

PROTECTED SECRET SHARING AND ITS APPLICATION TO
THRESHOLD CRYPTOGRAPHY

A THESIS IN
Computer Science

Presented to the Faculty of the University of
Missouri-Kansas City in partial fulfillment of
the requirements for the degree

MASTER OF SCIENCE

by

SPANDAN MANNAVA

B.Tech., IIITD&M Kancheepuram, 2014

Kansas City, Missouri

2016

©2016

SPANDAN MANNAVA

ALL RIGHTS RESERVED

PROTECTED SECRET SHARING AND ITS APPLICATION TO
THRESHOLD CRYPTOGRAPHY

Spandan Mannava, Candidate for the Master of Science Degree

University of Missouri-Kansas City, 2016

ABSTRACT

In the secret reconstruction of Shamir's (t,n) secret sharing scheme (SS), shares released by shareholders need to be protected otherwise, non-shareholders can also obtain the secret. Key establishment protocol can establish pairwise keys for any pair of shareholders. Then, shareholders can use these pairwise keys to protect shares in the secret reconstruction process. However, adding a key establishment in the secret reconstruction slows down the process significantly. Shamir's SS is based on a univariate polynomial. Shares generated by a bivariate polynomial enable pairwise keys to be shared between any pair of shareholders. But we proposed a new type of SS, called protected secret sharing scheme (PSS), in which shares of shareholders can not only be used to reconstruct the secret but also be used to protect the secrecy of shares in the secret reconstruction process. Thus, the recovered secret is only available to shareholders but not to non-shareholders. A basic (t,n) PSS based on a bivariate polynomial is proposed. Furthermore, we introduce to use this basic PSS in the applications of threshold cryptography. The PSS is unique since it protects the secrecy of the recovered secret in a very efficient way.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled “Protected Secret Sharing and Its Application to Threshold”, presented by Spandan Mannava, candidate for the Master of Science degree, and hereby certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Lein Harn, Ph.D, Committee Chair
School of Computing and Engineering

Vijay Kumar, Ph.D
School of Computing and Engineering

Sejun Song, Ph.D.
School of Computing and Engineering

CONTENTS

ABSTRACT	iii
GLOSSARY	ix
ACKNOWLEDGEMENTS	x
Chapter	
1. INTRODUCTION	1
1.1 Secret Sharing	1
1.2 Contributions	6
1.3 Thesis Outline	6
2. RELATED WORK	7
2.1 Basic Schemes	8
2.1.1 Shamir's Secret Sharing Scheme	8
2.1.2 Verifiable Secret Sharing Scheme	16
3. OUR SCHEME	21
3.1 Motivation	21
3.2 Contribution	22
3.2.1 Perfect Sharing Scheme (Perfect SS)	23
3.2.2 The Basic (t,n) PSS Using a Bivariate Polynomial	24
3.3 Share Generation and Authentication	25
3.3.1 Private Share Generation	25
3.3.2 Private Share Generation Protocol	25

3.3.3 Public Shares Authentication.....	25
3.4 Master Secret Generation and reconstruction.....	26
3.4.1 Master Secret Generation.....	26
3.4.2 Master Secret Generation Protocol.....	26
3.4.3 Master Secret Reconstruction.....	26
4. SECURITY ANALYSIS AND PERFORMANCE.....	28
4.1 Security Analysis.....	28
4.1.1 Inside Attack.....	28
4.1.2 Outside Attack.....	30
4.2 Performance.....	30
5. APPLICATION TO ALGORITHMS OF THRESHOLD CRYPTOGRAPHY....	32
6. CONCLUSION.....	33
6.1 Open Problems.....	34
6.2 Future Work.....	35
REFERENCES.....	36
VITA.....	41

TABLES

Table	Page
Table 2.1 Lagrange's interpolating formula	16

GLOSSARY

SSs:	Secret Sharing Scheme
PSS:	Protected Secret Sharing
VSS:	Verifiable Secret Sharing
BVSSs:	VSS using bivariate polynomial
SBVSSs:	Symmetric BVSSs
ABVSSs:	Asymmetric BVSSs
CRT:	Chinese Remainder Theorem
RSA Algorithm:	Rivest, Shamir and Adleman Algorithm
s	Secret
t	Threshold
GM	Group Manager

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor Dr. Lein Harn for his most valuable suggestions and encouragement without whom this work would have not been possible. I appreciate his generosity and valuable time that he spent to discuss and clarify my doubts that came up during this work.

I am very thankful to Dr. Sejun Song and Dr. Vijay Kumar for serving as members of my thesis committee.

I take this opportunity to offer my gratitude to my mother Mrs. Viswa Bharathi Mannava, and my family Mr. Satish Babu, Mrs. Suvi Saradha for their love, encouragement and support.

I extremely thank my friend Ms.Kavya Devineni for her support towards the Master's.

Finally, I would like to thank my academic advisor Ms. Coretta Carter, all my teachers and professors in my school and college and all my friends who have always been there for me.

CHAPTER 1

INTRODUCTION

In cryptography, a secret sharing scheme is a method for distributing a secret amongst a group of participants, each of which is allocated a share of the secret. The secret can only be reconstructed when the shares are combined. Individual shares are of no use on their own. Highly sensitive and highly important data like encryption keys, missile launch codes and numbered bank accounts, must be kept highly confidential as their exposure is highly disastrous and a single point failure might cause great loss. Secret sharing schemes are very useful for such kind of data, as they won't keep the entire secret in one place, so that both the above-mentioned problems can be overseen.

1.1 Secret Sharing

In the traditional encryption methods in which we store the secret in one place or keep duplicate of secret in multiple places. It does not avoid the single point failure in the former scenario, and the later scenario is much more dangerous as it endangers the confidentiality of the secret due to duplicates. And it also adds the up the attack vectors, increasing the chance of the secret to fall into wrong hands. Secret sharing solves these problems with arbitrarily high levels of confidentiality and reliability. Consider a bank account where there is a vault that must be opened every day. The bank employs several senior tellers, who are trusted enough to participate in the opening of the vault, but not trusted to the tent that they themselves own the combination to the vault. Vault can only be opened by all the senior teller participation but one individual cannot open it. This is one practical example to show the implementation of secret sharing.

In Cloud Computing, which is the emerging phenomena now, secret sharing is gaining more importance. Many servers had to share common resources in cloud computing. So, we distribute the secret among servers, and then reconstructed when needed by using shared key from each server. Sensor networks where the links are liable to be tapped, are secured by using secret sharing where we send the data in shares, which makes the task of eavesdropper hard. Security can be more enhanced by continuously changing the pattern of secret reconstruction.

Secret Sharing is very important in network applications. The security of operations that are happening over computer networks has become very predominant as everything is online now like payments, voting, mail etc. Bad users who may try to misuse the systems must be guarded as they might steal credit card numbers, impersonate other users, read personal mail and so on. “Reduced Trust”, is a well-known principle in the analog world, the less knowledge or power each entity has to keep a secret, the more secure is the secret. Many important files have one key, which is used to access them. If such a key is lost, due to any reason like hard disk crash, credentials not available, then all the files become inaccessible. We are facing a single point failure here and have to search for backup options which might lead to the loss of secret. While performing encryption, key is to be stored as to allow no one more power to use it, we have to ensure that the key is stored properly. To address these problems in networks, Secret Sharing is used.

Secret Sharing schemes enable some predetermined sets of parties to reconstruct a given secret. This makes it easy to store a secret information in a network such that only good subsets can reconstruct the secret and perform actions. One good example of secret sharing is that good passwords are hard to remember. A clever user can use a secret sharing scheme, to generate shares for the password and store one share in his address book, one in his bank

deposit safe, leave one secret with a friend, etc. If any day he forgets his password, he can reconstruct it easily. This is a typical example of a secure backup system.

Secret sharing involves all the shareholders to perform in secret reconstruction. But what if one shareholder is not available. What if his share is corrupted, lost or stolen? Then we again come back to the point of failure. Hence, we need to be able to reconstruct the secret even without participation of all shareholders. Then comes the point of threshold cryptography.

Consider an example where the director of a bank could generate shares for the bank's vault and hand them out to his employees. Even if he is not available, the vault can be opened, but only, when a certain number of employees do it together. That certain number is called threshold. Threshold cryptography is the art of chopping a secret into little bits, so that the secret can only be learned by possessing more than a threshold number of those bits. In the context of cloud computing, threshold cryptography is described as a highly sensitive action like decryption or signing, which is performed by a group of cooperative servers in such a way that no minority of servers are able to perform the operations themselves, nor are they able to prevent the other servers from performing the operation when it is required. A good example of an application whose security could be greatly improved with a threshold solution is a network certification authority, a trusted entity that certifies that a given public key corresponds to a given user.

The (t,n) secret sharing scheme (SS) was proposed by Shamir [1] and Blakley [2] separately in 1979. In a (t,n) SS, the dealer divides the secret into n shares and distributes each share to corresponding shareholder secretly such that (a) the secret can be recovered if there are t or more than t shares available, and (b) the secret cannot be recovered if there are fewer than t shares. Desmedt and Frankel, Pedersen, Gennaro et. al., are the major contributors

towards threshold cryptography. The (t,n) SS can be implemented by using many different mathematical tools. For example, Shamir's scheme is based on a linear polynomial, Blakely's scheme [1] is based on the geometry, Mignotte's scheme [3] and Asmuth-Bloom's scheme [4] are based on the Chinese remainder theorem (CRT). There are many research papers on the subject of SS in the literature and SS has become one of fundamental tools in secure multi-party computing. The secret reconstruction of Shamir's SS is very simple and is based on the Lagrange interpolation formula. However, in secret reconstruction, shares released by shareholders need to be protected; otherwise, non-shareholders can also obtain the secret. Key establishment protocol can establish pairwise keys for any pair of shareholders. Then, shareholders can use these pairwise keys to protect shares released by shareholders in the process. However, adding a key establishment protocol in the secret reconstruction can slow down the secret recovering process significantly.

A real time example of threshold scheme is referring to Time Magazine, May 4, 1992, control of nuclear weapons in Russia involves a two-out-of-three mechanism. In order to launch a nuclear missile, the cooperation of at least two parties out of three are needed. The three parties involved are the president, the Defense Minister, and the Defense Ministry.

Efficient threshold schemes can be very helpful in the management of cryptographic keys. In order to protect data, we can encrypt it, but in order to protect the encryption key we need a different method. The most secure key management scheme keeps the key in a single, well-guarded location. This scheme is highly unreliable since a single misfortune can make the information inaccessible. An obvious solution is to store multiple copies of the key at different locations, but this increases the danger of security breaches. By using a (t, n) threshold scheme we get a very robust key management scheme: We can recover the original key even when t

keys are used, but our opponents cannot reconstruct the key even when security breaches expose $t-1$ of the remaining t pieces. Threshold schemes are ideally suited to applications in which a group of mutually suspicious individuals with conflicting interests must cooperate. Ideally we would like the cooperation to be based on mutual consent, but the veto power this mechanism gives to each member can paralyze the activities of the group. By properly choosing the k and n parameters we can give any sufficiently large majority the authority to take some action while giving any sufficiently large minority the power to block it.

Shares in Shamir's (t,n) SS can be used to reconstruct only one secret. This is because, in the secret reconstruction, the secret and shares are revealed to all shareholders. To improve the efficiency of a SS, the threshold cryptography was first introduced by Desmedt in 1987 [5]. Threshold cryptography is the study of multiparty computation protocols for different cryptographic functions (e.g. signing or decrypting) in which each group member receives a share of a private key generated by the group manager (GM) initially and then uses the share later to jointly compute an output of the cryptographic function. Shamir's (t,n) SS has been used in conjunction with other public-key algorithms, such as RSA scheme [6] or ElGamal scheme [7], in threshold algorithms. For example, [8, 9] are based on the ElGamal scheme, Various [10-13] are based on the RSA, [14, 15] are based on the Elliptic Curve public-key scheme and [16] is based on Pairing. The RSA algorithm is the predominant mode used today for public-key cryptography. The basic goal of public-key threshold RSA cryptography is to efficiently apply RSA on behalf of a group in a way that ensures integrity, availability, and the security of the private key and the modulus factors. Threshold RSA systems propose a middle ground, which will allow a threshold of any t out of n participants to perform a private key RSA modular exponentiation, while $t-1$ parties cannot perform it and in fact are unable to gain

information about the private key. An important aspect of the proposed schemes is that the resulting group-generated signature is indistinguishable to a verifier from the RSA signature of a single signer. Since shares are protected by public-key algorithms, shares can be used repeatedly to compute multiple outputs. Both SS and threshold cryptography are very active research areas in cryptography. In the processing to compute output of a threshold function, values computed by group members also need to be protected; otherwise, non-members can obtain the output as well. For example, in a threshold decryption, non-members can recover the plaintext if information exchanged among group members is not protected. Thus, the security problem as we have described in the secret reconstruction process also exists in threshold cryptographic applications.

1.2 Contribution

In secret sharing schemes, a secret s is divided into n shares by a dealer D and is distributed among n shareholders. The shares are shared among the shareholders. Until now there is no way to authenticate the shareholder and check whether the shares are authentic. Non-Shareholders who claim to have a share may lead to false secret or can get access to the secret. So, to avoid the access to non-shareholder we introduce a new type of SS, called protected Secret Sharing(PSS) Scheme. In a PSS Scheme, shares of shareholders generated by the dealer initially can not only be used to recover the secret but also be used to protect shares in the secret reconstruction. Therefore, PSS is an efficient way to protect the recovered secret from non-shareholders. we also propose a basic PSS based on a bivariate polynomial. Bivariate polynomials have been used to design verifiable secret sharing scheme (VSS) [17-19] and pairwise key distribution [20-25]. We extend the PSS to the applications of threshold cryptography.

1.3 Thesis Outline

The Thesis is organized as follows, In the next Chapter We Discussed about work related to Secret Sharing, the Shamir Secret Sharing the following chapters we discuss a new type of Secret Sharing scheme which is called protected secret sharing (PSS), in which shares of shareholders can not only be used to compute the secret but also be used to protect shares in the secret reconstruction.

In the 3rd chapter we focused our approach on a basic (t,n) PSS based on a bivariate polynomial and is proposed. Our basic (t,n) PSS scheme is based on a bivariate polynomial is proposed. We gave detailed analysis of the proposed protocol in this chapter. In 4th Chapter we focused on the security and performance of the proposed Scheme and then we introduced to use PSS in applications of threshold cryptography. In 5th Chapter we focused on the application of our proposed scheme in threshold cryptography. The Last Chapter is concluded with the problems and future work of the proposed scheme.

CHAPTER 2

RELATED WORK

2.1 Basic Schemes

In this chapter, we briefly discuss Shamir's secret sharing scheme in detail. We addressed the problem of authenticating the shareholders in the later part of this section and schemes proposed to overcome this problem like Verifiable Secret Sharing (VSS) are discussed. We discussed about the proposal of using bivariate polynomial in Protected Secret Sharing to authenticate shareholders.

In cryptography, distributing a secret among a group of participants is called secret sharing. Each of the participant is allocated a part of secret called "share". The secret can only be reconstructed when the shares are combined together, individual shares are of no use. The primitives of cryptography mainly constitute of secret sharing and its variations. A share generation scheme and a secret reconstruction protocol are the main process in any secret sharing scheme. A protocol for distributing a secret among multiple parties is discussed in share generation and its reconstruction by combining shares from multiple parties is discussed in secret reconstruction. Basic schemes address the problem of secret key reconstruction assuming all parties to be honest.

Secret Sharing Schemes can be classified in to following:

1. Based on share's capabilities:
 - a. Dynamic Secret Sharing: the ability to change the access structure, the dealer has the ability to change a particular access structure, out of a given set and/or allow participants to reconstruct different secret in different time instants.

- b. Proactive Secret Sharing: updating of shares periodically, was proposed by Ostrovsky and Yung[34], which does not consider the old shares and uses new shares generated periodically. This concept was applied to secret sharing by Hezberg_et al [35].
 - c. Secret sharing with veto capability: blocking of reconstruction. It is a feature where qualified set can prevent any other set of participants from reconstructing the secret key.
2. Based on computation power of participants:
- a. Computational Secret Sharing: Participants are computationally bounded. Eg: Krawczyk[36], CSS allows achieving better information rate. Information rate (ρ) is defined as the ratio between average length of the share (in bits) given to the participants and the length of the secret.
 - b. Verifiable Secret Sharing: Qualifying set will be able to recover the secret and disqualified set should not recover the secret. As per verifiable secret sharing, honest players should be able to recover the secret and corrupted players should get no information on it. Tompa and Woll[37] initially introduces cheating in secret sharing, Individual user tricks other users by using fault shares, that is adopted in Shamir's (k,n) scheme. Ogata_et al.[38], finally provides an efficient mode of detecting cheating in secret sharing
 - c. Robust Secret Sharing: Recovering correct secrets in the presence of more number of faulty and corrupted shares. It allows the secret to be reconstructed in the presence of an active adversary who is to corrupt shares. McEliece and Sarwate[39], found first solution to the problem of designing Robust Secret Sharing.

3. CRT schemes: It rely on Chinese Remainder Theorem. CRT based Asmuth and Bloom[40], secret sharing scheme shares the secret S among 'n' parties by modulator arithmetic such that any 't' users can reconstruct the secret by the CRT.

In this work, we mainly concentrate on Verifiable Secret Sharing(VSS) and discuss it under threshold cryptography. In a secret sharing scheme, a dealer divides the secret among shareholders and any authorized set of shareholders can reconstruct the secret but any unauthorized set gain no information about secret.

In Shamir's (t,n) scheme, shareholders cannot verify the validity of their shares from the dealer. We can overcome this by using Protected Secret Sharing(PSS), which has its roots in Verifiability Secret Sharing (VSS) which was first proposed by Chor et al[26].

Protected Secret Sharing(PSS) addresses the problem of verifying the shareholders and is based on bivariate polynomial. Invalid shares maybe caused either by the dealer during share generation or by channel noise during transmission. If invalid shares are detected, shareholders can request the dealer to generate new shares. In a PSS scheme, shares of shareholders are not only used to recover the secret but also be used to protect the shares in secret reconstruction. In PSS, shares are generated using a bivariate polynomial, and it is reconstructed by establishing pairwise keys between shareholders and based on symmetric property of the polynomial coefficients.

The idea of Protected Secret Sharing is that; the recovered secret is to be protected from non-shareholders. Using a bivariate polynomial, shares are generated by the dealer using public information of shareholders and are sent to shareholders secretly. Next, if a particular number of shareholders want to reconstruct the secret, each shareholder uses his share to compute pairwise shared keys, between any other shareholder. It sends its share in an encrypted

form using shared key to all other shareholders. It also receives encrypted message from all other shareholders and decrypts it using the shared secret key. The, each shareholder can use recovered shares to establish linearly independent equations, which are coefficients of the bivariate polynomial. Then, the secret is reconstructed using $a_{0,0}=s$.

If we consider the following problem, where in we can't verify the shareholders, then the secret is reconstructed from the disqualified set if they are able to get the shares of qualified set. Because, Shamir's secret is based on univariate polynomial, where in we can't authenticate the shareholders. We need to add key establishment, which makes the process of secret reconstruction slow. In Protected Secret Sharing, we are using bivariate polynomial which is useful for authentication. Let us discuss the Shamir's (t,n) scheme in detail.

2.1.1 Shamir's Secret Sharing Scheme

A secret sharing scheme divides a secret s into n shares by a dealer D and distributes them among n shareholders $P = \{P_1, P_2, \dots, P_n\}$ in such a way that at least t shares are required to reconstruct the secret and less than t shares gain no information about the secret. The (t,n) threshold secret sharing schemes were introduced by Shamir [29] and Blakley [6] independently in 1979. A (t,n) threshold secret sharing scheme allows any t or more than t shareholders to reconstruct the secret; while fewer than t shareholders can gain no information about the secret. This technique enables the construction of robust key management schemes for cryptographic systems that can function securely and reliably even when misfortunes destroy half the pieces and security breaches expose all but one of the remaining pieces. key management schemes for cryptographic systems that can function securely and reliably even when misfortunes destroy half the pieces and security breaches expose all but one of the remaining pieces.

in [34], Liu considers the following problem:

“Eleven scientists are working on a secret project. They wish to lock up the documents in a cabinet so that the cabinet can be opened if and only if six or more of the scientists are present. What is the smallest number of locks needed? What is the smallest number of keys to the locks each scientist must carry?”

Solution to this problem is not hard to show that the minimal solution uses 462 locks and 252 keys per scientist. These numbers are clearly impractical, and they become exponentially worse when the numbers of scientists increase. we generalize the problem to one In which the secret is s (e.g., the safe combination) and in which non-mechanical solutions are also allowed. Our goal is to divide s Into pieces $U_1, U_2, U_3, \dots, U_n$ in such a way that:

- (1) knowledge of secret s is only possible with t or more than t Shares and finding s is easily computable;
- (2) knowledge of secret s is not possible with $t-1$ Shares or less and finding s is completely undetermined. In the sense that all its possible values are equally likely.

Such a scheme is called a (t,n) threshold scheme.

Efficient threshold schemes can be very helpful in the management of cryptographic keys. In order to protect data we can encrypt the data, but in order to protect the encryption key we need a different method (further encryptions change the problem rather than solve it). The most secure key management scheme keeps the key in a single, well-guarded location let's say a computer, a human brain, or a safe). This scheme is highly unreliable since a single misfortune such as a computer breakdown or sudden death or sabotage can make the information inaccessible. So an obvious solution for this problem is to store multiple copies of the key at different locations, but this increases the danger of security breaches like computer

hacking, betrayal or even human errors. By using a (t, n) threshold scheme with $n = 2t - 1$ we get a very robust key management scheme: We can recover the original key even when $\lfloor n/2 \rfloor = t - 1$ of the n pieces are destroyed, but our opponents cannot reconstruct the key even when security breaches expose $\lfloor n/2 \rfloor = t - 1$ of the remaining t pieces.

In other applications the tradeoff is not between secrecy and reliability, but between safety and convenience of use. Consider the following example, Given a Company that digitally signs all its checks If each executive is given a copy of the company's secret signature key, the system is convenient but easy to misuse. If the cooperation of all the company's executives is necessary in order to sign each check, the system is safe but inconvenient. The standard solution requires at least three signatures per check, and it is easy to implement with a $(3, n)$ threshold scheme. Each executive is given a small magnetic card with one U_i piece, and the company's signature generating device accepts any three of them in order to generate and later destroy a temporary copy of the actual signature key s . The device does not contain any secret information and thus It doesn't need to be a tamper-proof system. An unfaithful executive must have at least two accomplices in order to forge the company's signature in this scheme. Threshold schemes are Ideally suited to applications in which a group a of mutually suspicious individuals with conflicting interests must cooperate. Ideally we would like the cooperation to be based on mutual consent, but the veto power this mechanism gives to each member can paralyze the activities of the group. By properly choosing the k and n parameters we can give any sufficiently large majority the authority to take some action while giving any sufficiently large minority the power to block it

Some of the useful properties of this (t, n) threshold scheme are:

- (1). The size of each piece does not exceed the size of the original data.

(2). shares can be dynamically added or deleted without affecting the other shares keeping the threshold t fixed.

(3). By changing the polynomial to a new polynomial $f(x)$ with the same constant term. We can easily change the shares without changing the original secret s ; frequent change of this type can greatly enhance security since the shares exposed by security breaches cannot be accumulated unless all of them are values of the same edition of the $f(x)$ polynomial.

(4). By using tuples of polynomial values as shares, we can get a hierarchical scheme in which the number of shares needed to determine s depends on their importance. For example, To implement a project if we give the company's Manager has four values of $f(x)$, each team leader has three values of $f(x)$, and each Engineer has two value of $f(x)$, then a $(4, n)$ threshold scheme enables the project to be implemented either by any three members, or by any two team leaders one of whom is a Team leader, or by the Manager alone.

Share Generation Protocol: Dealer D divides the secret s among n shareholders such that at least t shares are required to reconstruct the secret. The share generation protocol is as follows:

(i). Dealer creates a random polynomial $f(x)$ of degree $(t-1)$ and a constant term a_0 where a_0 is the secret s in a finite field (which is known to all the shareholders and the dealer as well). $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1} \pmod{P}$ where a_1, a_2, \dots, a_{t-1} are random polynomials and P is a large prime.

(ii). Dealer randomly selects n distinct points $(x_i \neq 0)$, calculates each share value and distributes the share to each shareholder secretly. share is represented by $S_i = (x_i, f(x_i))$

where $i = 1, 2, \dots, n$. x_i value is a known value. So for our convenience, we chose $x_i = i$. Therefore, $s_1 = f(1)$, $s_2 = f(2)$, ..., $s_n = f(n)$ and the secret $s = f(0)$.

Secret Reconstruction Protocol:

In the n shareholders any subset of t or more shares can be used to reconstruct the secret. The shareholder's subset is: $f(1), f(2), \dots, f(t)$.

Then we use the Lagrange interpolating formula to find the polynomial $f(x)$, such that degree of $f(x) < t$ and $f(i) = S_i$ for $i = 1, 2, \dots, n$. In this the reconstructed secret is found by $f(0)$.

Given any t of n pairs of $(i, f(i))$, with distinct i values, there is a unique polynomial $f(x)$ of degree $t-1$, passing through all these points. This polynomial can be effectively computed from the pairs $(i, f(i))$.

Lagrange's interpolating formula:

In numerical analysis, Lagrange polynomials are used for polynomial interpolation. For a given set of distinct points x_i and numbers y_i , the Lagrange polynomial is the polynomial of the least degree that at each point x_i assumes the corresponding value y_i (i.e. the functions coincide at each point). The interpolating polynomial of the least degree is unique, however, and it is therefore more appropriate to speak of "the Lagrange form" of that unique polynomial rather than "the Lagrange interpolation polynomial", since the same polynomial can be arrived at through multiple methods. The polynomial $f(x)$ can be calculated using Lagrange's interpolating formula as stated below Given a set of $k + 1$ data points $(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$ where $x_i \neq x_j$,

The interpolation polynomial in the Lagrange form is a linear combination

Table 2.1 Lagrange's interpolating formula

$L_i(x)$	$f(x)$
$L_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}$	$f(x) = \sum_{i=1}^t f(i) * L_i(x)$
Where $j \neq i$	$i=1$ where $L_i(x)$ is the Lagrange interpolating polynomial.

Shamir's (t, n) Secret Sharing Scheme satisfies security requirements of a (t, n) SS. That are, with 't' or more than 't' shares we can reconstruct the secret and with fewer than 't' shares we cannot obtain any information of secret. During secret reconstruction phase, non-shareholders may claim that they have shares which are fake and thus the other honest shareholders get nothing but a faked secret. For the fair reconstruction of the secret, cheater detection and/or identification are very essential. However, Shamir's original secret sharing scheme doesn't prevent malicious behavior of dishonest shareholders. So, Shamir's SS scheme is unconditionally secure.

2.1.2 Verifiable Secret Sharing

Using Shamir's secret sharing, dealer may benefit from behaving maliciously. Dealer is assumed to be reliable. But, in reality, dealer can give inconsistent shares to the shareholders. Thus, t participants cannot be able to reconstruct the secret. Verifiable Secret Sharing addresses this issue. VSS has remained as an important area of cryptographic research for the last two decades. Using VSS,

1. Shares are verifiable without revealing shares and secret
2. Convinces shareholders that their shares are k-consistent
3. Each shareholder assures that every subset of k out of n defines the same secret
4. Detects malicious dealer or malicious shareholder

End-to-end auditable voting systems, threshold software key escrow, secure storage are some application of VSS.

In a VSS scheme a dealer, wishes to share a secret among a group of n parties, at most t of which (possibly including the dealer) may be actively malicious, is a fundamental cryptographic primitive, lying at the core of secure multi-party computation (MPC). The property of verifiability enables participants to verify that their shares are consistent according to the security property. In Feldman's VSS scheme is that the committed values are publicly known and the privacy of secret S depends on the difficulty of solving the discrete logarithm problem. In other words, Feldman's scheme is computationally secure. Later, Pedersen (Pedersen, 1992) used a commitment scheme to remove the assumption in Feldman's VSS scheme to propose a VSS scheme which is information theoretically secure. However, in Pedersen's VSS scheme the dealer can succeed in distributing incorrect shares if the dealer can solve the discrete logarithm problem.

The efficiency of a VSS protocol is measured by other parameters as well:

1. The number of rounds of communication required.
2. The number of bits which must be communicated between processors.
3. The number of computations the processors must do.

Chor, Goldwasser, Micali, and Awerbuch [CGMA] introduced the notion of VSS. They present a constant round interactive scheme for verifiable secret sharing based on the assumed intractability of factorization. In their solution, $t=O(\log n)$, $u=O(n)$; the communication complexity is exponential in t .

Feldman's Scheme:

A commonly used example of a simple VSS scheme is the protocol by Paul Feldman, which is based on Shamir's secret sharing scheme combined with any homomorphic encryption scheme. This scheme is, at best, secure for computationally bounded adversaries only. The following description gives the general idea, but is not secure as written. (Note, in particular, that the published value g^s leaks information about the dealer's secret s .)

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 $(\mathbf{Z}_q)^*$, where q is a prime such that p divides $q-1$.)

The dealer then computes (and keeps secret) a random polynomial P of degree t with coefficients in \mathbf{Z}_p , such that $P(0) = s$, where s is the secret. Each of the n shareholders will receive a value $P(1), \dots, P(n)$ modulo p . Any $t+1$ shareholders can recover the secret s by using polynomial interpolation modulo p , but any set of at most t shareholders cannot. (In fact, at this point any set of at most t shareholders has no information about s .)

So far, this is exactly Shamir's scheme. To make these shares verifiable, the dealer distributes commitments to the coefficients of P . If $P(x) = s + a_1x + \dots + a_t x^t$, then the commitments that must be given are $c_0 = g^s, c_1 = g^{a_1}, \dots, c_t = g^{a_t}$.

Once these are given, any party can verify their share. For instance, to verify that $v = P(i) \pmod q$, party i can check that

Introduction to Bivariate Polynomials:

A polynomial in two variables (that is a bivariate polynomial) with constant coefficients is given by $a_{nm} x^n y^m + \dots + a_{22} x^2 y^2 + a_{21} x^2 y + a_{12} x y^2 + a_{11} x y + a_{10} x + a_{01} y + a_{00}$ which can be generalized as $f(x,y) = \sum_{i,j} a_{i,j} x^i y^j$

Uniqueness:

It is possible to find one polynomial which has a degree $n-1$ passes through n points. Assuming the (x_i, y_j) where $i = 1, \dots, n$, pairs of values are unique. Since there are n points, the degree of the interpolating polynomial must have n terms. Thus, the form of the interpolating polynomial may be various, for example, given four points in a square, $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$, the logical choice is $P(x,y) = c_1 x y + c_2 x + c_3 y + c_4$

Let $f(x, y)$ be a function defined for a surface. Given points $((x_1, y_1), z_1)$, $((x_2, y_2), z_2)$, ..., $((x_n, y_n), z_n)$. To find an interpolating polynomial, we simply substitute the points into the bivariate polynomial, and obtained naturally a system of linear equations in the coefficients which may then be solved using Gaussian elimination or LU decomposition.

Proactive Secret Sharing:

Secret sharing with share refreshing is called proactive secret sharing. PSS reduces the window of vulnerability during which an adversary must compromise more than t servers in order to learn the secret. Without share refreshing, the window of vulnerability is unbounded; with PSS, the window of vulnerability is shortened to the period between two consecutive executions of share refreshing. We assume a system of n servers, $A = \{P_1, P_2, P_3, \dots, P_n\}$ that will share a secret value, x , through a $(t+1, n)$, threshold scheme. Each server in A is connected to a common broadcast channel, C , with a property that messages sent on C instantly reach every part connected to it.

In the proactive approach, the secret lifetime is divided into periods of time. At the beginning of each time period, the shareholders 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. At the end of update phase, the servers hold new shares of secret

x. Proactive secret sharing has numerous applications, primarily, maintaining data which is long-lived in scenarios where availability and secrecy are crucial.

Algorithm:

The secret $x \in Z_q$, is encoded into n pieces $x_1, x_2, x_3, \dots, x_n \in Z_q$, using a k -threshold Shamir's Secret Sharing scheme. After the initialization, at the beginning of every time period, all honest servers trigger an update phase, in which the servers perform a share renewal protocol. The shares computed in period t are denoted by $x_i^{(t)}$, $t=1, 2, \dots, n$. To renew the shares at period $t=1, 2, \dots, n$, we can update the polynomial, as follows:

1. P_i picks k random numbers $\{ \delta_{im} \}_{m \in \{1, \dots, k\}}$ from Z_q . These define a polynomial $\delta_i(z) = \delta_{i1}Z^1 + \delta_{i2}Z^2 + \dots + \delta_{ik}Z^k$.
2. For all those servers P_j , P_i secretly sends $u_{ij} = \delta_i(j) \pmod q$ to P_j .
3. After decrypting, u_{ji} , $\forall j \in \{1, 2, \dots, n\}$, P_i computes its new share $x_i^{(t)} = x_i^{(t-1)} + (u_{1i} + u_{2i} + u_{3i} + \dots + u_{ni}) \pmod q$ and erases all the variables it used except of its current key, $x_i^{(t)}$.

This protocol solves the share renewal problem against a malicious user learning the secret information available to corrupted servers. The idea of Proactive secret sharing is periodically renewing the shares without changing the secret, in such a way that any information learned by the adversary about the individual shares become obsolete after the time period.

CHAPTER 3

OUR SCHEME

3.1 Motivation

Threshold cryptography is the study of multiparty computation protocols for different cryptographic functions (e.g. signing or decrypting) in which each group member receives a share of a private key generated by the group manager (GM) initially and then uses the share later to jointly compute an output of the cryptographic function. Shamir's (t,n) SS has been used in conjunction with other public-key algorithms, such as RSA scheme [6] or ElGamal scheme [7], in threshold algorithms. For example, [8, 9] are based on the ElGamal scheme, [10-13] are based on the RSA, [14, 15] are based on the Elliptic Curve public-key scheme and [16] is based on Pairing. Since shares are protected by public-key algorithms, shares can be used repeatedly to compute multiple outputs. Both Secret Sharing and threshold cryptography are very active research areas in cryptography. In the processing to compute output of a threshold function, values computed by group members also need to be protected, i.e. shares with the respective shareholders needs to be protected, otherwise, non-members who is not a shareholder for the secret s can obtain the output as well. For example, in a threshold decryption, non-members can recover the plaintext if information exchanged among group members is not protected. So we encrypt the information exchanged among groups. Thus, the security problem as we have described in the secret reconstruction process also exists in threshold cryptographic applications.

3.2 Contribution

In the secret reconstruction of the Shamir's (t,n) secret sharing scheme (SS) the shares released by shareholders need to be protected otherwise any non-shareholders can also obtain

the secret. Key establishment protocol can establish pairwise keys for any pair of shareholders. Then, shareholders can use these pairwise keys to protect shares in the secret reconstruction process. But by adding a key establishment in the secret reconstruction slows down the process significantly. Shamir's Secret Sharing Scheme is based on a univariate polynomial. Shares generated by a bivariate polynomial enable pairwise keys to be shared between any pair of shareholders. In this paper, we introduce a new type of SS, called protected secret sharing scheme (PSS), in which shares of shareholders can not only be used to reconstruct the secret but also be used to protect the secrecy of shares in the secret reconstruction process. Thus, the recovered secret is only available to shareholders but not to non-shareholders. A basic (t,n) PSS based on a bivariate polynomial is proposed. We also introduced how to use this basic Protected Secret Sharing (PSS) in the applications of threshold cryptography. The Protected Secret Sharing (PSS) Scheme is unique since it protects the secrecy of the recovered secret in a very efficient way. There is one major difference between shares generated by a univariate polynomial and by a bivariate polynomial. The shares generated by a univariate polynomial are integers in $GF(p)$ but shares generated by a bivariate polynomial are univariate polynomials.

In Shamir's (t,n) SS [1], the dealer selects a univariate polynomial, $f(x)$ with degree $t-1$ and $f(0)=s$ where s is the secret. The dealer generates shares, $f(x_i) \bmod p$ where $i=1,2,3,\dots,n$ for shareholders, where p is a prime with $p>s$ and x_i is the public information associated with each shareholder U_i . Each share, $f(x_i)$ is an integer in $GF(P)$. Shamir's (t,n) SS satisfies security requirements of a (t,n) SS. That are, (a) with t or more than t shares can reconstruct the secret, and
 (b) with fewer than t shares cannot obtain any information of the secret.

Therefore, Shamir's Secret Sharing Scheme is unconditionally secure.

Using Shannon's definition of entropy, any SS involving a set, P of shareholders and having the access structure, Γ , needs to satisfy two requirements:

- i) Correctness: For any qualified subset, $A \subseteq P$ of shareholders can recover the secret, s . Formally, for all $A \in \Gamma$ it holds. $H(s|A) = 0$.
- ii) Security: For any unqualified subset, $A \subseteq P$ of shareholders cannot recover the secret, s . Formally, for all $A \notin \Gamma$ it holds $0 < H(s|A) \leq H(s)$

3.2.1 Definition 1. Perfect Secret Sharing Scheme (Perfect SS)

In above security requirement, if for any $A \in \Gamma$ it holds $H(s|A) = H(s)$ (that is, shareholders in A obtain no information on s), the SS is called a perfect SS.

In Shamir's (t,n) SS, shareholders cannot verify the validity of their shares obtained from the dealer. In 1985, Chor et al. [26] extended the notion of SS and proposed the first verifiable secret sharing (VSS). Verifiability is the property of a VSS which allows shareholders to verify their shares. Invalid shares may be caused either by the dealer during share generation or by channel noise during transmission. VSS is performed by shareholders after receiving their shares from the dealer and before using their shares to reconstruct the secret. If invalid shares have been detected, shareholders can request the dealer to regenerate new shares. There are many (t,n) VSSs [27-32] using bivariate polynomials, denoted them as BVSSs. A bivariate polynomial with degree $t-1$ can be represented as

$$F(x,y) = a_{0,0} + a_{1,0}x + a_{0,1}y + a_{2,0}x^2 + a_{0,2}y^2 + a_{1,2}xy^2 + a_{2,1}x^2y + a_{2,2}x^2y^2 + \dots + a_{t-1,t-1}x^{t-1}y^{t-1} \pmod{P}$$

where $a_{i,j} \in GF(p) \forall i, j \in [0, t-1]$

We can classify the BVSSs into two types, the Symmetric BVSSs, denoted them as SBVSSs [29-32] and the Asymmetric BVSSs, denoted them as ABVSSs, [27, 29, 31]. If the

coefficients satisfy $a_{i,j} = a_{j,i} \forall i, j \in [0, t - 1]$ then it is a symmetric bivariate polynomial. Shares generated by a bivariate polynomial can be used to establish pairwise keys between any pair of shareholders. In all (t,n) SBVSSs, the dealer selects a bivariate polynomial, $F(x,y)$ with degree $t-1$ and $F(0,0)=s$ where s is the secret. The dealer generates shares, $F(x_i,y) \bmod P$ where $i=1,2,\dots,n$, for shareholders, where P is a prime with $P > s$ and x_i is the public information associated with each shareholder, U_i . Each share, $F(x_i,y)$ is a univariate polynomial with degree $t-1$. Note that shares generated in a SBVSS satisfy $F(x_i,x_j) = F(x_j,x_i) \forall i, j \in [0, t - 1]$ the pairwise key, $F(x_i,x_j) = F(x_j,x_i)$ can be established between the pair of shareholders, U_i and U_j . In a similar way, in a ABVSS, the dealer generates a pair of shares, $F(x_i,y) \bmod P$ and $F(x,x_i) \bmod P$ where $i=1,2,\dots,n$, for each shareholder and the pairwise secret key, $F(x_i,x_j)$ or $F(x_j,x_i)$ can also be established between the pair of shareholders, U_i and U_j .

3.2.2 The Basic (t,n) PSS Using a Bivariate Polynomial

We first give the following definition.

Definition 2. Protected Secret Sharing Scheme (PSS)

In a PSS, shares of shareholders generated by the dealer initially can not only be used to compute the secret but also be used to protect shares in the secret reconstruction. Thus, the recovered secret is only available to shareholders but not available to non-shareholders.

In Shamir's (t,n) SS, additional key establishment protocol is needed to protect shares in the secret reconstruction; otherwise, any non-shareholders can also recover the secret. Thus, Shamir's (t,n) SS is not a PSS.

3.3 Share Generation and Authentication

3.3.1 Private Share Generation

A secret s into n shares and divides it among n shareholders by the Group Manager in such a way that at least t shares are required to reconstruct the secret where ‘ t ’ is the threshold value that can be decided by the Group Manager (GM). The protocol for private share generation is designed as follows:

3.3.2 Private Share Generation Protocol

Dealer selects a prime p where $p > s$, s being the secret.

The dealer selects a symmetric bivariate polynomial of $h-1$ degree (i.e., with $2t - 1 \geq h > 2t - 3$ We will explain this condition later in Theorem1

$$F(x,y) = a_{0,0} + a_{1,0}x + a_{0,1}y + a_{2,0}x^2 + a_{0,2}y^2 + a_{1,2}xy^2 + a_{2,1}x^2y + a_{2,2}x^2y^2 + \dots + a_{h-1,h-1}x^{h-1}y^{h-1} \pmod{P}$$

where. $A_{i,j} \in GF(p)$, $a_{i,j} = a_{j,i} \forall i, j \in [0, h - 1]$

The dealer computes shares, $s_i(y) = F(x_i, y) \pmod{P}$ for shareholders, U_i where $i=1,2,\dots,n$ where x_i is the public information associated with each shareholder, U_i The dealer sends each share, $s_i(y)$ to shareholder U_i secretly. VSS is performed by shareholders after receiving their shares from the dealer and before using their shares to reconstruct the secret

3.3.3 Public Shares Authentication

After renewing each master secret on the centralized server, dealer needs to broadcast public shares of the renewed master secret to all shareholders. All shareholders can use their private shares to work together to authenticate the public shares of the master secret.

3.4 Master Secret Generation and Reconstruction

3.4.1 Master Secret Generation

Dealer constructs a (t,n) secret sharing scheme in such a way that for a given secret s only t or more than t shareholders can reconstruct the secret and less than t shareholders cannot reconstruct the secret. In our scheme each share for a shareholder is a univariate polynomial.

3.4.2 Master Secret Generation Protocol

Let the threshold of the new master secret be ' t ', The dealer constructs a polynomial $f(x,y)$ with degree $h-1$ and such that $s_i(y)=F(x_i,y) \bmod P$, for $i = 1,2,\dots, n$, and $F(0,0) = s$ where s is the secret., where s is the master secret, where x_i is a publicly known parameter associated with each shareholder U_i .

Remarks: (a) Dealer can use the Lagrange Interpolating formula to construct the polynomial

$$f(x) = \sum_{i=0}^n y_i \prod_{j=0, j \neq i}^n \frac{x-x_j}{x_i-x_j} \bmod p$$

3.4.3 Master Secret Reconstruction

Secret Reconstruction is done in four basic following steps as listed below

Let's assume that U shareholders represented as $\{u_{v_1}, u_{v_2}, \dots, u_{v_u}\}$ want to reconstruct the secret.

Here $t \leq u \leq n$

Step 1. Each U_{v_i} uses his/her share, $S_{v_i}(y)$ to compute the pairwise shared keys $k_{i,j}$, $k_{i,j} =$

$$S_{v_i}(x_{v_j})$$

$= F(x_{v_i}, x_{v_j})$ where $j=1,2,\dots,u$ and $j \neq i$ Here $k_{i,j}$ is the secret key shared between

shareholders, U_{v_i}

and, U_{v_j}

Step 2. (a) If $t \leq u \leq h$ each shareholder U_{v_i} computes the cipher text by encrypting the share

$S_{v_i}(y)$, $C_{i,j} = E_{k_{i,j}}(S_{v_i}(y))$ where $j=1,2,\dots,u$ $j \neq i$ Here $E_{k_{i,j}}(S_{v_i}(y))$ denotes the conventional encryption of share $S_{v_i}(y)$ using the key $k_{i,j}$

(b) Otherwise, if $h \leq u$ each shareholder U_{v_i} computes the cipher text by encrypting

$$C_{i,j} = E_{k_{i,j}}(w_{v_i}) \quad j=1,2,\dots,u \quad j \neq i \quad \text{where} \quad w_{v_i} = S_{v_i}(0) \prod_{l=1, l \neq i}^u \frac{-x_{v_l}}{x_{v_l} - x_{v_i}} \pmod{P}$$

Each shareholder U_{v_i} sends the cipher $c_{i,j}$, $j=1,2,\dots,u$ $j \neq i$ to other shareholders

Step 3. After receiving cipher text, $c_{j,i}$, $j=1,2,\dots,u$ $j \neq i$ from other shareholders, U_{v_i} computes

the decryption $D_{k_{i,j}}(c_{j,i})$, $j=1,2,\dots,u$ $j \neq i$ where $D_{k_{i,j}}(c_{j,i})$ denotes the decryption of $c_{j,i}$ using the key $k_{i,j}$

Step 4. (a) If $t \leq u < h$ each shareholder U_{v_i} computes $D_{k_{i,j}}(c_{j,i}) = S_{v_j}(y)$ $j=1,2,\dots,u$ $j \neq i$. Then,

each shareholder U_{v_i} can use recovered shares to establish at least $\frac{h(h+1)}{2}$ linearly independent equations, for example, by computing $S_{v_j}(y)$, $y=1,2,\dots,h$ and $j=1,2,\dots,u$ (i.e., $F(x_{v_j}, y)$ where $y=1,2,\dots,h$ and $j=1,2,\dots,u$. In other words, these linearly independent equations are expressed in terms of the coefficients, $a_{i,j}$ of the polynomial $F(x,y)$ Then, the secret can be obtained from $a_{0,0}=s$

(b) Otherwise, if $h \leq u$ each shareholder U_{v_i} computes $D_{k_{i,j}}(c_{j,i}) = W_{v_j}$ $j=1,2,\dots,u$ $j \neq i$

Then, the secret is recovered by computing $\sum_{i=1}^u W_{v_i} \pmod{p} = s$

CHAPTER 4

SECURITY ANALYSIS AND PERFORMANCE

4.1 Security Analysis

In this section, we will first prove that the basic protocol meets the security requirements of a SS. Then, we will prove the protocol is a perfect SS as defined in Definition 1 and a PSS as defined in Definition 2. We will also prove that our scheme is secure with different types of attacks Inside and Outside attack.

4.1.1 Inside Attack

Theorem 1. With $2t - 1 \geq h > 2t - 3$ the proposed scheme satisfies both security requirements of a (t,n) Secret Sharing Scheme. That are,

- (a) with t or more than t shares can recover the secret
- (b) with fewer than t shares cannot recover the secret.

Proof. $F(x,y) = a_{0,0} + a_{1,0}x + a_{0,1}y + a_{2,0}x^2 + a_{0,2}y^2 + a_{1,2}xy^2 + a_{2,1}x^2y + a_{2,2}x^2y^2 + \dots + a_{h-1,h-1}x^{h-1}y^{h-1} \pmod{P}$ is a symmetric polynomial with $a_{i,j} \in GF(p)$, $a_{i,j} = a_{j,i} \forall i, j \in [0, h - 1]$

Containing $\frac{h(h+1)}{2}$ different coefficients of the polynomial $F(x,y)$. In the proposed scheme, each share, $s_i(y)$ is a univariate polynomial with degree $h-1$. In other words, each shareholder can use his share to establish h linearly independent equations in terms of the coefficients of the polynomial $F(x,y)$ With $t-1$ shares together, it can establish $h(t-1)$ linearly independent equations. Since

$h > 2t - 3$ as specified in the share generation, we get $\frac{h(h+1)}{2} > h(t-1)$. Thus, any $t-1$ colluded shareholders cannot recover the secret. This conclusion is obtained without making any computational assumption. On the other hand, when there are 't' or more than 't' shareholders trying to recover the secret, with their shares together, they can establish 'ht' linearly independent equations. Since $2t-1 \geq h$ as specified in the share generation, we have $ht \geq \frac{h(h+1)}{2}$

Thus, any t or more than t shareholders can recover the secret.

Corollary 1.1. For any given threshold, t the degree of the symmetric polynomial, $F(x,y)$ can either be $2t-1$ or $2t-2$.

Proof. This corollary can be obtained directly from Theorem 1.

We now want to prove that our proposed basic scheme is a perfect Secret Sharing Scheme.

Theorem 2. The proposed basic scheme using a bivariate polynomial is a perfect SS as defined in Definition 1.

Proof. Let us prove this theorem following Definition 1 Correctness and Security.

i) Correctness: We consider two cases separately with $t \leq u < h$ and with $h \leq u$.

(a). If $t \leq u < h$, as we have discussed in Theorem 1, with t or more than t shareholders we can recover the secret. In Step 4, each shareholder U_{v_j} after obtaining u shares, can establish at least $\frac{h(h+1)}{2}$ linearly independent equations, such as $s_{v_j}(y)$, $y=1,2,..h$ and $j=1,2,..u$ (i.e., $F(x_{v_j}, y)$, $y=1,2,..h$ and $j=1,2,..u$ in terms of the coefficients, a_{ij} of the polynomial $F(x,y)$)

Then, the secret can be obtained as $a_{0,0} = s$

(b). If $h \leq u$ according to the Lagrange interpolation formula, we can get

$$\sum_{i=1}^u S_{v_i}(y) \prod_{l=1, l \neq i}^u \frac{x-x_{v_l}}{x_{v_i}-x_{v_l}} \text{mod } P = F(x,y). \text{ Thus, in Step 4, we get } \sum_{i=1}^u w_{v_i} \text{mod } P =$$

$$\sum_{i=1}^u S_{v_i}(0) \prod_{l=1, l \neq i}^u \frac{-x_{v_l}}{x_{v_i}-x_{v_l}} \text{mod } P = F(0,0) = s$$

This concludes that for any qualified subset, $A = \{u_{v_1}, u_{v_2}, \dots, u_{v_u}\} \in \Gamma$, of shareholders can work together to recover the secret. Hence, it holds that $H(s|A) = 0$

ii) Security: In our proposed scheme, all the information exchanged among shareholders is encrypted using pairwise shared keys. Since non-shareholder does not own any share generated by the dealer, non-shareholder cannot decrypt any cipher text. Thus, the recovered secret is not available to non-shareholder. In other words, non-shareholder obtain no information on s . Furthermore, since dealer selects a $h-1$ degree symmetric polynomial, $F(x,y)$ as we have discussed in Theorem 1, it needs at least t shares to recover the secret polynomial $F(x,y)$ For any unqualified subset, $B = \{u_{v_1}, u_{v_2}, \dots, u_{v_r}\} \notin \Gamma$, with $0 < r < t$ of shareholders cannot establish sufficient number of linearly independent equations to recover $F(x,y)$ and therefore cannot get s from $\sum_{i=1}^r w_{v_i} \text{mod } P$ Namely, shareholders in B obtain no information on s Hence, it holds that $H(s|A) = H(s)$. Therefore, according to Definition 1, the proposed basic scheme is a perfect Secret Sharing Scheme.

4.1.2 Outside Attack

Outside Attack: The proposed basic scheme is a PSS as defined in Definition 2.

Proof. In this basic scheme, from Theorems, 1 and 2, shares of shareholders generated by the dealer can not only be used to compute the secret in Step 1 but also be used to protect information exchanged among shareholders in Step 2. Thus, non-shareholders cannot obtain the secret. According to Definition 2, the basic scheme is a PSS.

4.2 Performance

In the basic PSS, each share, $S_i(y)$ is a univariate polynomial with degree $h-1$. Thus, each shareholder needs to store h coefficients of a univariate polynomial. The memory storage of each shareholder is $h \log_2 p$ bits, where P is the modulus. Horner's rule [33] can be used to evaluate polynomials. In the following discussion, we show the cost for computing $w_{v_i} = S_{v_i}(0) \prod_{l=1, l \neq i}^u \frac{-x_{v_l}}{x_{v_i} - x_{v_l}} \text{mod } P$ in the secret reconstruction. From Horner's rule, evaluating a polynomial of degree $h-1$ needs $h-1$ multiplications and h additions. Since multiplication takes more time than addition, the performance is only addressed to the number of multiplications needed. The computational cost in Step 1 to compute W_{v_i} is to evaluate one polynomial. The computational cost in Step 2 to compute pairwise shared keys, $k_{i,j} = S_{v_i}(x_{v_j})$ where $j=1, 2, \dots, u$ and $j \neq i$ is to evaluate $u-1$ polynomials, where u is the number of shareholders participated in the secret reconstruction. Overall, the computational cost to reconstruct the secret of each shareholder is to compute 'uh' multiplications.

CHAPTER 5

APPLICATION TO ALGORITHMS OF THRESHOLD CRYPTOGRAPHY

In a threshold algorithm, the GM is responsible to select a pair of public and private keys of the group and to register group members initially. The GM follows our proposed basic PSS in Section 3 to treat the private key of the group as the secret and generate shares of group members. The share of each group member is a univariate polynomial with degree $h-1$. In the process to compute the threshold function, each group member uses his/her share to compute the function output and pairwise keys shared with other group members. Then, the output is encrypted using pairwise shared keys with other group members and send the cipher text to other group members. Similarly, all received cipher text needs to be decrypted using pairwise shared keys with other group members. The output of the threshold function is finally obtained from these decrypted cipher text. Since non-members do not have any share generated by the GM, non-members cannot decrypt the cipher text. Thus, the output of threshold function is not available to non-members.

CHAPTER 6

CONCLUSION

A new type of SS, called protected secret sharing (PSS), is introduced . In a PSS, shares of shareholders can not only be used to compute the secret but also be used to protect shares in the secret reconstruction. A basic (t,n) PSS using a bivariate polynomial is proposed. Security and performance analysis of the scheme is also included. We extend the basic scheme to threshold algorithms. The PSS is unique since it protects the secrecy of the recovered secret in an efficient way. In secret sharing scheme, the master secret and all the private shares of shareholders are to be maintained secretly. In all the existing secret sharing schemes, we can reconstruct the secret but cannot authenticate the shareholder. But in our approach, we can authenticate the shareholders thereby preventing the malicious users gaining knowledge of secret.

All the existing sharing schemes, assume that the shareholders are reliable and cannot be attacked. Therefore, we authenticate only the shares from shareholders but not the shareholders itself. In our scheme, we authenticate the shareholders to determine the shareholder identity before reconstructing the secret. If the shareholder is not authenticated, then the secret cannot be reconstructed using his share. A new share is requested from the dealer by this shareholder and then he can participate in the secret reconstruction using his new share only if he is authenticated using his new share. By implementing this PSS, not only the master key is secured but also its reconstruction is secured.

We proposed a scheme based on bivariate polynomial, where we use the property of symmetry to authenticate the shareholders. We added this feature of authentication to the existing features like secrecy, efficiency, confidentiality, which are to be satisfied by every

secret sharing scheme. We mentioned our approach of Protected Secret Sharing, public and private share generation, and authentication protocols. Then we discussed master secret reconstruction protocol which uses VSS and symmetric property of bivariate polynomial coefficients to authenticate shareholders and reconstruct the secret.

6.1 Open Problems

One of the open problem for secret sharing is to improve the $O(n/\log n)$ bound at least by a factor of $\log n$. The above bound is a consequence of Shannon's inequalities for the entropy function. One more problem of the SS scheme is to have limitation for finite n instead of going infinite i.e. does there exist an (infinite, ∞) scheme where secret is determined by arbitrary infinite collections of shares, but which is independent of any finite collection of shares. To use secret-sharing schemes, we should also require that the sharing process and the reconstruction are efficient. That is, when using secret-sharing schemes we want the honest parties, which share secrets and reconstruct them, to run in polynomial time.

If all the private shares are able to recover the master secret, then the secret is no longer secure. So it is necessary to renew the master secret keeping all the private shares secure. If all the private shares are kept secure during secret reconstruction, dealer only needs to renew master secret but not the private shares. So in this way, the secret sharing scheme becomes even more efficient as the private shares can be reused for a longer period of time.

If once the shareholders are able to recover the master secret, then the secret is no longer secure. So it is necessary to refresh both the shares and secret at the same time.

One of the limitations of protected secret scheme is that they assume the set of shareholders remain same for a given secret sharing. The protected secret sharing scheme

could further be improved by adding and deleting shareholders while sharing the same master secret.

6.2 Future Work

This Protected Sharing Schemes addresses the main problem of authentication in secret sharing. We used bivariate polynomials not just for authentication but also for secret sharing. If this is employed to all the exiting secret sharing schemes, security is enhanced to the maximum level. PSS can be used in Public Key Infrastructure(PKI) schemes where certificate authority(CA), is used to generate digital signatures. CA needs to authenticate the parties for secret reconstruction. PSS can be used for this purpose. CA has a private key for generating digital signatures. Instead of storing the entire key secretly, it is better to divide the key into number of shares and share the key. This private key can be stored as a secret, and is shared in a secret sharing scheme. By using PSS, we can authenticate, shareholders thereby preventing the secret reconstruction in the existence of malicious users. Adding and removing shareholders is also easy, as they need to only satisfy that Symmetric property, then can perform in secret reconstruction.

REFERENCES

- [1] A. Shamir, "How to share a secret", In: Communications of the ACM, vol. 22, no. 11, pp. 612-613, 1979.
- [2] G. R. Blakley, "Safeguarding cryptographic keys," In Proceedings of AFIPS'79 Nat. Computer Conf., vol. 48, AFIPS Press, pp. 313-317, 1979.
- [3] M. Mignotte, "How to share a secret," In Cryptography-Proceedings of the Workshop on Cryptography, in: LNCS, vol. 149, Springer-Verlag, pp. 371-375, 1983.
- [4] C. A. Asmuth, J. Bloom, "A modular approach to key safeguarding," IEEE Transactions on Information Theory, vol. IT-29, no. 2, pp. 208-210, 1983.
- [5] y. Desmedt, "Society and group oriented cryptography: a new concept," In Proc. Advances in Cryptology-Crypto'87, in: LNCS, vol. 293, Springer-Verlag, pp. 120-127, 1987.
- [6] R. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. Assoc. Comp. Mach., vol. 21, no. 2, pp. 120-126, 1978.
- [7] T. A. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," IEEE Trans. Information Theory, vol. 31, no. 4, pp. 469-472, 1985.
- [8] Y. Desmedt and Y. Frankel, "Threshold cryptosystems," In Advances in Cryptology-Crypto '89, pp. 307-315, 1989.
- [9] L. Harn, "Group-oriented (t, n) threshold digital signature scheme and digital multisignature," IEE Proc.-Comput. Digit. Tech., vol. 141, no. 5, pp. 307-313, Sep. 1994.
- [10] Y. Desmedt and Y. Frankel, "Shared generation of authenticators and signatures," In Advances in Cryptology-Crypto '91, pp. 457-569, 1991.
- [11] A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung, "How to share a function securely," In 26th Annual ACM Symposium on Theory of Computing, pp. 522-533, 1994.

- [12] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust and efficient sharing of RSA functions," In *Advances in Cryptology-Crypto '96*, pp. 157-172, 1996.
- [13] V. Shoup, "Practical threshold signatures," In *Advances in Cryptology-Eurocrypt 2000*, pp. 207-220, 2000.
- [14] Ertaul, Levent, and Weimin Lu. "ECC based threshold cryptography for secure data forwarding and secure key exchange in MANET (I)." *International Conference on Research in Networking*. Springer Berlin Heidelberg, 2005.
- [15] Y. Shang, X. Wang, Y. Li and Y. Zhang, "A general threshold signature scheme based on Elliptic Curve," *Proceedings of the 2012 2nd International Conference on Computer and Information Application (ICCIA 2012)*.
- [16] W. Gao, G. Wang, X. Wang and Z. Yang, "One-round ID-based threshold signature scheme from bilinear pairings," *INFORMATICA*, vol. 20, no. 4, pp. 461-476, 2009.
- [17] M. Fitzi, J. Garay, S. Gollakota, C.P. Rangan, K. Srinathan, "Round-optimal and efficient verifiable secret sharing", In *3rd Theory of Cryptography Conference, TCC'06*, in: LNCS, vol. 3876, pp. 329–342, 2006.
- [18] R. Kumaresan, A. Patra and C.P. Rangan, "The round complexity of verifiable secret sharing: the statistical case." In *Advances in Cryptology-Asiacrypt'10*, in: LNCS, vol. 6477, pp. 431-447, 2010.
- [19] A. Patra, A. Choudhary, T. Rabin and C.P. Rangan, "The round complexity of verifiable secret sharing revisited", In *Advances in Cryptology-Crypto'09*, in: LNCS, vol. 5677, pp. 487-504, 2009.

- [20] Liu, Donggang, Peng Ning, and Rongfang Li. "Establishing pairwise keys in distributed sensor networks." *ACM Transactions on Information and System Security (TISSEC)* 8.1 (2005): 41-77.
- [21] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks", *ACM Trans. Inf. Syst. Secur.*, 8, pp. 41-77, 2005.
- [22] Cheng, Yi, and Dharma P. Agrawal., "A improved key distribution mechanism for large-scale hierarchical wireless sensor networks", *Journal of Ad Hoc Networks*, vol. 5, no. 1, pp. 35-48, 2005.
- [23] G. Song and Q. Zhuhong, "A Compromise-resilient group rekeying scheme for hierarchical wireless sensor networks", *Wireless Communications and Networking Conference (WCNC)*, pp.1-6, 2010.
- [24] H. Liang and C. Wang, "An energy efficient dynamic key management scheme based on polynomial and cluster in wireless sensor networks", *Journal of Convergence Information Technology*, vol. 6, no. 5, pp. 321-328, 2011.
- [25] N. Saxena, G. Tsudik and J. H., Yi, "Efficient node admission and certificateless secure communication in short-lived MANETs", *IEEE Trans. on Parallel and Distributed Systems*, vol. 20, no. 2, pp.158-170, 2009.
- [26] B. Chor, S. Goldwasser, S. Micali and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," *Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science*, Oregon, Portland, pp. 383-395, 1985.

- [27] R. Cramer, I. Damgard, S. Dziembowski, M. Hirt, T. Rabin, "Efficient multiparty computations secure against an adaptive adversary", in: Proceedings of 18th Annual IACR EUROCRYPT, Prague, Czech Republic, in: LNCS, vol. 1592, pp. 311-326, 1999.
- [28] M. Fitzi, J. Garay, S. Gollakota, C. Pandu Rangan, K. Srinathan, "Round-optimal and efficient verifiable secret sharing", in: S. Halevi and T. Rabin, editors, Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, 4-7 March, 2006, in: LNCS, vol. 3876, Springer, pp. 329-342, 2006.
- [29] Gennaro, Rosario, et al. "The round complexity of verifiable secret sharing and secure multicast." Proceedings of the thirty-third annual ACM symposium on Theory of computing. ACM, 2001.
- [30] J. Katz, C. Koo, R. Kumaresan, "Improved the round complexity of VSS in point-to-point networks", in: Proceedings of ICALP '08, Part II, in: LNCS, vol. 5126, Springer, pp. 499-510, 2008.
- [31] R. Kumaresan, A. Patra and C.P. Rangan, "The round complexity of verifiable secret sharing: the statistical case", in: Advances in Cryptology - ASIACRYPT 2010, in: LNCS, vol. 6477, Springer, pp. 431-447, 2010.
- [32] V. Nikov and S. Nikova, "On proactive secret sharing schemes", in: LNCS, vol. 3357, Springer, pp. 308-325, 2005.
- [33] D. E. Knuth. The Art of Computer Programming, Semi-numerical Algorithms,, volume II. Addison Wesley, Reading Massachusetts, 1981.
- [34] R. Ostrovsky and M. Yung, "How to withstand mobile virus attacks." Proceedings of the tenth annual ACM symposium on Principles of distributed computing. ACM, 1991.

- [35] A. Herzberg, et al. "Proactive secret sharing or: How to cope with perpetual leakage." Annual International Cryptology Conference. Springer Berlin Heidelberg, 1995.
- [36] H. Krawczyk, "Secret sharing made short." Annual International Cryptology Conference. Springer Berlin Heidelberg, 1993.
- [37] M. Tompa and H. Woll, "How to share a secret with cheaters." Journal of Cryptology 1.3 (1989): 133-138.
- [38] K. Kurosawa, S. Obana, and W. Ogata, " t -Cheater Identifiable (k, n) Threshold Secret Sharing Schemes." Annual International Cryptology Conference. Springer Berlin Heidelberg, 1995.
- [39] R.J. McEliece and D.V. Sarwate, "On sharing secrets and Reed-Solomon codes." Communications of the ACM 24.9 (1981): 583-584.
- [40] C. Asmuth and J. Bloom, "A modular approach to key safeguarding." IEEE transactions on information theory 30.2 (1983): 208-210.

VITA

Spandan Mannava was born in Guntur, Andhra Pradesh, India. on June 8th 1993. After completion of 12th standard at Sri Chaitanya, He went to Indian Institute of Information Technology Design and Manufacturing Kancheepuram, Chennai, Tamil Nadu, India. He is very ambitious and moved to Kansas City in spring 2015 to pursue master's in Computer Science at University of Missouri-Kansas City. During his graduate studies, he is Software Developer Intern at Networks International Corporation, Overland Park, Kansas.