voting was suggested by several authors however true or false voting mechanism is mentioned. The proposed algorithm generalize the use of secret sharing homomorphism to E-voting which provides secrecy, computational efficiency, trust and reliability. The system does not also leave any trace of the vote made by a voter.

The strong requirement of the scheme mentioned here is a secure channel for sending shares. The shares can be send through different channels to different collection centres. The intruder have to get access to t different channels for breaking the security of the scheme. For additional security, the shares can also be encrypted by using the public keys of the collection centre. There are several homomorphic encryptions which support this or ordinary encryption decryption can be used.

The system works efficiently for a moderate election with less number of voters. If the number of voters and candidates are more, the encoded vote will have a large value and the system has to chose a field of large

size. This will result in large share size, which will results in too much communication overhead. This can be avoided by breaking the encoded vote into smaller code and makes shares of it. However the complexity involved in the implementation will increase.

We have done a preliminary implementation of the scheme using Java [150]. Additional modules are incorporated as per the requirement. Another feature that can be incorporated is the implementation of digital signature scheme, which ensures integrity and authenticity of the shares. Verifiable secret sharing techniques can also be incorporated which ensures the consistency of the shares. However it slow down the system performance. We are looking for a more sophisticated implementation guaranteeing authentication using mobile phones and OTP (One Time Password) for all the users using adhar details. Instead of voting terminals every one can vote using the registered mobile phones which is our future plan. The research is still challenging in the cryptographic community to design more powerful and secure e-voting schemes.

10.3 Cheque Truncation System

10.3.1 Introduction

Cheque Truncation System (CTS) is an automatic cheque clearance system implemented by RBI. CTS uses cheque image instead of the physical cheque itself for cheque clearance. This will reduce the turn around time for cheque clearance drastically. This approach holds back the physical movement of cheque from presenting bank to the drawee bank. In CTS, digital image of the cheque is protected using standard

Some results of this section are included in the following paper.
S. R. Sreela,Binu V. P,G. Santhosh Kumar,"Establishing Security in Cheque Truncation System using Secret Image Sharing", Eighth International Conference on Computer Communication Networks (ICCN 2014), Banglore, Elsevier,ISBN :9789351072539,P-29

public key and symmetric key encryptions like RSA, triple DES etc. This involves a lot of computation overhead and key management. The security also depends on the hard mathematical problem and is only computationally secure. Information theoretically secure, secret image sharing techniques can be used in the CTS for the secure and efficient processing of cheque image. In this section, we present two simple and efficient secret image sharing schemes and a CTS based on these algorithms. In the proposed scheme, the presenting bank is acting as the dealer and the participants are the customer and the drawee bank. The dealer should generate the shares of cheque and distributes it to customer and drawee bank. The validity of the shares are important during the reconstruction process. The proposed scheme also suggests a method for cheating detection which identify any invalid shares submitted by the customers, using the hashing technique. The experimental results shows that the proposed scheme is efficient and secure compared with the existing schemes.

Cheques represent a significant segment of payment instruments in India. Cheque Truncation System (CTS) or ICS (Image Based Clearing System) in India is a project undertaken by Reserve Bank of India (RBI) for faster clearing of cheques. CTS is basically an online image-based cheque clearing system where cheque images and Magnetic Ink Character Recognition (MICR) data are captured at the collecting bank branch and transmitted electronically. Manual clearing of cheque needs human intervention and is a time consuming task. Cheque truncation [6] involves stopping the flow of the physical cheques issued by a drawer to the drawee branch. An electronic image of the cheque is sent to the drawee branch along with the relevant information like the MICR fields, date of presentation, presenting banks etc. The point of truncation is left to the discretion of the presenting bank. Thus Cheque truncation would eliminate the need to move the physical instruments across branches and

hence result in effective reduction in the time required for payment of cheques, the associated cost of transit and delays in processing etc. This will speed up the process of collection or realization of cheques and thus reduce the turn around time.

The system offers following benefits to the bank and customers. Banks can expect multiple benefits through the implementation of CTS like faster clearing cycle, better reconciliation/verification process. Besides it reduces operational risk by securing the transmission route. Reduction of manual tasks leads to reduction of errors. Customer satisfaction will be enhanced due to the reduced turn around time. Real-time tracking and visibility of the cheques, less fraudulent cases with secured transfer of images to the RBI are other possible benefits that banks may derive from this solution. For customer's CTS / ICS substantially reduces the time taken to clear the cheques as well as increases operational efficiency by cutting down on overheads involved in the physical cheque clearing process. In addition, it also offers better reconciliation and fraud prevention.

The use of the Public Key Infrastructure (PKI) ensures data authenticity, integrity and non-repudiation. This adds strength to the entire system. The presenting bank is required to affix digital signature on the images and data from the point of truncation itself. The image and data are secured using the PKI through out the entire cycle covering capture system, which include the presenting bank, the clearing house and the drawee bank. This system needs a lot of computation and overhead in key management is high. The proposed scheme avoids this by incorporating secret sharing based techniques. Two efficient schemes are proposed which are computationally secure and efficient. The secret sharing based schemes avoids the complicated encryption decryption process and also the overhead in key management. A cheating detection scheme is also proposed which avoids the use of invalid shares during the

reconstruction.

10.3.2 Related Work

CTS system is implemented by RBI to reduce the complexity of cheque processing. CTS system is implemented in India in 2010. Grid based CTS is implemented in Chennai, Delhi, Kolkata etc. Different security schemes are also applied in cheque processing. Pasupathinathan et al [166] describes privacy enhanced electronic cheque system in 2005. In 2011, Rigel Gjomemo et al [81] explains the digital cheque forgery attack on CTS. Kota et al [122] explains the method for detecting tampered cheque images in CTS using Difference Expansion based Watermarking in 2014.

The secret image sharing schemes are based on visual cryptography, number theory, information hiding theory, error diffusion technique, boolean operation etc. Most of the schemes in the literature have complicated operations involved in share generation and reconstruction. These schemes having pixel expansion and are also lossy. In the proposed application we are considering $(2, 3)$ secret image sharing schemes which can be easily implemented and also having less computational complexity. Binu et al [22] proposed efficient $(2, 3)$ secret sharing schemes which uses XOR based operation. These schemes are used here for the construction of CTS.

10.3.3 CTS Architecture

The process flow of CTS is explained below. In CTS, the presenting bank (or its branch) captures the data on the MICR band and the images of a cheque using their capture system comprising of a scanner, core banking or other application. Images and data should meet the specifications and standards prescribed for data and images. The architecture of CTS is explained in

Figure 10.3.

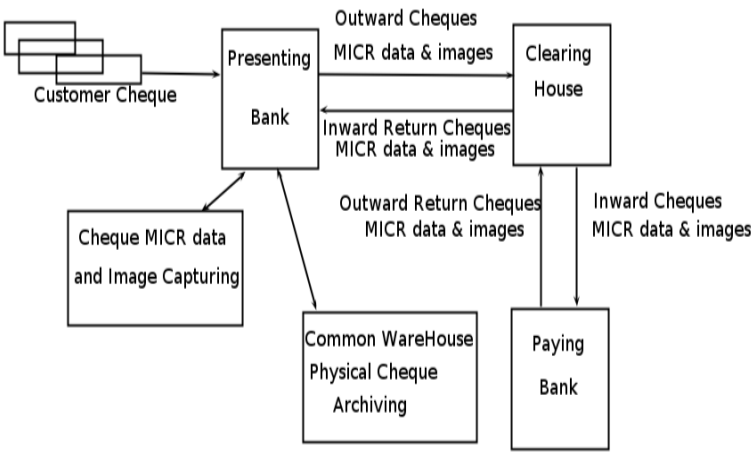

Figure 10.3: CTS Architecture

To ensure security, end-to-end Public Key Infrastructure (PKI) has been implemented in CTS for protecting data and image. The presenting bank sends the data and captured images duly signed and encrypted to the Clearing House (the central processing location) for onward transmission to the paying bank (destination or drawee bank). For the purpose of participation, the presenting and drawee banks are provided with an interface/gateway called the Clearing House Interface (CHI) that enables them to connect and transmit data and images in a secure and safe manner to the Clearing House (CH). The CTS uses public key infrastructure (PKI) like digital signature and encryption for protecting cheque images and data. The standards defined for PKI are hash algorithm SHA-1, padding algorithm, RSA asymmetric encryption with 1024 bit key length, Triple DES symmetric encryption with 168 bit key length and Certificates in X.509v3 format. Cheque image is protected

using encryption techniques. These techniques need a lot of computation and complicated key management.

10.3.4 Proposed System

The architecture of the proposed system is given in Figure 10.4. The system architecture describes how secret image sharing scheme is applied in the CTS. In this architecture, the Dealer should be the Clearing House Interface (CHI). The participants are presenting bank, clearing house (CH) and drawee bank.

In order to reduce the computation and usage of keys, cheque image can be protected using secret image sharing technique. In the proposed system, two secret image sharing methods are proposed for protecting cheque images. Cheating occurs, if any one of the participant do malpractice on the share. Cheating detection is a major requirement in critical applications. Cheating detection and identification scheme based on number theory and hash function is incorporated in the scheme.

Threshold $(2, 3)$ secret sharing scheme is used for implementing security in CTS. Presenting bank captures cheque MICR data and images using specially designed scanners. The CHI generates the shares and distribute them to the presenting bank, clearing house and drawee bank. Customer should use the share to get the information of the processing cheque through online. Drawee bank should reconstruct the cheque image using the share from the CH and share from CHI of presenting bank. Drawee bank cannot reconstruct the cheque image using its own share. It is noted that only shares are communicated through different channels in one clearing cycle. To implement security in CTS any one of the secret image sharing scheme mentioned can be used.

The important steps involved in the proposed CTS using secret image sharing are as follows:

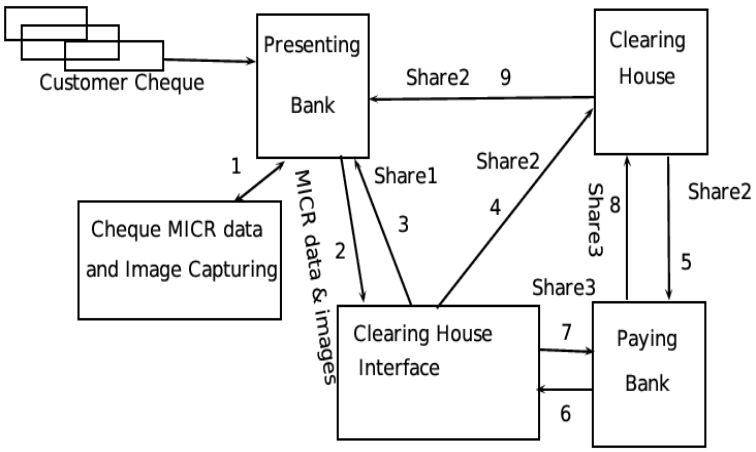

Figure 10.4: System Architecture

- Customer submits the cheque to the presenting bank and the capture system captures the MICR data and image of cheque.
- Send the data and image to the presenting Clearing House Interface(CHI). Presenting CHI provide security to the cheque image using $(2, 3)$ secret image sharing scheme.
- Send first share of the cheque image(SC_1) to the presenting bank. This share is used for authentication and for viewing the details of cheque processing.
- Send second share of the cheque image (SC_2) to the Clearing House.
- The Clearing House send data and one share of the cheque image (SC_2) to the drawee bank (paying bank).
- The drawee bank request another share of the cheque image from the presenting bank through receiving CHI.

- The presenting bank submit third share (SC_3) to the drawee bank through CHI.
- The paying bank reconstructs the cheque image using shares SC_2 and SC_3.
- Bank process the cheque using image processing algorithm.
- Send data and shares to the presenting bank through CH for further processing.

10.3.5 Partition Scheme

A simple and efficient $(2, 3)$ scheme can be implemented based on this technique. In this scheme, three shares are created and the original image is reconstructed using at least two shares. Less than two shares will not give any information about the secret image. The share images are created by dividing pixel into four bits. In this scheme, the share size is reduced to half i.e., each share image pixel is only 4 bits. Share generation is explained in Algorithm 10.3. Recovery algorithm is explained in Algorithm 10.4. The original secret image is reconstructed by using any two shares from three shares.
Consider an image matrix as

$$\begin{bmatrix} 157 & 160 & 190 & 130 \\ 89 & 255 & 224 & 192 \\ 10 & 220 & 255 & 224 \\ 64 & 128 & 192 & 255 \end{bmatrix}$$

Let the secret image pixel is 190. Its binary representation is 10111110. The share1 pixel (sc_1=6(0110)) is created using even bits. The share2 pixel (sc_2=15 (1111)) is created using odd bits. The share3 pixel (sc_3=9(1001)) is created by XORing sc_1 and sc_2. The partition scheme is applied on the above image S. The three shares SC_1, SC_2 and SC_3

obtained are as follows:

$$SC_1 = \begin{bmatrix} 7 & 0 & 6 & 0 \\ 13 & 15 & 8 & 8 \\ 0 & 14 & 15 & 8 \\ 8 & 0 & 8 & 15 \end{bmatrix}$$

$$SC_2 = \begin{bmatrix} 10 & 12 & 15 & 9 \\ 2 & 15 & 12 & 8 \\ 3 & 10 & 15 & 12 \\ 0 & 8 & 8 & 15 \end{bmatrix}$$

$$SC_3 = \begin{bmatrix} 13 & 12 & 9 & 9 \\ 15 & 0 & 4 & 0 \\ 3 & 4 & 0 & 4 \\ 8 & 8 & 0 & 0 \end{bmatrix}$$

Algorithm 10.3: Share generation-Partition

Input: Secret grayscale image S of sixe $M \times N$
Output: Share images SC_1, SC_2, SC_3

1 **for** *each pixel* $S(i,j) \in \{S(i,j) | 1 \leqslant i \leqslant M, 1 \leqslant j \leqslant N\}$ **do**
2 Pixelvalue, $pv = S(i,j)$, pv is the binary array containing the pixel intensity binary representation. Create share1 pixel $SC_1(i,j)$ using even bits of $S(i,j)$ pixel

$$SC_1(i,j) = \sum_{k=0}^{3} (pv(2k) \times 2^k)$$

3 Create share2 pixel $SC_2(i,j)$ using odd bits of $S(i,j)$ pixel

$$SC_2(i,j) = \sum_{k=0}^{3} (pv(2k+1) \times 2^k)$$

4 Create share3 $SC_3(i,j)$ pixel by XORing $SC_1(i,j)$ and $SC_2(i,j)$

$$SC_3(i,j,k) = SC_1(i,j) \oplus SC_2(i,j)$$

5 **end**
6 Output shares SC_1, SC_2 and SC_3.

Algorithm 10.4: Recovery algorithm-Partition

Input: Share images SC_1, SC_2, SC_3
Output: Reconstructed Secret image S from different shares

`/* Reconstruction of secret image from` SC_1 `and` SC_2 `*/`

for *each pixel values in* SC_1 *and* SC_2 **do**
    $S(i,j)$ is obtained by intermixing bits of $SC_1(i,j)$ and $SC_2(i,j)$ in even and odd positions respectively.
end
Output image S.

`/* Reconstruction of secret image from` SC_1 `and` SC_3 `*/`

for *each pixel values in* SC_1 *and* SC_3 **do**
    $b = SC_1(i,j) \oplus SC_3(i,j)$.
    $S(i,j)$ is obtained by intermixing bits of $SC_1(i,j)$ and b in even and odd positions respectively.
end
Output image S.

`/* Reconstruction of secret image from` SC_2 `and` SC_3 `*/`

for *each pixel values in* SC_2 *and* SC_3 **do**
    $b = SC_2(i,j) \oplus SC_3(i,j)$.
    $S(i,j)$ is obtained by intermixing bits of b and $SC_2(i,j)$ in even and odd positions respectively.
end
Output image S.

In this scheme, the size of the share is half the size of the original image. The number of bits for representing a pixel in each share is only 4 bits. If the $M \times N$ secret gray scale image has a size of $8 \times M \times N$ bits, then the size of the share is only $4 \times M \times N$ bits. So the storage space of the share is reduced. The quality of the reconstructed image is same as the original image. In this scheme, there is no pixel expansion and also it is a lossless scheme.

10.3.6 XOR Based Scheme

This method use simple XOR and concatenation operation of secret image pixel with random numbers for creating shares. The scheme is ideal unlike the partition scheme. The share size is same as the secret size. This scheme is simple and easy to implement. The share generation algorithm is explained in Algorithm 10.5. Algorithms 10.6, 10.7, 10.8 describe the reconstruction of the secret image from different combination of shares.

Algorithm 10.5: Share generation-XOR Scheme

Input: MXN Secret gray scale image S.
Output: Share images SC_1, SC_2, SC_3.

1 **for** *each pixel in* S **do**

2 Let s be the pixel of the secret image S and r be a random number in 0-255.

3 s is split into s_1 and s_2 and r is split into r_1 and r_2.

4 Share1 pixel is created by concatenating $s_2 \oplus r_2$ and r_1 $sc_1 = (s_2 \oplus r_2)||r_1$

5 Share2 pixel is created by concatenating $s_1 \oplus r_1$ and r_2 $sc_2 = (s_1 \oplus r_1)||r_2$

6 Share3 pixel is created by concatenating $s_2 \oplus r_1$ and $s_1 \oplus r_2$ $sc_3 = (s_2 \oplus r_1)||(s_1 \oplus r_2)$

7 **end**

8 Output three shares $share1(SC_1), share2(SC_2), share3(SC_3)$.

Algorithm 10.6: Reconstruction using share1 and share2

Input: Share images SC_1, SC_2.
Output: Reconstructed Secret image S.

/* Original image is reconstructed from SC_1 and SC_2 by the following algorithm. */

for *each pixel* sc_1 *and* sc_2 *in* SC_1 *and* SC_2 **do**
    The share1 pixel sc_1 is divided into two equal parts sc_{11} and sc_{12}.
    The share2 pixel $sc2$ is divided into two equal parts sc_{21} and sc_{22}.
    /* The second part of the original image pixel s_2 is reconstructed by XOR-ing first part of the share1 pixel sc_{11} and second part of the share2 pixel sc_{22}. */

$$s_2 = sc_{11} \oplus sc_{22}$$

    /* The first part of the original image pixel s_1 is reconstructed by XOR-ing second part of the share1 pixel sc_{12} and first part of the share2 pixel sc_{21}. */

$$s_1 = sc_{12} \oplus sc_{21}$$

    /* The original image pixel s is obtained by concatenating s_1 and s_2. */

$$s = s_1 || s_2$$

end
Output secret image S.

Algorithm 10.7: Reconstruction using share1 and share3

Input: Share images SC_1, SC_3.
Output: Reconstructed Secret image S.

/* Original image is reconstructed from SC_1 and SC_3 by the following algorithm. */

for *each pixel* sc_1 *and* sc_3 *in* SC_1 *and* SC_3 **do**
    The share1 pixel is divided into two equal parts sc_{11} and sc_{12}.
    The share3 pixel is divided into two equal parts sc_{31} and sc_{32}.
    /* The second part of the original image pixel s_2 is obtained by XOR-ing second part of the share1 pixel sc_{12} and first part of the share3 pixel sc_{31}. */

$$s_2 = sc_{12} \oplus sc_{31}$$

$$b = sc_{11} \oplus sc_{32}$$

    /* The first part of the original image pixel s_1 is obtained by */
    $s_1 = b \oplus s_2$
    /* Secret image pixel s is reconstructed by concatenating s_1 and s_2 */

$$s = s_1 || s_2$$

end
Output the secret image S

Algorithm 10.8: Reconstruction using share2 and share3

Input: Share images SC_2, SC_3.
Output: Reconstructed Secret image S.

`/* Original image is reconstructed from share2 and share3 by applying following steps. */`

for *each pixel* sc_2 *and* sc_3 *in* SC_2 *and* SC_3 **do**
    The share2 pixel is divided into two equal parts sc_{21} and sc_{22}.
    The share3 pixel is divided into two equal parts sc_{31} and sc_{32}.
    `/* The first part of the original image pixel` s_1 `is obtained by XOR-ing second part of the share2 pixel` sc_{22} `and second part of the share3 pixel` sc_{32} `*/`

$$s_1 = sc_{22} \oplus sc_{32}$$

$$b = sc_{21} \oplus sc_{31}$$

    `/* The second part of the original image is obtained by */`
    $s_2 = b \oplus s_1$
    `/* Secret image pixel` s `is reconstructed by combining` s_1 `and` s_2`. */`

$$s = s_1 || s_2$$

end
Output secret image S

10.3.7 Cheating detection using Hash function

A threshold scheme for secret sharing can protect a secret with high reliability and flexibility. These advantages can be achieved only when all the participants are honest. i.e., all the participants willing to pool their shadows shall always present the true ones. Cheating detection is an important issue in the secret sharing scheme. However cheater identification is more effective than cheating detection in realistic applications. If some dishonest participants exist, the other honest participants will obtain a false secret, while the cheaters may individually obtain the true secret. The use of one-way hash function along with arithmetic coding helps to design cheating detection techniques. The proposed method can be used to deterministically detect cheating and identify the cheaters, no matter how many cheaters are involved in the secret reconstruction.

Two important theorems used in cheating detection using hash function are as follows. Let a_i be the random shares of the secret data and p be the randomly generated prime number.

Theorem 10.3.1. *[219]: Let* $T = \sum_{i=1}^{n} a_i p^{i-1}$, *where* $0 \leqslant a_i < p$. *Then*

$$\left\lfloor \frac{T}{p^{j-1}} \right\rfloor \pmod{p} = a_j \tag{10.4}$$

Extended from the previous theorem, we have the following result.

Theorem 10.3.2. *[219]: Let* $T = \sum_{i=1}^{n} a_i p^{2(i-1))} + \sum_{i=1}^{n-1} c p^{2i-1}$, *where* $-p < a_i < p$ *and* $1 \leqslant c < p$. *Then*

$$\left\lfloor \frac{T}{p^{2(j-1)}} \right\rfloor \pmod{p} = a_j \pmod{p} \tag{10.5}$$

Combining this result with secret image sharing scheme, cheating detection and cheater identification is incorporated. Cheating detection

and cheater identification is explained in Algorithm 10.9.

Algorithm 10.9: Cheating detection and identification using hash function

Input: Shares of the secret.

Output: Display the cheater details.

1 Dealer generates the shares for cheque image using secret image sharing algorithm. He also generates public parameters T and p as in the following steps.

2 Choose a one-way function $h(.)$ and a prime number p such that $h(.) < p$.

3 Generates hash value of share image using hash function.

4 Compute $T = \sum_{i=1}^{n} h(SC_i)p^{2(i-1)} + \sum_{i=1}^{n-1} cp^{2i-1}$ where c is a positive constant randomly chosen over $GF(p)$.

5 Publish T and p.

6 Dealer distributes shadow SC_i to participants U_i, for $i = 1, 2, \ldots, n$.

`/* In the receiver side, cheating detection and cheater identification can easily be achieved by applying the following procedure. */`

7 Participants U_j present their shadows SC'_j.

8 Compute $T' = \sum_{U_j} h(SC'_j)p^{2(i-1)}$. For each $U_j \in G$, check

$$\left\lfloor \frac{T - T'}{p^{2(j-1)}} \right\rfloor \pmod{p} \stackrel{?}{=} 0$$

9 If the equation holds, participant U_j is honest otherwise U_j is a cheater.

The hash value of the share image is generated using the content of the image. In the secret image sharing, any simple change in the share is treated as a cheating. Any change in the image is reflected in the hash value of image. We use the hash generation method using content of the image. The cheque processing needs lot of image processing algorithm for the on-line manipulation. The application build can replace the existing

(a) cheque image

(b) Share1

(c) Share2

(d) Share3

(e) Reconstructed cheque image

Figure 10.5: Result of XOR based scheme

CTS system of RBI. We are exploring more details of this implementation in the future enhancement.

10.3.8 Experimental Results

The CTS system is implemented in Java. The running time of the algorithms depends on the size of the image. Sharing and reconstruction of a pixel value will take constant time. Thus the running time is in the order of image size. The experimental result obtained for XOR based scheme using the 500×225 gray scale cheque image is shown in Figure 10.5. The reconstructed image has the same quality as original image. This algorithm is also useful for color images. In this case, the algorithm have to be applied on each channel (Red, Blue and Green) separately.

The mean square error (MSE) is used to measure the difference between original (I) and recovered image(I') of size $M \times N$ and is calculated by using

the equation

$$MSE = \frac{1}{MN} \sum_{i=1,M} \sum_{j=1,N} (I'(i,j) - I(i,j))^2$$

The MSE between original and recovered image is 0.

In the cheating detection phase, the hash value of the share images are calculated in the sender side. In the receiver the value of T' is computed. If the remainder is zero, the cheating does not occur in the shares of the cheque image. If the cheating does not occur in the shares, the cheque image is reconstructed from the shares. Otherwise the drawee bank request for the correct shares from the participants.

10.3.9 Conclusions

Cheque Truncation System accelerates the process of collection of cheques resulting in better service to customers, reduces the scope for clearing-related frauds or loss of instruments in transit, lowers the cost of collection of cheques and removes reconciliation related and logistics related problems. This will add a lot of benefits to the cheque clearance system. In this chapter, two secret image sharing schemes are proposed for providing security to the cheque image in the CTS. The proposed partition scheme is simple and efficient but it is not ideal. It can be used in low storage device where memory become a constraint. The share size is only half of the original image and it is a lossless scheme. XOR scheme have the properties such as no pixel expansion and it is also a lossless scheme. The scheme is also ideal and can be implemented very easily.

The experimental result shows that the proposed system provides better security and efficiency in Cheque Truncation System. The operations involved are simple XOR and it also avoids the complicated encryption decryption operations which are time consuming. The secret image sharing scheme doesn't need any key management and the integrity

of the shares are maintained with simple hash function. The shares are also verified with the help of public parameters. We are looking forward for improved cheque processing using advanced image processing technique which helps in automatic cheque processing. The operational efficiency, speed, accuracy, security, integrity and authentication are the major design objectives.

Chapter 11

Summary and Future Directions

In this chapter, we summarize our contribution in this thesis, draw several useful inferences and suggest several problems for future investigations.

11.1 Brief Summary

Secret sharing technique is an active area of research from 1979 after Shamir and Blackley came up independently with the idea of threshold scheme. The area is really vast and the mathematical foundation is really fascinating. The secret sharing have found several useful applications in modern cryptology. Stinson et al [206] maintains a bibliography of the important contributions in this area.

We have done a detailed review of the threshold and generalized secret sharing schemes. This helps in the thorough understanding of the existing schemes and their drawbacks. Development of application specific schemes are our major objective. Simple and efficient schemes are developed using number theoretic techniques and XOR operation. The $(2, 3)$ and $(2, 4)$

threshold schemes are suitable for the distributed data storage. Space efficient and ideal schemes are considered and the application areas of these schemes are also explored. Secret sharing schemes corresponds to a generalized monotone increasing access structure is explored. We have used the cumulative array for secret sharing scheme with general access structure. An efficient (n, n) threshold secret sharing scheme using POB is then combined with cumulative array for the implementation of generalized access structure based secret sharing. This scheme is space efficient and also simple XOR operation can be used to reconstruct the secret. Extended capabilities of the secret sharing schemes are then studied and evaluated. Verifiability, cheating detection and cheater identification are the major capabilities analyzed. Several existing schemes are analyzed in this regard. We have included these capabilities in the proposed secret sharing schemes.

Development of multi secret sharing schemes is another major achievement in this dissertation. There are several existing multi secret sharing schemes realizing the threshold and generalized access structure. We have done a detailed investigation and comparative study of the existing multi secret sharing scheme realizing the general access structure. A scheme with general access structure is then developed to share multi secret having the capability to detect cheaters. The scheme is simple and easy to implement. The scheme is analyzed for security and is found strong for sharing multi secrets.

We investigated the use of elliptic curve and pairing in multi secret sharing. The basis of elliptic curve and pairing is studied in depth and then we looked into the secret sharing schemes based on them. There are not much proposals for multi secret sharing based on elliptic curve. We have developed two schemes for multi secret sharing based on elliptic curve and pairing. One scheme is based on point sharing technique and self pairing. This scheme realize a threshold access structure. Share

verification, cheating detection and cheater identification is also incorporated. We have also done an implementation of the above scheme using SAGE and Python for validating it. Another scheme we have developed is based on elliptic curve and bilinear pairing for realizing the general access structure. In this the secret shares are chosen by the participant itself and are kept secret. Hence the same share can be used to reconstruct different secrets and it is a multi use scheme. During the reconstruction phase the combiner can also check the validity of the shares. Pairing technique is used for cheating detection and identification of the cheaters. The scheme is easy to implement compared with other general access structure based multi secret sharing scheme using elliptic curve and pairing.

Finally we have given a theoretical frame work and also the implementation of two prominent applications of secret sharing. These applications are in preliminary stages and under revision to include more sophisticated features. Secret sharing homomorphism and their application to e-voting is suggested by different authors. However a coding scheme by which the vote gained by each contesting candidate can be efficiently obtained using the proposed scheme. The scheme is very simple and easy to implement. Another application called CTS (Cheque Truncation System) in which the simple secret sharing schemes developed based on number theory and XOR operations are incorporated for the efficient implementation, which replaces the existing encryption based implementation of CTS by RBI.

The following are the summary of the major contributions

- Development of simple and easy to implement scheme based on number theory and XOR operations. These schemes are suitable for distributed data storage and secret image sharing.
- Development of a generalized secret sharing scheme using POB.

- Study of multi secret sharing schemes and extended capabilities that can be incorporated to build secure secret sharing schemes.
- Development of a multi secret sharing scheme with general access structure, which is easy to implement and also having cheating detection and cheater identification capability. Discrete logarithm problem and Shamir's scheme are the building blocks.
- Investigation of elliptic curve and pairing in the development of secure secret sharing schemes.
- Developed a general access structure based multi secret sharing scheme using elliptic curve and bilinear pairing.
- Developed a threshold multi secret sharing scheme using elliptic curve and self pairing. Implementation of the scheme is done using SAGE and Python.
- An e-voting application is developed with each contesting candidate votes are easily obtained by using simple encoding and decoding of votes and secret sharing homomorphism.
- Development of a Cheque Truncation System using simple and easy to implement secret sharing schemes using XOR operations.

11.2 Future Directions

There are several open problems still exist in the area of secret sharing. In this section we provide some future enhancement in purview of the thesis viewpoint and also some future directions that are beyond this work's viewpoint.

- It is always better to develop more efficient secret sharing scheme which can be applied in a particular application area. We have developed simple and easy to implement $(2, 3)$ and $(2, 4)$ threshold schemes using number theory and also using XOR operations. XOR based schemes are gaining more attention because the shares can be generated and reconstructed with simple XOR operation. A simple and perfect (t, n) threshold scheme using XOR is still a challenge.

- POB system is used for the development of threshold scheme. We have used the (n, n) POB scheme for the generalized secret sharing using cumulative arrays. The efficiency of POB system can be further explored in the area of secret image sharing and secure distribution of data.

- The coding theory is a good choice for the development of robust secret sharing schemes. There are codes with specific property, which can be used for the development of threshold as well as generalized secret sharing schemes. Robust schemes can be developed using coding theory techniques.

- Multi secret sharing is gaining more importance when the data are outsourced in cloud storage. Users want to access different documents, which are encrypted with different keys using the same secret key. Most of the multi secret sharing scheme uses a public notice board. Multi secret sharing with each participant holds only a single share and also less number of public parameters are the major design criteria. Additional capabilities can also be considered.

- The use of elliptic curve and pairing is not much explored in the area of multi secret sharing. It is found that pairing based constructions provides more security and validity. There is an opportunity to find

more secure and reliable multi secret sharing schemes using elliptic curve and pairing.

- We have developed a threshold multi secret sharing scheme using self pairing. The use of elliptic curve and self pairing can be further explored to develop secret sharing schemes with more generalized access structure.
- SAGE provides extensive support for handling elliptic curve functions. Security application development using Python and SAGE is a good choice. More application or packages can be designed and built using these open source tools.
- There are several application areas, where the secret sharing technique can be applied. We have considered secret image sharing and secure multi party computation. Broadcast encryption, attribute based encryption, access control, generalized oblivious transfer etc are some of the new areas where secret sharing techniques are used.
- We have considered only manual verification of the identity of the voter, which needs to be automated in a secure way in the e-voting scheme developed.
- In the CTS system developed, the entire cheque processing can be automated which needs hand written character recognition and digit reorganization etc. The application developed is in preliminary stage, which needs further enhancement.

Appendix A

List of Notations

$A \subseteq B$	A is a subset of B.
$A \subset B$	A is a subset of B and $A \neq B$.
$\cup$	The set union.
$\cap$	The set intersection.
$\setminus$	The set difference.
$\mathcal{P}(A)$	The power set of A.
$\bar{A}$	The compliment os set with respect to a superset.
$\|X\|$	The cardinality of the set.
$\emptyset$	The empty set.
$\times$	The Cartesian product.
X^n	$X \times X \times \cdots \times X$
$\mathbb{Z}$	The set of integers.
$\mathbb{N}$	The set of natural numbers.
$\mathbb{R}$	The field of real numbers.
$\mathbb{Q}$	The field of rational numbers.
$\mathbb{C}$	The field of complex numbers.
$\mathbb{F}_p$	The finite filed $\mathbb{Z}/p\mathbb{Z}$.
$\mathbb{GF}_q$	The Galois field of order q, where q is a prime power.

gcd	Greatest common divisor
$(a_1, \ldots, a_n)$	The greatest common divisors of the integers $a_1, \ldots, a_n$.
$[a_1, \ldots, a_n]$	The least common multiple of integers $a_1, \ldots, a_n$.
$\mathbb{Z}_m$	The set $\{0, 1, \ldots, m-1\}$, for some $m \geqslant 1$.
$\mathbb{Z}_m^*$	The set $\{a \in \mathbb{Z}_m \mid (a, m) = 1\}$
$\mathbb{Z}/m\mathbb{Z}$	The ring of integers modulo m.
$(\mathbb{Z}/m\mathbb{Z})*$	The group of units in $\mathbb{Z}/m\mathbb{Z}$.
$b \mid a$	b divides a.
$b \nmid a$	b does not divide a.
$a \mod b$	The remainder of the integer division of a by b.
$n!$	n factorial $1 \times 2 \times \cdots \times n$
$a \equiv b(\mod m)$	a and b are congruent modulo m.
$a^{-1} \mod n$	Multiplicative inverse of a modulo m for some $a \in \mathbb{Z}_m^*$
$\phi(n)$	Euler's totient function.
$\|n\|$	The bit length of n
$\oplus$	The XOR operation.
$ord(m)$	The order of an element.
g	The generator of the group.
g^x	Exponent of g in a group.
$O(n)$	The time complexity of an algorithm,
$log_b a$	The discrete logarithm of a to the base b.
$\mathcal{F}_p^d$	Finite field with p^d elements.
$\binom{n}{r}$	Combinatorial symbol n choose r.
$\left(\frac{a}{p}\right)$	The Legendre symbol of a modulo p.
P_r	Probability function.
$P_r(F\|E)$	Conditional probability of F on E.
$H(X)$	The entropy of the random variable X.
$H(X\|Y)$	The conditional entropy of the random variable X given Y.
$e_k(P)$	Encryption based on key k.
$d_k(C)$	Decryption based on key k.

H	Hash function.
$\oplus$	Addition on elliptic curve points.
$\mathcal{O}$	Point at infinity on elliptic curve.
$\triangle_E$	Discriminant of the elliptic curve E.
$E(\mathcal{F}_p)$	Points of the elliptic curve with coordinates in $\mathcal{F}_p$.
$log_P(Q)$	The elliptic curve discrete logarithm of Q with respect to P.
$E[m]$	Points of order m on elliptic curve.
D	Divisor on elliptic curve
$deg(D)$	Degree of the divisor D.
$Sum(D)$	Sum of points in the divisor.
$Div(C)$	group of divisors of curve C.
e_m	The Weil pairing on an elliptic curve.
$\tau(P,Q)$	Tate pairing on an elliptic curve.
$\hat{\tau}(P,Q)$	Modified Tate pairing on elliptic curve.
$\hat{e_m}$	Modified Weil pairing on elliptic curve.
n	Number of participants in secret sharing
t	Threshold defined in secret sharing.
$\mathcal{P}$	The set of all participants.
$\mathcal{K}$	The space of keys.
$\mathcal{M}$	The space of messages.
$\mathcal{S}$	The space of shares.
P_i, U_i	The participant or user i.
S_i	Share assigned to user i.
K	The secret.
$\mathcal{D}, D$	The Dealer.
$\mathcal{A}$	Authorized access structure.
Γ	Authorized access structure.
$\mathcal{A}_{min}, \Gamma^-$	Minimal authorized access structure.
$\bar{\mathcal{A}}, \bar{\Gamma}$	Unauthorized access structure.
$\bar{\mathcal{A}}_{max}, \bar{\Gamma}^+$	Maximal unauthorized access structure.

Appendix B

List of Publications Related to This Thesis

Part of the work presented in this thesis has been published/communicated to journals

1. Binu V. P., and A. Sreekumar."Lossless secret Image Sharing Scheme.", International Journal of Computational Intelligence and Information Security.Vol-4, No-4, P-42-48, April 2013, ISSN: 1837-7823.

2. Binu V. P., and A. Sreekumar. "An Epitome of Multi Secret Sharing Schemes for General Access Structure.", International Journal of Information Processing, 8(2), 13-28, 2014, ISSN : 0973-8215

3. Binu V. P., and A. Sreekumar."Efficient Multi Secret Sharing with Generalized Access Structures.", International Journal of Computer Applications 07/2014; 90(12). DOI:10.5120/15769-4446.

4. Binu V. P., and A. Sreekumar. "Simple and Efficient Secret Sharing Schemes for Sharing Data and Image.", International Journal of

Computer Science and Information Technologies, Vol. 6 (1), 2015, 404-409.ISSN:0975-9646.

5. Sreela S. R.,G. Santhosh Kumar, Binu V. P. "Secret Image Sharing Based Cheque Truncation System with Cheating Detection." International Journal of Information Processing, 8(4), 56-67, 2014, ISSN : 0973-8215

6. Binu V. P.,Divya G Nair, Sreekumar A."Secret Sharing Homomorphism and Secure E-voting", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 22 (2015) pp 42934-42941

7. Binu V. P.,Sreekumar A.,"Threshold Multi Secret Sharing Using Elliptic Curve and Pairing", International Journal of Information Processing, 9(4), 100-112, 2015, ISSN : 0973-8215

8. Binu V. P.,Sreekumar A.,"Secure and Efficient Secret Sharing Scheme with General Access Structures based on Elliptic Curve and Pairing", Wireless Personal Communications-Springer, ISSN: 0929-6212. DOI 10.1007/s11277-016-3619-8. (Accepted For Publication)

Part of the work include in the thesis has been presented in various National/International conferences

1. Binu V. P,Sreekumar A,"An improved Lossless Secret Image Sharing Scheme". National conference in "Security Monitoring"(NCSM-2013) on 15th & 16th February 2013,Amruta School of Arts & Science,Cochin, Kerala, India http://asaskochi.com/news/wordpress/?p=458 (**Best Paper Award**)

2. Binu V. P, Sreekumar A,"Generalized Secret Sharing using Permutation Ordered Binary System", Sapience'14 - International Conference on Security and Authentication , 27th to 28th March 2014, Sree Narayana Gurukulam College,Ernakulam, Kerala, India ISBN: 978-93-83459-32-2, http://conference.bonfring.org/conferenceproceedings.php?id=1486

3. Binu V.P.,"Secret Sharing and Applications",(presented as an invited talk),National Seminar on Algebra and Number Theory(NSANT-2014), Pavanatma College,Iduki,Kerala,India http://www.mathematicspavanatma.org/sites/default/files/Abstract.pdf

4. Nair D.G., Binu V.P., Kumar, G.S., "An Effective Private Data Storage and Retrieval System Using Secret Sharing Scheme Based on Secure Multi-party Computation", International Conference on Data Science & Engineering (ICDSE), 2014 , pp.210,214, 26-28 Aug. 2014 doi: 10.1109/ICDSE.2014.6974639.IEEEXpolre.

5. Divya G. Nair,Binu V. P,G. Santhosh Kumar,"An Improved E-Voting Scheme using Secret Sharing based Secure Multi-Party Computation", Eighth International Conference on Computer Communication Networks (ICCN 2014), Banglore, Elsevier,ISBN :9789351072539,P-17

6. S. R. Sreela,Binu V. P, G. Santhosh Kumar,"Establishing Security in Cheque Truncation System using Secret Image Sharing", Eighth International Conference on Computer Communication Networks (ICCN 2014), Banglore, Elsevier,ISBN :9789351072539,P-29

Bibliography

[1] Alfred V Aho and John E Hopcroft, *Design & analysis of computer algorithms*, Pearson Education India, 1974.

[2] Charles Asmuth and John Bloom, *A modular approach to key safeguarding*, Information Theory, IEEE Transactions on **29** (1983), no. 2, 208–210.

[3] Charles Asmuth and John Bloom, *A modular approach to key safeguarding*, IEEE transactions on information theory **29** (1983), no. 2, 208–210.

[4] Mohan Atreya, Stephen Paine, Benjamin Hammond, Stephen Wu, and Paul Starrett, *Digital signatures*, Osborne/McGraw-Hill, 2002.

[5] Li Bai and XuKai Zou, *A proactive secret sharing scheme in matrix projection method*, International Journal of Security and Networks **4** (2009), no. 4, 201–209.

[6] KS Bajwa, *Cheque truncation system*, Banking Frontiers **7** (2008), no. 3, 18.

[7] Paulo SLM Barreto, Hae Y Kim, Ben Lynn, and Michael Scott, *Efficient algorithms for pairing-based cryptosystems*, Advances in cryptologyCRYPTO 2002, Springer, 2002, pp. 354–369.

[8] Philippe Béguin and Antonella Cresti, *General short computational secret sharing schemes*, Advances in CryptologyEUROCRYPT95, Springer, 1995, pp. 194–208.

[9] Amos Beimel, *Secure schemes for secret sharing and key distribution*, Ph.D. thesis, Technion-Israel Institute of technology, Faculty of computer science, 1996.

[10] Amos Beimel, *Secret-sharing schemes: a survey*, Coding and Cryptology, Springer, 2011, pp. 11–46.

[11] Amos Beimel, Anna Gál, and Mike Paterson, *Lower bounds for monotone span programs*, Computational Complexity **6** (1996), no. 1, 29–45.

[12] Amos Beimel, Tamir Tassa, and Enav Weinreb, *Characterizing ideal weighted threshold secret sharing*, Theory of Cryptography, Springer, 2005, pp. 600–619.

[13] Amos Beimel and Enav Weinreb, *Monotone circuits for monotone weighted threshold functions*, Information Processing Letters **97** (2006), no. 1, 12–18.

[14] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson, *Completeness theorems for non-cryptographic fault-tolerant distributed computation*, Proceedings of the twentieth annual ACM symposium on Theory of computing, ACM, 1988, pp. 1–10.

[15] Josh Benaloh and Jerry Leichter, *Generalized secret sharing and monotone functions*, Advances in CryptologyCRYPTO88, Springer, 1990, pp. 27–35.

[16] Josh Benaloh and Dwight Tuinstra, *Receipt-free secret-ballot elections*, Proceedings of the twenty-sixth annual ACM symposium on Theory of computing, ACM, 1994, pp. 544–553.

[17] Josh Cohen Benaloh, *Secret sharing homomorphisms: Keeping shares of a secret secret*, Advances in CryptologyCRYPTO86, Springer, 1987, pp. 251–260.

[18] Josh Daniel Cohen Benaloh, *Verifiable secret-ballot elections*, Yale University. Department of Computer Science, 1987.

[19] Shimshon Berkovits, *How to broadcast a secret*, Advances in CryptologyEUROCRYPT91, Springer, 1991, pp. 535–541.

[20] Michael Bertilsson and Ingemar Ingemarsson, *A construction of practical secret sharing schemes using linear block codes*, Advances in CryptologyAUSCRYPT'92, Springer, 1993, pp. 67–79.

[21] John Bethencourt, Amit Sahai, and Brent Waters, *Ciphertext-policy attribute-based encryption*, Security and Privacy, 2007. SP'07. IEEE Symposium on, IEEE, 2007, pp. 321–334.

[22] VP Binu and A Sreekumar, *Simple and efficient secret sharing schemes for sharing data and image*, arXiv preprint arXiv:1502.07475 (2015).

[23] Ian F Blake, Gadiel Seroussi, and Nigel Smart, *Elliptic curves in cryptography*, vol. 265, Cambridge university press, 1999.

[24] George Robert Blakley et al., *Safeguarding cryptographic keys*, Proceedings of the national computer conference, vol. 48, 1979, pp. 313–317.

[25] George Robert Blakley and Catherine Meadows, *Security of ramp schemes*, Advances in Cryptology, Springer, 1985, pp. 242–268.

[26] RG Blakley and Grigorii Anatol'evich Kabatiansky, *Generalized ideal secret-sharing schemes and matroids*, Problemy Peredachi Informatsii **33** (1997), no. 3, 102–110.

[27] Carlo Blundo, Alfredo De Santis, Douglas R. Stinson, and Ugo Vaccaro, *Graph decompositions and secret sharing schemes*, Journal of Cryptology **8** (1995), no. 1, 39–64.

[28] Carlo Blundo, Alfredo De Santis, and Ugo Vaccaro, *Efficient sharing of many secrets*, STACS 93, Springer, 1993, pp. 692–703.

[29] Dan Boneh, *The decision diffie-hellman problem*, Algorithmic number theory, Springer, 1998, pp. 48–63.

[30] Dan Boneh and Matt Franklin, *Identity-based encryption from the weil pairing*, Advances in CryptologyCRYPTO 2001, Springer, 2001, pp. 213–229.

[31] Fabrice Boudot and Jacques Traoré, *Efficient publicly verifiable secret sharing schemes with fast or delayed recovery*, Information and Communication Security, Springer, 1999, pp. 87–102.

[32] Colin Boyd, *A new multiple key cipher and an improved voting scheme*, Advances in CryptologyEUROCRYPT89, Springer, 1990, pp. 617–625.

[33] An Braeken, Svetla Nikova, and Ventzislav Nikov, *On cheating immune secret sharing.*, IACR Cryptology ePrint Archive **2004** (2004), 200.

[34] Ernest F Brickell, *Some ideal secret sharing schemes*, Journal of Combinatorial Mathematics and Combinatorial Computing **9** (1989), no. 2, 105–113.

[35] Ernest F Brickell, *Some ideal secret sharing schemes*, Advances in CryptologyEUROCRYPT89, Springer, 1990, pp. 468–475.

[36] Ernest F Brickell and Daniel M Davenport, *On the classification of ideal secret sharing schemes*, Journal of Cryptology **4** (1991), no. 2, 123–134.

[37] Ernest F Brickell and Douglas R Stinson, *The detection of cheaters in threshold schemes*, Proceedings on Advances in cryptology, Springer-Verlag New York, Inc., 1990, pp. 564–577.

[38] Ernest F. Brickell and Douglas R. Stinson, *Some improved bounds on the information rate of perfect secret sharing schemes*, Journal of Cryptology **5** (1992), no. 3, 153–166.

[39] W Steven Brown, *On euclid's algorithm and the computation of polynomial greatest common divisors*, Journal of the ACM (JACM) **18** (1971), no. 4, 478–504.

[40] Christian Cachin, *On-line secret sharing*, Cryptography and coding, Springer, 1995, pp. 190–198.

[41] Christian Cachin, Klaus Kursawe, Anna Lysyanskaya, and Reto Strobl, *Asynchronous verifiable secret sharing and proactive cryptosystems*, Proceedings of the 9th ACM conference on Computer and communications security, ACM, 2002, pp. 88–97.

[42] Jan L Camenisch, Jean-Marc Piveteau, and Markus A Stadler, *Blind signatures based on the discrete logarithm problem*, Advances in CryptologyEUROCRYPT'94, Springer, 1995, pp. 428–432.

[43] Renato M. Capocelli, Alfredo De Santis, Luisa Gargano, and Ugo Vaccaro, *On the size of shares for secret sharing schemes*, Journal of Cryptology **6** (1993), no. 3, 157–167.

[44] Marco Carpentieri, *A perfect threshold secret sharing scheme to identify cheaters*, Designs, Codes and Cryptography **5** (1995), no. 3, 183–187.

[45] Chao-Wen Chan and Chin-Chen Chang, *A scheme for threshold multi-secret sharing*, Applied Mathematics and Computation **166** (2005), no. 1, 1–14.

[46] C Charnes and J Pieprzyk, *Cumulative arrays and generalised shamir secret sharing schemes*, Seventeenth Annual Computer Science Conference (ACSC-17), New Zealand, vol. 16, 1994, pp. 519–528.

[47] David Chaum, *Blind signatures for untraceable payments*, Advances in cryptology, Springer, 1983, pp. 199–203.

[48] David Chaum, Claude Crépeau, and Ivan Damgard, *Multiparty unconditionally secure protocols*, Proceedings of the twentieth annual ACM symposium on Theory of computing, ACM, 1988, pp. 11–19.

[49] David Chaum and Torben Pryds Pedersen, *Wallet databases with observers*, Advances in CryptologyCRYPTO92, Springer, 1993, pp. 89–105.

[50] David L Chaum, *Untraceable electronic mail, return addresses, and digital pseudonyms*, Communications of the ACM **24** (1981), no. 2, 84–90.

[51] Chin-Ling Chen, Yu-Yi Chen, Jinn-Ke Jan, and Chih-Cheng Chen, *A secure anonymous e-voting system based on discrete logarithm problem*, Appl. Math **8** (2014), no. 5, 2571–2578.

[52] Hung-Yu Chien, JAN Jinn-Ke, and Yuh-Min Tseng, *A practical (t,n) multi-secret sharing scheme*, IEICE transactions on fundamentals of

electronics, communications and computer sciences **83** (2000), no. 12, 2762–2765.

[53] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch, *Verifiable secret sharing and achieving simultaneity in the presence of faults*, Foundations of Computer Science, 1985., 26th Annual Symposium on, IEEE, 1985, pp. 383–395.

[54] Josh D Cohen and Michael J Fischer, *A robust and verifiable cryptographically secure election scheme*, Yale University. Department of Computer Science, 1985.

[55] Ronald Cramer, Ivan Damgård, and Ueli Maurer, *General secure multi-party computation from any linear secret-sharing scheme*, Advances in CryptologyEUROCRYPT 2000, Springer, 2000, pp. 316–334.

[56] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers, *A secure and optimally efficient multi-authority election scheme*, European transactions on Telecommunications **8** (1997), no. 5, 481–490.

[57] Ivan Damgård, Mads Jurik, and Jesper Buus Nielsen, *A generalization of pailliers public-key system with applications to electronic voting*, International Journal of Information Security **9** (2010), no. 6, 371–385.

[58] Paolo DArco, Wataru Kishimoto, and Douglas R Stinson, *Properties and constraints of cheating-immune secret sharing schemes*, Discrete applied mathematics **154** (2006), no. 2, 219–233.

[59] Angsuman Das and Avishek Adhikari, *An efficient multi-use multi-secret sharing scheme based on hash function*, Applied mathematics letters **23** (2010), no. 9, 993–996.

[60] E Dawson, ES Mahmoodian, and Alan Rahilly, *Orthogonal arrays and ordered threshold schemes*, Australasian Journal of Combinatorics **8** (1993), 27–44.

[61] Massoud Hadian Dehkordi and Samaneh Mashhadi, *An efficient threshold verifiable multi-secret sharing*, Computer Standards & Interfaces **30** (2008), no. 3, 187–190.

[62] Romar dela Cruz and Huaxiong Wang, *Cheating-immune secret sharing schemes from codes and cumulative arrays*, Cryptography and Communications **5** (2013), no. 1, 67–83.

[63] Yvo Desmedt and Yair Frankel, *Shared generation of authenticators and signatures*, Advances in CryptologyCRYPTO91, Springer, 1992, pp. 457–469.

[64] Whitfield Diffie and Martin Hellman, *New directions in cryptography*, Information Theory, IEEE Transactions on **22** (1976), no. 6, 644–654.

[65] Cunsheng Ding, *Chinese remainder theorem*, World Scientific, 1996.

[66] E Knuth Donald, *The art of computer programming*, Sorting and searching **3** (1999), 426–458.

[67] Ratna Dutta, Rana Barua, and Palash Sarkar, *Pairing-based cryptographic protocols: A survey.*, IACR Cryptology ePrint Archive **2004** (2004), 64.

[68] Taher ElGamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*, Information Theory, IEEE Transactions on **31** (1985), no. 4, 469–472.

[69] Ziba Eslami and Saideh Kabiri Rad, *A new verifiable multi-secret sharing scheme based on bilinear maps*, Wireless Personal Communications **63** (2012), no. 2, 459–467.

[70] Mitra Fatemi, Reza Ghasemi, Taraneh Eghlidos, and Mohammad Reza Aref, *Efficient multistage secret sharing scheme using bilinear map*, Information Security, IET **8** (2014), no. 4, 224–229.

[71] Paul Feldman, *A practical scheme for non-interactive verifiable secret sharing*, Foundations of Computer Science, 1987., 28th Annual Symposium on, IEEE, 1987, pp. 427–438.

[72] Matthew Franklin and Moti Yung, *Communication complexity of secure computation*, Proceedings of the twenty-fourth annual ACM symposium on Theory of computing, ACM, 1992, pp. 699–710.

[73] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta, *A practical secret voting scheme for large scale elections*, Advances in CryptologyAUSCRYPT'92, Springer, 1993, pp. 244–251.

[74] Eiichiro Fujisaki and Tatsuaki Okamoto, *A practical and provably secure scheme for publicly verifiable secret sharing and its applications*, Advances in CryptologyEUROCRYPT'98, Springer, 1998, pp. 32–46.

[75] Steven D Galbraith, *Supersingular curves in cryptography*, Advances in CryptologyASIACRYPT 2001, Springer, 2001, pp. 495–513.

[76] Steven D Galbraith, Keith Harrison, and David Soldera, *Implementing the tate pairing*, Algorithmic number theory, Springer, 2002, pp. 324–337.

[77] H Ghodosi, J Pieprzyk, GR Chaudhry, and J Seberry, *How to prevent cheating in pinch's scheme*, Electronics Letters **33** (1997), no. 17, 1453–1454.

[78] Hossein Ghodosi, *Comments on harn–lins cheating detection scheme*, Designs, Codes and Cryptography **60** (2011), no. 1, 63–66.

[79] Hossein Ghodosi, Josef Pieprzyk, and Rei Safavi-Naini, *Secret sharing in multilevel and compartmented groups*, Information Security and Privacy, Springer, 1998, pp. 367–378.

[80] Hossein Ghodosi, Josef Pieprzyk, Rei Safavi-Naini, and Huaxiong Wang, *On construction of cumulative secret sharing schemes*, Information Security and Privacy, Springer, 1998, pp. 379–390.

[81] Rigel Gjomemo, Hafiz Malik, Nilesh Sumb, VN Venkatakrishnan, and Rashid Ansari, *Digital check forgery attacks on client check truncation systems*, Financial Cryptography and Data Security, Springer, 2014, pp. 3–20.

[82] Meenakshi Gnanaguruparan and Subhash Kak, *Recursive hiding of secrets in visual cryptography*, Cryptologia **26** (2002), no. 1, 68–76.

[83] Oded Goldreich, Silvio Micali, and Avi Wigderson, *How to play any mental game*, Proceedings of the nineteenth annual ACM symposium on Theory of computing, ACM, 1987, pp. 218–229.

[84] Oded Goldreich, Dana Ron, and Madhu Sudan, *Chinese remaindering with errors*, Proceedings of the thirty-first annual ACM symposium on Theory of computing, ACM, 1999, pp. 225–234.

[85] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters, *Attribute-based encryption for fine-grained access control of encrypted data*, Proceedings of the 13th ACM conference on Computer and communications security, ACM, 2006, pp. 89–98.

[86] Dimitris A Gritzalis, *Principles and requirements for a secure e-voting system*, Computers & Security **21** (2002), no. 6, 539–556.

[87] Dimitris A Gritzalis, *Secure electronic voting*, vol. 7, Springer Science & Business Media, 2012.

[88] Darrel Hankerson, Alfred J Menezes, and Scott Vanstone, *Guide to elliptic curve cryptography*, Springer Science & Business Media, 2006.

[89] Lein Harn, *Efficient sharing (broadcasting) of multiple secrets*, IEE Proceedings-Computers and Digital Techniques **142** (1995), no. 3, 237–240.

[90] Lein Harn, Miao Fuyou, and Chin-Chen Chang, *Verifiable secret sharing based on the chinese remainder theorem*, Security and Communication Networks (2013).

[91] Lein Harn and Changlu Lin, *Detection and identification of cheaters in (t, n) secret sharing scheme*, Designs, Codes and Cryptography **52** (2009), no. 1, 15–24.

[92] Lein Harn and Changlu Lin, *Authenticated group key transfer protocol based on secret sharing*, Computers, IEEE Transactions on **59** (2010), no. 6, 842–846.

[93] Jingrui He and Ed Dawson, *Multisecret-sharing scheme based on one-way function*, Electronics Letters **31** (1995), no. 2, 93–95.

[94] Somayeh Heidarvand and Jorge L Villar, *Public verifiability from pairings in secret sharing schemes*, Selected Areas in Cryptography, Springer, 2009, pp. 294–308.

[95] Javier Herranz, Alexandre Ruiz, and Germán Sáez, *New results and applications for multi-secret sharing schemes*, Designs, Codes and Cryptography (2013), 1–24.

[96] Javier Herranz, Alexandre Ruiz, and Germán Sáez, *Sharing many secrets with computational provable security*, Information Processing Letters (2013).

[97] Amir Herzberg, Stanisław Jarecki, Hugo Krawczyk, and Moti Yung, *Proactive secret sharing or: How to cope with perpetual leakage*, Advances in CryptologyCRYPT095, Springer, 1995, pp. 339–352.

[98] Martin Hirt and Kazue Sako, *Efficient receipt-free voting based on homomorphic encryption*, Advances in CryptologyEUROCRYPT 2000, Springer, 2000, pp. 539–556.

[99] Jeffrey Hoffstein, Jill Catherine Pipher, Joseph H Silverman, and Joseph H Silverman, *An introduction to mathematical cryptography*, Springer, 2008.

[100] Sun Hua and Wang Aimin, *A multi-secret sharing scheme with general access structures based on elliptic curve*, Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on, vol. 2, IEEE, 2010, pp. V2–629.

[101] Ren-Junn Hwang and Chin-Chen Chang, *An on-line secret sharing scheme for multi-secrets*, Computer Communications **21** (1998), no. 13, 1170–1176.

[102] Subariah Ibrahim, Maznah Kamat, Mazleena Salleh, and Shah Rizan Abdul Aziz, *Secure e-voting with blind signature*, Telecommunication Technology, 2003. NCTT 2003 Proceedings. 4th National Conference on, IEEE, 2003, pp. 193–197.

[103] Sorin Iftene, *General secret sharing based on determinants*, Symbolic and Numeric Algorithms for Scientific Computing, 2005. SYNASC 2005. Seventh International Symposium on, IEEE, 2005, pp. 4–pp.

[104] Sorin Iftene, *Secret sharing schemes with applications in security protocols.*, Sci. Ann. Cuza Univ. **16** (2006), 63–96.

[105] Sorin Iftene, *General secret sharing based on the chinese remainder theorem with applications in e-voting*, Electronic Notes in Theoretical Computer Science **186** (2007), 67–84.

[106] Sorin Iftene and Ioana Boureanu, *Weighted threshold secret sharing based on the chinese remainder theorem.*, Sci. Ann. Cuza Univ. **15** (2005), 161–172.

[107] Mitsuru Ito, Akira Saito, and Takao Nishizeki, *Secret sharing scheme realizing general access structure*, Electronics and Communications in Japan (Part III: Fundamental Electronic Science) **72** (1989), no. 9, 56–64.

[108] Kenneth R Iversen, *A cryptographic scheme for computerized general elections*, Advances in CryptologyCRYPTO91, Springer, 1992, pp. 405–419.

[109] Wen-Ai Jackson and Keith M Martin, *Cumulative arrays and geometric secret sharing schemes*, Advances in CryptologyAUSCRYPT'92, Springer, 1993, pp. 48–55.

[110] Wen-Ai Jackson, Keith M Martin, and Christine M OKeefe, *Multisecret threshold schemes*, Advances in CryptologyCRYPTO93, Springer, 1994, pp. 126–135.

[111] Markus Jakobsson, *A practical mix*, Advances in CryptologyEUROCRYPT'98, Springer, 1998, pp. 448–461.

[112] Stanislaw Jarecki, *Proactive secret sharing and public key cryptosystems*, Ph.D. thesis, Citeseer, 1995.

[113] Mahabir Prasad Jhanwar, *A practical (non-interactive) publicly verifiable secret sharing scheme*, Information Security Practice and Experience, Springer, 2011, pp. 273–287.

[114] Xing Xing Jia, Dao Shun Wang, and Yu Jiang Wu, *Publicly verifiable secret sharing scheme based on the chinese remainder theorem*, Applied Mechanics and Materials **278** (2013), 1945–1951.

[115] Antoine Joux, *The weil and tate pairings as building blocks for public key cryptosystems*, Algorithmic number theory, Springer, 2002, pp. 20–32.

[116] Mauricio Karchmer and Avi Wigderson, *On span programs*, Structure in Complexity Theory Conference, 1993., Proceedings of the Eighth Annual, IEEE, 1993, pp. 102–111.

[117] Ehud Karnin, Jonathan Greene, and Martin Hellman, *On secret sharing systems*, Information Theory, IEEE Transactions on **29** (1983), no. 1, 35–41.

[118] Kamer Kaya and Ali Aydın Selçuk, *A verifiable secret sharing scheme based on the chinese remainder theorem*, Progress in Cryptology-INDOCRYPT 2008, Springer, 2008, pp. 414–425.

[119] Anthony W Knapp, *Elliptic curves*, vol. 40, Princeton University Press, 1992.

[120] Donald E Knuth, *The art of computer programming (volume 2)*, 1981.

[121] Neal Koblitz, *Elliptic curve cryptosystems*, Mathematics of computation **48** (1987), no. 177, 203–209.

[122] Saranya Kota and Rajarshi Pal, *Detecting tampered cheque images in cheque truncation system using difference expansion based*

watermarking, Advance Computing Conference (IACC), 2014 IEEE International, IEEE, 2014, pp. 1041–1047.

[123] SC Kothari, *Generalized linear threshold scheme*, Advances in Cryptology, Springer, 1985, pp. 231–241.

[124] Hugo Krawczyk, *Secret sharing made short*, Advances in CryptologyCRYPTO93, Springer, 1994, pp. 136–146.

[125] Jun Kurihara, Shinsaku Kiyomoto, Kazuhide Fukushima, and Toshiaki Tanaka, *A fast (3, n)-threshold secret sharing scheme using exclusive-or operations*, IEICE transactions on fundamentals of electronics, communications and computer sciences **91** (2008), no. 1, 127–138.

[126] Jun Kurihara, Shinsaku Kiyomoto, Kazuhide Fukushima, and Toshiaki Tanaka, *A new (k, n)-threshold secret sharing scheme and its extension*, Information Security, Springer, 2008, pp. 455–470.

[127] Kaoru Kurosawa, Satoshi Obana, and Wakaha Ogata, *t-cheater identifiable (k, n) threshold secret sharing schemes*, Advances in CryptologyCRYPT095, Springer, 1995, pp. 410–423.

[128] Kaoru Kurosawa and Koji Okada, *Combinatorial interpretation of secret sharing schemes*, Advances in CryptologyASIACRYPT'94, Springer, 1995, pp. 55–64.

[129] John B Lacy, Donald P Mitchell, and William M Schell, *Cryptolib: Cryptography in software.*, USENIX Security, 1993.

[130] Leslie Lamport, *Password authentication with insecure communication*, Communications of the ACM **24** (1981), no. 11, 770–772.

[131] Serge Lang, *Elliptic functions*, Springer, 1987.

[132] Byoungcheon Lee and Kwangjo Kim, *Receipt-free electronic voting through collaboration of voter and honest verifier*, Proceeding of JW-ISC2000, Citeseer, 2000.

[133] Hyang-Sook Lee, *A self-pairing map and its applications to cryptography*, Applied Mathematics and Computation **151** (2004), no. 3, 671–678.

[134] Hung-Yu Lin and Lein Harn, *A generalized secret sharing scheme with cheater detection*, Advances in CryptologyASIACRYPT'91, Springer, 1993, pp. 149–158.

[135] Iuon Chang Lin and Chin Chen Chang, *A (t, n) threshold secret sharing system with efficient identification of cheaters*, Computing and Informatics **24** (2012), no. 5, 529–541.

[136] Chung Laung Liu, *Introduction to combinatorial mathematics*, vol. 181, McGraw-Hill New York, 1968.

[137] Duo Liu, Dongping Huang, Ping Luo, and Yiqi Dai, *New schemes for sharing points on an elliptic curve*, Computers & Mathematics with Applications **56** (2008), no. 6, 1556–1561.

[138] E Keith Lloyd, *The art of computer programming, vol. 2, seminumerical algorithms , donald e. knuth, addison-wesley, reading, mass, 1981. no. of pages: xiv+ 688. price:£ 17· 95. isbn 0 20103822 6*, Software: Practice and Experience **12** (1982), no. 9, 883–884.

[139] Shoulun Long, Josef Pieprzyk, Huaxiong Wang, and Duncan S Wong, *Generalised cumulative arrays in secret sharing*, Designs, Codes and Cryptography **40** (2006), no. 2, 191–209.

[140] Chunli Lv, Xiaoqi Jia, Lijun Tian, Jiwu Jing, and Mingli Sun, *Efficient ideal threshold secret sharing schemes based on exclusive-or operations*, Network and System Security (NSS), 2010 4th International Conference on, IEEE, 2010, pp. 136–143.

[141] F Florence Jessie MacWilliams and NJ Neil James Alexander Sloane, *The theory of error-correcting codes: Part 2*, vol. 16, Elsevier, 1977.

[142] Dahlia Malkhi, Ofer Margo, and Elan Pavlov, *E-voting without cryptography*, Financial Cryptography, Springer, 2003, pp. 1–15.

[143] James L Massey, *Minimal codewords and secret sharing*, Proceedings of the 6th Joint Swedish-Russian International Workshop on Information Theory, Citeseer, 1993, pp. 276–279.

[144] Robert J. McEliece and Dilip V. Sarwate, *On sharing secrets and reed-solomon codes*, Communications of the ACM **24** (1981), no. 9, 583–584.

[145] Alfred J Menezes, Tatsuaki Okamoto, and Scott A Vanstone, *Reducing elliptic curve logarithms to logarithms in a finite field*, Information Theory, IEEE Transactions on **39** (1993), no. 5, 1639–1646.

[146] Maurice Mignotte, *How to share a secret*, Cryptography, Springer, 1983, pp. 371–375.

[147] Victor S Miller, *Use of elliptic curves in cryptography*, Advances in CryptologyCRYPTO85 Proceedings, Springer, 1986, pp. 417–426.

[148] Victor S Miller, *The weil pairing, and its efficient calculation*, Journal of Cryptology **17** (2004), no. 4, 235–261.

[149] Paz Morillo, Carles Padró, Germán Sáez, and Jorge Luis Villar, *Weighted threshold secret sharing schemes*, Information Processing Letters **70** (1999), no. 5, 211–216.

[150] Divya G Nair, VP Binu, and G Santhosh Kumar, *An improved e-voting scheme using secret sharing based secure multi-party computation*, arXiv preprint arXiv:1502.07469 (2015).

[151] Divya G Nair, VP Binu, and G Sathish Kumar, *An effective private data storage and retrieval system using secret sharing scheme based on secure multi-party computation*, Data Science & Engineering (ICDSE), 2014 International Conference on, IEEE, 2014, pp. 210–214.

[152] Moni Naor and Adi Shamir, *Visual cryptography*, Advances in CryptologyEUROCRYPT'94, Springer, 1995, pp. 1–12.

[153] Moni Naor and Avishai Wool, *Access control and signatures via quorum secret sharing*, Parallel and Distributed Systems, IEEE Transactions on **9** (1998), no. 9, 909–922.

[154] C Andrew Neff, *A verifiable secret shuffle and its application to e-voting*, Proceedings of the 8th ACM conference on Computer and Communications Security, ACM, 2001, pp. 116–125.

[155] Ventzislav Nikov, Svetla Nikova, Bart Preneel, Joos Vandewalle, et al., *Applying general access structure to proactive secret sharing schemes.*, IACR Cryptology ePrint Archive **2002** (2002), 141.

[156] Satoshi Obana, *Almost optimum t-cheater identifiable secret sharing schemes*, Advances in Cryptology–EUROCRYPT 2011, Springer, 2011, pp. 284–302.

[157] Tatsuaki Okamoto, *Receipt-free electronic voting schemes for large scale elections*, Security Protocols, Springer, 1998, pp. 25–35.

[158] Rafail Ostrovsky and Moti Yung, *How to withstand mobile virus attacks*, Proceedings of the tenth annual ACM symposium on Principles of distributed computing, ACM, 1991, pp. 51–59.

[159] Carles Padro, Germán Sáez, and Jorge Luis Villar, *Detection of cheaters in vector space secret sharing schemes*, Designs, Codes and cryptography **16** (1999), no. 1, 75–85.

[160] Haijun Pan, Edwin Hou, and Nirwan Ansari, *Enhanced name and vote separated e-voting system: an e-voting system that ensures voter confidentiality and candidate privacy*, Security and Communication Networks **7** (2014), no. 12, 2335–2344.

[161] Liao-Jun Pang and Yu-Min Wang, *A new (t,n) multi-secret sharing scheme based on shamirs secret sharing*, Applied Mathematics and Computation **167** (2005), no. 2, 840–848.

[162] Abhishek Parakh and Subhash Kak, *A tree based recursive information hiding scheme*, Communications (ICC), 2010 IEEE International Conference on, IEEE, 2010, pp. 1–5.

[163] Abhishek Parakh and Subhash Kak, *Space efficient secret sharing for implicit data security*, Information Sciences **181** (2011), no. 2, 335–341.

[164] Choonsik Park, Kazutomo Itoh, and Kaoru Kurosawa, *Efficient anonymous channel and all/nothing election scheme*, Advances in CryptologyEUROCRYPT93, Springer, 1994, pp. 248–259.

[165] Daniel Pasailă, Vlad Alexa, and Sorin Iftene, *Cheating detection and cheater identification in crt-based secret sharing schemes*, Computing Online (2009).

[166] Vijayakrishnan Pasupathinathan, Josef Pieprzyk, and Huaxiong Wang, *Privacy enhanced electronic cheque system*, Seventh IEEE International Conference on E-Commerce Technology (CEC'05), IEEE Computer Society, 2005, pp. 431–434.

[167] Torben Pryds Pedersen, *Non-interactive and information-theoretic secure verifiable secret sharing*, Advances in CryptologyCRYPTO91, Springer, 1992, pp. 129–140.

[168] Kun Peng, Riza Aditya, Colin Boyd, Ed Dawson, and Byoungcheon Lee, *Multiplicative homomorphic e-voting*, Progress in Cryptology-INDOCRYPT 2004, Springer, 2005, pp. 61–72.

[169] RGE Pinch, *On-line multiple secret sharing*, Electronics Letters **32** (1996), no. 12, 1087–1088.

[170] Stephen C Pohlig and Martin E Hellman, *An improved algorithm for computing logarithms over¡ img src="/images/tex/125. gif" alt=" gf (p)"¿ and its cryptographic significance (corresp.)*, Information Theory, IEEE Transactions on **24** (1978), no. 1, 106–110.

[171] NF Pub, *Draft fips pub 202: Sha-3 standard: Permutation-based hash and extendable-output functions*, Federal Information Processing Standards Publication (2014).

[172] Li Qiong, Wang Zhifang, Niu Xiamu, and Sun Shenghe, *A non-interactive modular verifiable secret sharing scheme*, Communications, Circuits and Systems, 2005. Proceedings. 2005 International Conference on, vol. 1, IEEE, 2005, pp. 84–87.

[173] Michaël Quisquater, Bart Preneel, and Joos Vandewalle, *On the security of the threshold scheme based on the chinese remainder theorem*, Public Key Cryptography, Springer, 2002, pp. 199–210.

[174] Michael O Rabin, *Randomized byzantine generals*, Foundations of Computer Science, 1983., 24th Annual Symposium on, IEEE, 1983, pp. 403–409.

[175] Michael O Rabin, *Efficient dispersal of information for security, load balancing, and fault tolerance*, Journal of the ACM (JACM) **36** (1989), no. 2, 335–348.

[176] Tal Rabin and Michael Ben-Or, *Verifiable secret sharing and multiparty protocols with honest majority*, Proceedings of the twenty-first annual ACM symposium on Theory of computing, ACM, 1989, pp. 73–85.

[177] Irving S Reed and Gustave Solomon, *Polynomial codes over certain finite fields*, Journal of the Society for Industrial & Applied Mathematics **8** (1960), no. 2, 300–304.

[178] Ronald L Rivest, Adi Shamir, and Len Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM **21** (1978), no. 2, 120–126.

[179] Phillip Rogaway and Mihir Bellare, *Robust computational secret sharing and a unified account of classical secret-sharing goals*, Proceedings of the 14th ACM conference on Computer and communications security, ACM, 2007, pp. 172–184.

[180] Partha Sarathi Roy and Avishek Adhikari, *Multi-use multi-secret sharing scheme for general access structure*, Annals of the University of Craiova-Mathematics and Computer Science Series **37** (2010), no. 4, 50–57.

[181] Alexandre Ruiz and Jorge Luis Villar, *Publicly verfiable secret sharing from paillier's cryptosystem.*, WEWoRC **74** (2005), 98–108.

[182] Kazue Sako and Joe Kilian, *Secure voting using partially compatible homomorphisms*, Advances in CryptologyCRYPTO94, Springer, 1994, pp. 411–424.

[183] Kazue Sako and Joe Kilian, *Receipt-free mix-type voting scheme*, Advances in CryptologyEUROCRYPT95, Springer, 1995, pp. 393–403.

[184] Takakazu Satoh, *Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves*, Commentarii Math. Univ. Sancti Pauli **47** (1998), no. 1, 81–92.

[185] Takakazu Satoh, *The canonical lift of an ordinary elliptic curve over a finite field and its point counting*, JOURNAL-RAMANUJAN MATHEMATICAL SOCIETY **15** (2000), no. 4, 247–270.

[186] Berry Schoenmakers, *A simple publicly verifiable secret sharing scheme and its application to electronic voting*, Advances in CryptologyCRYPTO99, Springer, 1999, pp. 148–164.

[187] René Schoof, *Elliptic curves over finite fields and the computation of square roots mod* , Mathematics of computation **44** (1985), no. 170, 483–494.

[188] René Schoof, *Counting points on elliptic curves over finite fields*, Journal de théorie des nombres de Bordeaux **7** (1995), no. 1, 219–254.

[189] David A Schultz, Barbara Liskov, and Moses Liskov, *Mobile proactive secret sharing*, Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing, ACM, 2008, pp. 458–458.

[190] Adi Shamir, *How to share a secret*, Communications of the ACM **22** (1979), no. 11, 612–613.

[191] Bhavani Shankar, Kannan Srinathan, and C Pandu Rangan, *Alternative protocols for generalized oblivious transfer*, Distributed Computing and Networking, Springer, 2008, pp. 304–309.

[192] Jun Shao, *Efficient verifiable multi-secret sharing scheme based on hash function*, Information Sciences **278** (2014), 104–109.

[193] Jun Shao and Zhenfu Cao, *A new efficient (t,n) verifiable multi-secret sharing (vmss) based on ych scheme*, Applied Mathematics and Computation **168** (2005), no. 1, 135–140.

[194] Runhua Shi, Hong Zhong, and Liusheng Huang, *A (t, n)-threshold verified multi-secret sharing scheme based on ecdlp*, Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on, vol. 2, IEEE, 2007, pp. 9–13.

[195] Joseph H Silverman, *Advanced topics in the arithmetic of elliptic curves*, vol. 151, Springer Science & Business Media, 1994.

[196] Joseph H Silverman and John T Tate, *Rational points on elliptic curves*, Springer Science & Business Media, 2013.

[197] Gustavus J Simmons, *Robust shared secret schemes or'how to be sure you have the right answer even though you don't know the question'*, Congr. Numer **68** (1989), 215–248.

[198] Gustavus J Simmons, *How to (really) share a secret*, Proceedings on Advances in cryptology, Springer-Verlag New York, Inc., 1990, pp. 390–448.

[199] Gustavus J Simmons, *An introduction to shared secret and/or shared control schemes and their application*, Contemporary Cryptology: The Science of Information Integrity (1992), 441–497.

[200] Gustavus J Simmons, W Jackson, and Keith Martin, *The geometry of shared secret schemes*, Bulletin of the ICA **1** (1991), no. 2, 230–236.

[201] A Sreekumar, *Secret sharing schemes using visual cryptography*, Ph.D. thesis, Cochin University of Science and Technology, 2009.

[202] A Sreekumar and S Babu Sundar, *An efficient secret sharing scheme for n out of n scheme using pob-number system*, Hack. in 2009 (2009), 33.

[203] Markus Stadler, *Publicly verifiable secret sharing*, Advances in CryptologyEUROCRYPT96, Springer, 1996, pp. 190–199.

[204] Douglas R. Stinson, *An explication of secret sharing schemes*, Designs, Codes and Cryptography **2** (1992), no. 4, 357–390.

[205] Douglas R Stinson and Ruizhong Wei, *Unconditionally secure proactive secret sharing scheme with combinatorial structures*, Selected Areas in Cryptography, Springer, 2000, pp. 200–214.

[206] DR Stinson and Ruizhong Wei, *Bibliography on secret sharing schemes*, 2003.

[207] Hung-Min Sun, *On-line multiple secret sharing based on a one-way function*, Computer communications **22** (1999), no. 8, 745–748.

[208] Chunming Tang, Dingyi Pei, Zhuojun Liu, and Yong He, *Non-interactive and information-theoretic secure publicly verifiable secret sharing.*, IACR Cryptology ePrint Archive **2004** (2004), 201.

[209] Tamir Tassa, *Generalized oblivious transfer by secret sharing*, Designs, Codes and Cryptography **58** (2011), no. 1, 11–21.

[210] Tamir Tassa and Nira Dyn, *Multipartite secret sharing by bivariate interpolation*, Journal of Cryptology **22** (2009), no. 2, 227–258.

[211] Youliang Tian, Changgen Peng, and Jianfeng Ma, *Publicly verifiable secret sharing schemes using bilinear pairings.*, IJ Network Security **14** (2012), no. 3, 142–148.

[212] Youliang Tian, Changgen Peng, Renping Zhang, and Yuling Chen, *A practical publicly verifiable secret sharing scheme based on bilinear pairing*, Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on, IEEE, 2008, pp. 71–75.

[213] Martin Tompa and Heather Woll, *How to share a secret with cheaters*, journal of Cryptology **1** (1989), no. 3, 133–138.

[214] V Vinod, Arvind Narayanan, K Srinathan, C Pandu Rangan, and Kwangjo Kim, *On the power of computational secret sharing*, Progress in Cryptology-INDOCRYPT 2003, Springer, 2003, pp. 162–176.

[215] Daoshun Wang, Lei Zhang, Ning Ma, and Xiaobo Li, *Two secret sharing schemes based on boolean operations*, Pattern Recognition **40** (2007), no. 10, 2776–2785.

[216] Shiuh-Jeng Wang, Yuh-Ren Tsai, and Chien-Chih Shen, *Verifiable threshold scheme in multi-secret sharing distributions upon extensions of ecc*, Wireless Personal Communications **56** (2011), no. 1, 173–182.

[217] Xiuqun Wang, *A novel adaptive proactive secret sharing without a trusted party.*, IACR Cryptology ePrint Archive **2011** (2011), 241.

[218] Chen Wei, Long Xiang, Bai Yuebin, and Gao Xiaopeng, *A new dynamic threshold secret sharing scheme from bilinear maps*, Parallel Processing Workshops, 2007. ICPPW 2007. International Conference on, IEEE, 2007, pp. 19–19.

[219] T-C Wu and T-S Wu, *Cheating detection and cheater identification in secret sharing schemes*, Computers and Digital Techniques, IEE Proceedings-, vol. 142, IET, 1995, pp. 367–369.

[220] Tsu-Yang Wu and Yuh-Min Tseng, *A pairing-based publicly verifiable secret sharing scheme*, Journal of Systems Science and Complexity **24** (2011), no. 1, 186–194.

[221] Chou-Chen Yang, Ting-Yi Chang, and Min-Shiang Hwang, *A (t,n) multi-secret sharing scheme*, Applied Mathematics and Computation **151** (2004), no. 2, 483–490.

[222] Chan Yeob Yeun and Chris J Mitchell, *How to identify all cheaters in pinchs scheme*, Proceedings of JWIS98, Singapore (1998), 129–133.

[223] Futai ZHANG and Jie ZHANG, *Efficient and information-theoretical secure verifiable secret sharing over bilinear groups*, Chinese Journal of Electronics **23** (2014), no. 1.

[224] HongYu Zhang, Qianzi You, and Junxing Zhang, *A lightweight electronic voting scheme based on blind signature and kerberos mechanism*, Electronics Information and Emergency Communication (ICEIEC), 2015 5th International Conference on, IEEE, 2015, pp. 210–214.

[225] Xian-Mo Zhang and Josef Pieprzyk, *Cheating immune secret sharing*, Information and Communications Security, Springer, 2001, pp. 144–149.

[226] Jianjie Zhao, Dawu Gu, and Yong Wang, *Novel verifiable general secret sharing using weil pairing*, Web Information Systems and Mining, 2009. WISM 2009. International Conference on, IEEE, 2009, pp. 524–528.

[227] Jianjie Zhao, Jianzhong Zhang, and Rong Zhao, *A practical verifiable multi-secret sharing scheme*, Computer Standards & Interfaces **29** (2007), no. 1, 138–141.