CENTRALIZED SECRET SHARING SCHEMES AND DISTRIBUTED GROUP KEY AGREEMENT PROTOCOLS

A thesis submitted during 2023 to the University of Hyderabad in partial fulfillment of the award of a Ph.D. degree in Computer Science

by

ROLLA SUBRAHAMANYAM

Enrollment No. 17MCPC06

School of Computer and Information Sciences

University of Hyderabad

P.O. Central University, Gachibowli

Hyderabad, Telangana

India, 500046

July, 2023

CERTIFICATE

This is to certify that the thesis entitled **Centralized Secret Sharing Schemes and Distributed Group Key Agreement Protocols** submitted by **Rolla Subrahmanyam** bearing **Reg. No. 17MCPC06** for partial fulfillment of the requirements for the award of **Doctor of Philosophy in Computer Science** is a bonafide work carried out by him under our supervision and guidance.

The thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma. The student has the following publications before submission of the thesis for adjudication and has produced the evidence for the same.

1. Subrahmanyam, Rolla, N. Rukma Rekha, and Y. V. Subba Rao. "Multipartite verifiable secret sharing based on CRT." Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021. Springer Singapore, 2022. (Published, Index: Scopus) (Published, Index: Scopus)

2. Subrahmanyam, Rolla, N. Rukma Rekha, and Y. V. Subba Rao. "Compartmented Proactive Secret Sharing Scheme." International Conference on Intelligent Sustainable Systems. Singapore: Springer Nature Singapore, 2023. (Published, Index: Scopus)

3. Subrahmanyam, Rolla, N. Rukma Rekha, and YV Subba Rao. "Authenticated Distributed Group Key Agreement Protocol Using Elliptic Curve Secret Sharing Scheme." IEEE Access (2023). (Published, Index: SCIE)

4. Subrahmanyam, Rolla, N. Rukma Rekha, and YV Subba Rao. (2023) 'Multi-Group Key Agreement Protocol Using Secret Sharing Scheme', Int. J. Security and Networks, (Accepted, Index : Scopus)

Further, the student has passed the following courses towards fulfillment of coursework requirement for Ph.D.

Course Code	Name	Credits	Pass/Fail
CS 801	Data structure and Programming Lab	2	Pass
CS 802	Algorithms	4	Pass
CS 803	Research Methods in Computer Science	4	Pass
CS805	Cryptography	4	Pass

N. Rukma Rekha 24/7/23

Dr. N. Rukma Rekha
Supervisor, School of Computer
and Information Sciences,
University of Hyderabad

Prof. Atul Negi
Dean, School of Computer
and Information Sciences,
University of Hyderabad

DECLARATION

I, **Rolla Subrahmanyam**, hereby declare that this thesis entitled **Centralized Secret Sharing Schemes and Distributed Group Key Agreement Protocols** submitted by me under the guidance and supervision of **Dr. N. Rukma Rekha** is a bonafide research work. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma. A report on plagiarism statistics from the University Librarian is enclosed.

Date: 24|7|23

R Subrahmanyam
Signature of the Student

Name: **Rolla Subrahmanyam**
Reg. No. **17MCPC06**

Acknowledgements

I am profoundly grateful to my supervisor, **Dr.N. Rukma Rekha.** Her unwavering guidance, combined with her comprehensive intellectual insights, have been a cornerstone in my research journey. Her exceptional communication skills, along with her friendly disposition and remarkable patience, have not only made working under her an enriching and enlightening experience but also fostered a sense of freedom and autonomy in my work. The confidence and trust she placed in me allowed for open, constructive dialogues that greatly enriched my research perspective. Her belief in my capabilities encouraged me to step beyond the boundaries of conventional thought and seek innovative solutions.

Equally, my deepest gratitude goes to **Dr. Subba Rao YV.** His wisdom and nurturing mentor ship have been unparalleled. As a strong mathematician, his scholarly rigor and integrity have significantly influenced my work and inspired my growth as a researcher. His expert knowledge, as well as his ability to simplify complex mathematical concepts, have proven invaluable in enhancing my research career. His friendly and approachable nature, coupled with a keen understanding of the value of academic freedom, made every interaction with him not just a lesson in humility and grace, but also a testament to the respect he accords to individual research paths. His commitment to fostering an environment of exploration and freedom in academic pursuits significantly enhanced my ability to delve deeply and independently into my research topic.

I extend my sincere appreciation to **Prof. Chakravarthy Bhagvati.** His expert contribution, extensive knowledge, and sage advice have been instrumental in the successful completion of my research. His warm-hearted and

compassionate attitude, combined with his relentless pursuit of academic excellence, have been a source of continuous inspiration.

I find myself compelled to express my deep admiration for the **University of Hyderabad** campus. The vast, verdant landscapes and tranquil settings have provided a soothing backdrop for countless hours of study and research. The campus, in its quiet, majestic way, has fostered an environment conducive to intellectual growth and introspection. Truly, its beauty is beyond words, and I find myself fortunate to have spent a significant part of my academic journey in such a nurturing and visually stunning environment.

I offer my deep appreciation to **Prof. Atul Negi,** the Dean of the School of Computer and Information Science, University of Hyderabad. His sterling leadership, dedication, and unwavering support have nurtured an environment that promotes scholarly pursuit and academic excellence.

The **canteen** and **mess staff** at the **University of Hyderabad** deserve special mention for their hard work and dedication. Their consistent services have been an integral part of our daily lives. Also, I extend my gratitude to the maintenance staff of our hostel, who ensured that our living conditions were always kept optimal.

My **lab mates** have played an essential role in creating a stimulating, collaborative, and supportive research environment. Their insights, camaraderie, and shared experiences have considerably enriched my academic journey.

I deeply appreciate the **faculty** and **non-teaching** staff of the School of Computer and Information Science at the University of Hyderabad. Their professionalism, dedication, and consistent support have been a driving force in my academic journey.

All my **teachers** deserve recognition for their relentless commitment to imparting knowledge and instilling in me a lifelong passion for learning. Their teachings have shaped my intellectual growth and underpinned my academic achievements.

To my beloved parents, **Rolla Bhaskar** and **Polamma** your unconditional love, unwavering faith, and incalculable sacrifices have been the beacon guiding all my achievements. Your ceaseless support has anchored me throughout this journey.

I reserve special praise for my **friends**. Their unflinching faith in me, unyielding support, and regular encouragement have been a vital source of strength. In the rigorous pursuit of knowledge, their uplifting spirit, shared laughter, and endless optimism have been a sanctuary. Notably, their hands-on assistance and intellectual contributions to my research have been invaluable. This work is as much a testament to their friendship as it is to my perseverance.

I would like to extend my heartfelt appreciation to the **reviewers** of this paper. Their keen insights, beneficial critiques, and commitment to scholarly rigor have significantly enhanced the caliber and substance of this study.

Finally, I thank **God** for His unerring guidance, wisdom, and grace. His providence has been my strength during challenging times and my solace in moments of doubt.

The successful completion of this thesis would not have been possible without the collective support, faith, and contributions of everyone mentioned here, and many more unnamed. This thesis is a testament to your unwavering faith in me and my efforts. Thank you all.

ROLLA SUBRAHMANYAM

Abstract

The following research provides a comprehensive exploration of cryptographic protocols, focusing particularly on Centralized Secret Sharing Schemes and Distributed Group Key Agreement Protocols. The study's central goal revolves around the enhancement of privacy, security, and computational efficiency in both centralized and distributed environments.

A significant portion of this work is devoted to the development of a novel scheme in Centralized Secret Sharing, labeled the Multipartite Verifiable Secret Sharing Scheme. This scheme, uses the Chinese Remainder Theorem (CRT), such that the participants are equipped with an ability to validate their respective shares, so as to nullify any malicious dealer activity. We further contribute to this domain by proposing a novel Compartmented Proactive Secret Sharing Scheme, effectively addressing issues related to participant privacy, share renewal, and verification in the conventional Compartmented Secret Sharing Schemes (CSSS). This scheme enhances participant privacy while facilitating the renewal and validation of shares.

However, the concern of a single point of failure persists in centralized secret sharing schemes. In an effort to mitigate this, our research extends its scope to the realm of distributed group key agreement protocols. We introduce the Elliptic Curve Secret Sharing Scheme (ECSSS), specially designed for light weight distributed environments. The ECSSS is then incorporated into a newly developed Authenticated Distributed Group Key Agreement Protocol. This protocol demonstrates exceptional efficiency in managing key

distribution under limited resource conditions, demonstrating rapid computations, reduced key sizes, and minimal storage requirements.

In addition, we have pioneered a Multi Group Key Agreement Protocol using Secret Sharing Scheme, which paves the way for multiple groups to establish common keys for secure communication. An emphasis has been placed on the correctness and security analysis of these new schemes. The security of the centralized secret sharing schemes hinges on the discrete logarithm problem, while the distributed group key agreement protocols leverage the Elliptic Curve Discrete Logarithm Problem (ECDLP).

This work constitutes a significant contribution to the field of cryptography, introducing new techniques and protocols to bolster security, privacy, and efficiency. Although our research offers substantial advancements, potential areas for future research include examining share renewal in the Multipartite Verifiable Secret Sharing Scheme, and share recovery in the Compartmented Proactive Secret Sharing Scheme. Further research is also needed to understand Authenticated Distributed Group Key Agreement Protocol using ECSSS in a dynamic environment where participants join and leave the group dynamically to address scenarios where multiple groups meet the threshold in the Multi Group Key Agreement Protocols, potentially causing unintended secret reconstruction.

Contents

Chapter 1

Introduction

Information security has grown increasingly vital as electronic communications have become an integral part of our daily lives. The primary focus of security is to ensure the confidentiality, availability, and integrity of information, whether it is stored or transmitted. Secret sharing schemes serve as valuable methods for securing data in various cryptographic applications and for establishing additional security protocols.

Consider an intricate scenario involving a highly classified project, which includes a team of eleven highly specialized scientists. To ensure utmost confidentiality and to mitigate the risk of trust-related issues, the team decides to keep all project documentation under lock and key. This sensitive information is stored in a secure cabinet, which, by their design, can only be accessed if at least six of the scientists are present simultaneously.

This seemingly effective solution presents a couple of inherent challenges: the quantity of locks necessary and the number of keys each scientist must hold. According to the binomial coefficient formula, there would need to be $\binom{11}{6} = 462$ locks on the cabinet, and each scientist would have to carry $\binom{10}{5} = 252$ keys for it to work. This approach would certainly be impractical and unfeasible, given the huge number of keys and locks required.

Nevertheless, these complications can be adeptly tackled by employing a Secret Sharing Scheme (SSS). An SSS can dramatically simplify the security arrangement by

only requiring one lock for the cabinet and one key for each scientist. Thus, SSS provides an efficient, feasible solution to ensure secure access control in this high-security scenario. This method effectively balances the necessity of security with the practicalities of day-to-day operations, making it a compelling approach to solve the problem at hand.

1.1 Background and Context of the Research

A Secret Sharing Scheme distributes a secret among a group of participants, and few or all of them are required for the reconstruction of the secret. There are various types of secret sharing, including threshold sharing, which consists of dividing a secret into shares and reconstructing the secret based on the threshold number of shares. A subset of participants above the threshold can do the reconstruction of the secret while any number of participants below the threshold cannot reconstruct the secret. There are numerous applications for Secret Sharing Schemes, including key management, secure online auctions, fair exchange, secure multiparty computations, and secure voting systems.

In this thesis, we investigated the usage of Secret Sharing Schemes in both centralized and distributed environments. In the first part, Secret Sharing Schemes and their significance in a centralized environment is explored where information is distributed centrally, while in the second case, a distributed key agreement is formed between single and multiple groups using secret sharing techniques in a distributed environment is explored. The goal of this research is to advance state-of-the-art in secret sharing and key agreement protocols by improving the security and privacy of these schemes.

1.1.1 Centralized Secret Sharing Schemes

In a Centralized Secret Sharing Scheme, Dealer computes shares from a secret and distributes them to participants via a secure channel, then authorized or threshold participants can reconstruct the secret.

Shamir [70] introduced a Secret Sharing Scheme in 1979. Around the same time, Blakely [9] and Asmuth bloom also [4] introduced Secret Sharing Scheme. In these Secret Sharing Schemes, Dealer may be malicious and can send false shares to participants. With these shares, participants will reconstruct an incorrect secret. This affects the overall correctness of the scheme and the integrity of the secret as well. To address this issue, Feldman [34] , Iften [45], Qiong et al. [66], Kaya et al.[52], etc introduced a verifiable Secret Sharing Scheme.

Verifiable Secret Sharing Scheme [34] gives each participant his share and allows them to check the consistency of the share they received. If a participant receives a corrupted share from the Dealer, they should request a consistent share from the Dealer. Verifiable Secret Sharing Scheme (VSSS) helps to identify Dealer is malicious. The participant can use this technique to ensure that the Dealer he is dealing with is genuine. For long lived secrets, SSS and VSSS may not be sufficient as the adversary may slowly gain information about the secret from multiple locations. Herzberg et al. [43] introduced a proactive Secret Sharing Scheme where participants shares are periodically renewed while the secret remains the same. Unless all the information about the secret is gained in the same period, no adversary can break the secret.

Proactive Secret Sharing is nothing but share renewal plus Verifiable SSS. In all of the schemes above discussed, the secret is distributed to a single or unipartite level. Multipartite and Compartmented [32] secret sharing is introduced by Farras et al. [32] and Ghodosi et al. [32] respectively in order to enhance security and address the task of sharing secrets at multiple levels.

In Multipartite Secret Sharing [32], the Dealer splits the secret among multi levels. The secret is divided into shares and distributed to all levels, and only a sufficient number of shares are required to reconstruct the original secret. This type of scheme is aimed at providing a secure, controlled way to transmit sensitive information among a group with various levels. Various situations can benefit from this scheme, including business deals, military operations, and corporate governance.

Compartmented Secret Sharing Scheme [35] is a variant of Multipartite Secret Sharing. In Compartmented Secret Sharing Scheme [35], Dealer chooses the secret, and he computes the compartment or level secret from the secret. Then level secret is distributed to the corresponding level through a secure channel. In each level, threshold number of participants can reconstruct the compartment or level secret. Finally, all levels are required to reconstruct the original secret from their level secrets. A single level cannot rebuild the secret.

Two significant research gaps in the area of Centralized Secret Sharing are identified. The first gap pertains to Multipartite Secret Sharing Schemes [44], which currently lack mechanisms for verifying the consistency of the shares of the participants . There are potential vulnerabilities that could compromise the security of the shared secret as a result of this omission. Also, Compartmented Secret Sharing Scheme [35] suffer from lack of mechanisms for share verification and renewal. And, in a Multipartite Secret Sharing Scheme, a level number is made public which may compromise the privacy of the participants. These research gaps in centralized secret sharing must be addressed in order to improve their effectiveness and security.

All the above Centralized SSS can be used for key exchange protocols in a centralized environment, where the dealer can act as a key generation center. However, when it comes to a distributed environment, key agreement protocols are more popular.

1.1.2 Distributed Group Key Agreement Protocols

In Distributed Group Key Agreement Protocols [39] , there is no key generation center or Dealer. Only participants communicate with each other without any secure channel agree on the common key for communication.

A Key Agreement Protocol is a technique for safely creating a shared secret key via an unreliable communication channel. A key agreement protocol's objective is to enable two or more parties to decide on a shared secret key without risking the key's interception by a third party. Typically, public key cryptography is used to accomplish this; in this method, each side has a private key and a public key, and the shared secret key is created by fusing the two keys together. Key agreement methods are

frequently combined with other cryptographic protocols, including symmetric encryption and authentication, to offer secure communication. Diffie-Hellman (DH) [27] and Elliptic Curve Diffie-Hellman (ECDH) [38], and are a few examples of key agreement protocols. This key agreement protocols works for two participants, but not for a group of participants.

A Group Key Agreement Protocol is a procedure used in cryptography to create a shared secret key among a group of participants. The goal of employing a group key agreement protocol is to enable secure communication between group members without requiring each user pair to provide a separate key, These protocols are used to safeguard user communication in applications like group chats. These protocols promote security and privacy by enabling group members to create a shared secret key without the requirement of a centralized authority. Li et al.[59], Zhang [95], Cui et al.[23], Cao et al.[16], Alzahrani et al.[2], Cheng et al.[19], Zhang et al.[94], Sun et al. [79] and Lei zhang et al.[93] introduced various group key agreement protocols using various techniques such as bilinear pairings, polynomials, elliptic curve.

A Group Key Agreement Protocol using a Secret Sharing Scheme [39], particularly refers to a technique for creating shares of a secret key and distributing them to the participants. Only particular subsets of the group can reconstruct the key using their respective shares. This is frequently employed in circumstances where the shared key's security is relatively easy and the risk of a single point of failure is significant. Harn et al. [39] introduced group key agreement protocols using SSS in a distributed environment to achieve efficiency in terms of computational cost.

Although Secret Sharing Schemes are used for Distributed Key Agreement Protocols to achieve efficiency, there are few major research gaps. Firstly, no efficient lightweight group key agreement protocols with respect to computational cost are available in the literature. In addition to that, there is a dire need to look into the needs of multigroup environments. To the best of our knowledge, there are no key agreement protocols for multi group environments. Distributed Group Key Agreement Protocols Using Secret Sharing Schemes would be improved by addressing these research gaps.

1.2 Objectives of Research

The following are the objectives of our research

Centralized Secret Sharing Schemes

1. In a Multipartite Secret Sharing Scheme, malicious Dealer may send incorrect shares to participants. With those shares, participants can reconstruct the incorrect secret which may effect the correctness of the Secret Sharing Scheme . Hence, share verification is required in Multipartite Secret Sharing Schemes to ensure the consistency of the shares.

2. Multipartite Secret Sharing Schemes are proving to be inefficient for long lived secrets as the adversary may slowly gain information about the secret from various participants. Share renewal is required to ensure that no information is gained from the shares.

3. In a Compartmented Secret Sharing Scheme (a variant of Multipartite secret Sharing Schemes), the compartment/ level number is made public, thereby compromising the privacy of the participant. This information helps the adversary in gaining more knowledge about the secret.

Distributed Group Key Agreement Protocols

4. Although Group Key Agreement Protocols have been developed for distributed environments in the literature, efficient group key agreement protocol for lightweight environments with respect to computational cost is still lacking. Therefore, there is a dire need for an efficient group key agreement protocol that employs a mechanism tailored for lightweight environments.

5. In the literature, numerous Group Key Agreement Protocols have been developed for single groups. However, there is a noticeable absence of Multi Group Key Agreement Protocols. As a result, there is a need to establish key agreement protocols that cater to multi group environments as well.

1.3 Contributions of Research

A contribution of this research is the development of enhanced Centralized Secret Sharing Schemes and Distributed Grroup Key Agreement Protocols that improve security and privacy. Our research focuses primarily on addressing the identified research gaps, such as share verification in Multipartite Secret Sharing Schemes, participant privacy, share verification, and share renewal in Compartmented Secret Sharing Schemes, Group Key Agreement Protocols in a light weight environments, and Secret Sharing Schemes for Multiple Group Key Agreement Protocols.

In order to achieve the first research objective, we developed a Multipartite Verifiable Secret Sharing Scheme, which enables participants to verify the consistency of their shares after receiving them from dealers. This scheme will ensure the consistency of the shares received by the participants.

The second and third research objectives were addressed together by a Compartmented Proactive Secret Sharing Scheme, which provides privacy for the participants and also includes mechanisms for share verification and share renewal. All participants are able to verify and renew their shares as needed, and the privacy of the participant is achieved by his respective share.

The fourth research objective was addressed by an Authenticated Distributed Group Key Agreement Protocol using Elliptic Curve Secret Sharing Schemes. Elliptic curves are quite popular in lightweight environments to achieve security. However, their complex operations become a hindrance for usage in real time environments. We introduced Elliptic curve Secret Sharing Scheme to achieve efficiency and used it in Group Key Agreement Protocol that is more suitable for lightweight environments.

The final research objective was addressed by Multi Group Key Agreement Protocol using Secret Sharing Schemes for multiple groups. As a result of the incorporation of Secret Sharing Schemes into these protocols, group key agreements are made more secure. To the best of our knowledge, ours is the first Multi Group Key Agreement

Protocol that can ensure participants from multiple groups to share a key.

Ultimately, these contributions can help advance the state-of-the-art for Centralized Secret Sharing Schemes and Distributed Group Key Agreement Protocols using Secret Sharing Schemes by providing valuable insights and solutions.

1.4 Overall Structure of the Thesis

The thesis is organized into seven chapters.

In Chapter 1, Brief description of the introduction and context of research, objectives of the research, and significance of work and the overall structure of the thesis is presented.

In Chapter 2, The important mathematical primitives and basic cryptographic preliminaries that are utilized in the design of our proposed schemes are introduced and a literature survey of the proposed work is presented.

In Chapter 3, Multipartite Verifiable Secret Sharing based on CRT is proposed for share verification in Multipartite SSS. This chapter also contains the correctness and security analysis of the scheme as well as comparisons with other schemes.

In Chapter 4, Compartmented Proactive Secret Sharing Scheme is proposed that addresses share verification share renewal, and privacy of participant issues in Compartmented SSS. This chapter also includes the correctness and security analysis of the scheme.

In Chapter 5, Authenticated Distributed Group Key Agreement Protocol based on an Elliptic Curve Secret Sharing Scheme is proposed for lightweight environments. This chapter also contains the correctness and security analysis of the scheme as well as comparisons with other schemes.

In Chapter 6, Multi Group Key Agreement Protocol using Secret Sharing Scheme is proposed to share a secret among multiple groups. This chapter also contains the correctness and security analysis of the scheme as well as comparisons with other schemes.

In Chapter 7, The research efforts concluding remarks and future work are offered, as well as further extensions and future directions of the proposed scheme.

1.5 Publications

- Subrahmanyam, Rolla, N. Rukma Rekha, and Y. V. Subba Rao. "Multipartite verifiable secret sharing based on CRT." Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021. Springer Singapore, 2022. (Published, Index: Scopus) (Published, Index: Scopus)

- Subrahmanyam, Rolla, N. Rukma Rekha, and Y. V. Subba Rao. "Compartmented Proactive Secret Sharing Scheme." International Conference on Intelligent Sustainable Systems. Singapore: Springer Nature Singapore, 2023. (Published, Index: Scopus)

- Subrahmanyam, Rolla, N. Rukma Rekha, and YV Subba Rao. "Authenticated Distributed Group Key Agreement Protocol Using Elliptic Curve Secret Sharing Scheme." IEEE Access (2023). (Published, Index: SCIE)

- Subrahmanyam, Rolla, N. Rukma Rekha, and YV Subba Rao. (2023) 'Multi-Group Key Agreement Protocol Using Secret Sharing Scheme', Int. J. Security and Networks, (Accepted, Index : Scopus)

Chapter 2

Preliminaries and Literature Survey

In this chapter, we delve into the essential mathematical and cryptographic principles and perform a comprehensive review of existing literature that serves as the cornerstone for our proposed methodologies.

2.1 Lagrange Interpolating Polynomial(LPI)

A Lagrange interpolating polynomial passes through a set of given points in a plane and interpolates a function with them. Assuming n distinct points in the plane $(x_1, y_1), \cdots, (x_n, y_n)$ the Lagrange Interpolating Polynomial is,

$$g(x) = \sum_{i=1}^{n} L_i(x) y_i \tag{2.1}$$

where $L_i(x) = \frac{(x-1)\cdots(x-(i-1))(x-(i+1))\cdots(x-n)}{(i-1)\cdots(i-(i-1))(i-(i+1))\cdots(i-n)}$. Polynomial $g(x)$ has degree less than or equal to $n-1$ and is unique.

The Lagrange Interpolating Polynomial finds application in Chapter 4, which deals with the implementation of a Compartmented Proactive Secret Sharing Scheme. Furthermore, Chapter 7 elaborates on a Multigroup Key Agreement Protocol, which also employs a secret sharing scheme utilizing this mathematical technique.

2.2 Discrete Logarithm Problem (DLP)

The discrete logarithm problem (DLP) is a mathematical problem with significant cryptographic applications. In a modular arithmetic equation, the DLP is to find x from given a, p and b such that $a^x = b \pmod{p}$, where a, b, and p are all integers and p is a prime number.

In these systems, the security is predicated on the notion that finding x for given a, b, and p is computationally impossible. Compartmented Proactive Secret Sharing Scheme from Chapter 4, uses the discrete logarithm problem.

2.3 Chinese Remainder Theorem

A Chinese remainder theorem is defined as follows. Given n linear congruences,

$$\begin{aligned} x &\equiv a_1 \pmod{p_1} \\ x &\equiv a_2 \pmod{p_2} \\ &\vdots \\ x &\equiv a_n \pmod{p_n}. \end{aligned}$$

There exists a unique solution $X \pmod{M}$, where $M = p_1 \times p_2 \times ... \times p_n$, where $p_1, p_2, \cdots, p_n$ are pairwise relatively prime integers. Solution X can be expressed as follows:

$$X = a_1 M_1 y_1 + a_2 M_2 y_2 + ... + a_n M_n y_n \pmod{M}$$

where $M_1 = \frac{M}{p_1}, M_2 = \frac{M}{p_2}, \cdots, M_n = \frac{M}{p_n}$, and $y_1, y_2, \cdots, y_n$ are the modular inverses of $p_1, p_2, \cdots p_n$ respectively. The Chinese Remainder Theorem is used in the Multipartite Secret Sharing Scheme from Chapter 3.

2.4 Field

A field is a mathematical structure made up of the non-empty set $\mathbb{F}$ and the binary addition (+) and multiplication (.) operations that satisfy the following axioms:

- Closure: For all a, b in $\mathbb{F}$, both $a + b$ and $a.b$ are also in $\mathbb{F}$.
- Commutativity: For all a, b in $\mathbb{F}, a + b = b + a$ and $a.b = b.a$.
- Associativity: For all a, b, and c in $\mathbb{F}, (a+b)+c = a+(b+c)$ and $(a.b).c = a.(b.c)$.
- Existence of Identity elements: There are two elements in $\mathbb{F}, 0$ and 1, such that for all a in $\mathbb{F}, a + 0 = a$ and $a.1 = a$.
- Existence of Inverse elements: For each a in $\mathbb{F}$ other than 0, there is an element $-a$ and a^{-1} in $\mathbb{F}$ such that $a + (-a) = 0$ and $a.a^{-1} = 1$.
- Distributive property: For any a, b, and c in $\mathbb{F}$, $a.(b + c) = (a.b) + (a.c)$ and $(a + b)c = (a.c) + (b.c)$

These axioms guarantee that the field is predictable and has well-defined behavior as well as the existence of specific features in the set of elements and operations, as the existence of additive and multiplicative inverses for all elements other than zero. All chapters use field operation.

2.5 Elliptic Curves

Assume that $q > 3$ is a prime number, and we represent $\mathbb{F}_q$ as a field $\mathbb{F}_q$. A curve of the form $E : y^2 = x^3 + ax + b$ over $\mathbb{F}_q$ is an elliptic curve if the discriminant $\Delta = 4a^3 + 27b^2 \neq 0$, where a, b are constants. In addition to the point at infinity $\mathcal{O}$, the collection of all points on E over $\mathbb{F}_q$ are denoted as $E(\mathbb{F}_q)$. Point addition and doubling are operations on Elliptic curves. That are defined below:

2.5.1 Elliptic Curves Point Addition and Doubling

Let E be an elliptic curve, and P and Q be two points on it. Let R be sum of P and Q. The sum can be obtained as follows. Draw a line L passing through P and Q, which intersects the curve at another point, say R'. The reflection of R' about $x-$ axis is R, and we write $R = P + Q$. If P and Q are the same, the line L is tangent at P, and it intersects the curve at another point R'. The reflection of R' about $x-$axis is R, and we write $R = 2P$. Examples of point addition and point doubling on elliptic curves are depicted in 2.1 and 2.2, respectively.

Suppose that $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ then the formula for R is given below. Let $R = P + Q = (x_3, y_3)$
If $P \neq Q$, then

$$\begin{aligned} x_3 &= m^2 - x_1 - x_2 \\ y_3 &= m(x_1 - x_3) - y_1 \end{aligned}$$

where $m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$ is slope of line L passing through P and Q.

If $P = Q$, then $R = P + P$.

$$\begin{aligned} x_3 &= m^2 - 2x_1 \\ y_3 &= m(x_1 - x_3) - y_1 \end{aligned}$$

where $m = \frac{(3x_1^2 + a)}{(2y_1)}$ is slope of tangent line L at P. This is called as point doubling.

2.6 Elliptic Curve Discrete Logarithm Problem (ECDLP)

In the area of cryptography, the Elliptic Curve Discrete Logarithm Problem (ECDLP) is a crucial problem because it gives stronger security with a small key size. Let P and Q be points on elliptic curve E over field $\mathbb{F}$ such that $Q = kP$, where k is an integer.

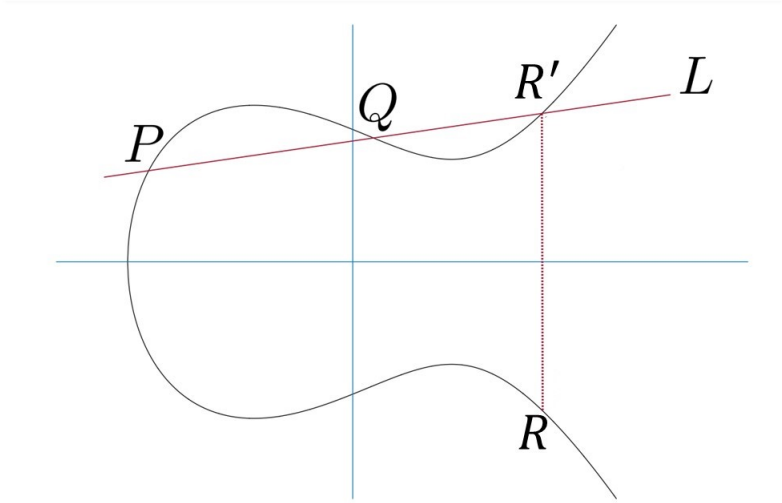

Figure 2.1: Point addition on Elliptic curve

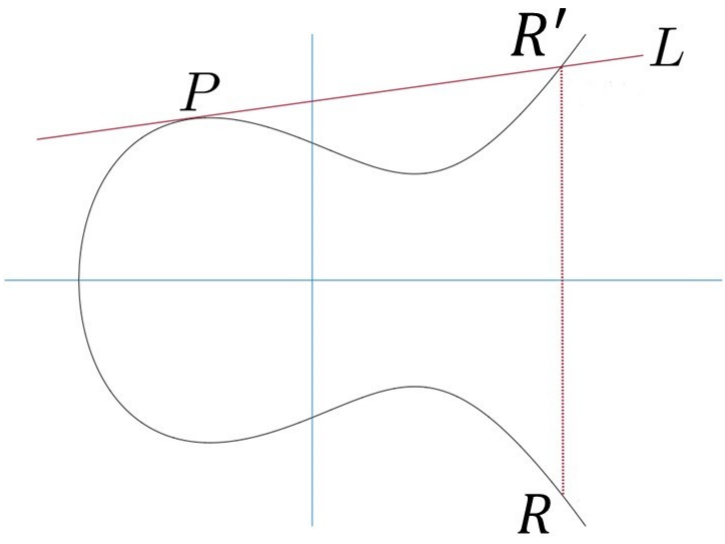

Figure 2.2: Point doubling on Elliptic curve

From given P and Q, computing k is difficult.

Numerous public key cryptographic systems hinge on the premise that solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) is computationally implausible for sufficiently large fields, which forms the foundation of their security. Key examples of such systems include the Elliptic Curve Digital Signature Algorithm (ECDSA) and the Elliptic Curve Diffie-Hellman (ECDH). This basically implies that even for highly capable attackers, determining the value of k given P and Q is considered to be an insurmountable computational task.

The Elliptic Curves Point Addition and Doubling operations along with ECDLP are applied in the Authenticated Distributed Group Key Agreement Protocol using Elliptic Curve Secret Sharing Scheme of chapter 6.

2.7 Vandermode Matrix

A vandermode matrix [54] is a matrix of the form

$$A = \begin{bmatrix} 1 & z_1 & z_1^2 & \cdots & z_1^{n-1} \\ 1 & z_2 & z_2^2 & \cdots & z_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_n & z_n^2 & \cdots & z_n^{n-1} \end{bmatrix}_{n \times n}$$

with $z_i \neq z_j$ for $i \neq j$. The matrix A is always invertible. Every vandermode matrix is invertible as the determinant of the matrix A is $\prod_{1 \leq i < j \leq n} (z_j - z_i)$, which is non-zero.

Vandermode Matrix is used in Authenticated Distributed Group Key Agreement Protocol using Elliptic Curve Secret Sharing Scheme of Chapter 5.

2.8 Secret Sharing Scheme

A secret sharing scheme is an advanced cryptographic protocol that enables the division of a single secret into various shares, which are then disseminated among a collection of participants. The unique characteristic of this scheme is that the original secret can only be regenerated by assembling a predetermined number of shares, this particular

number is referred to as the threshold. Typically, a secret sharing scheme encompasses two crucial algorithms.

An (n, n) secret sharing scheme, often known as a perfect secret sharing scheme, is where the secret is partitioned into n parts or shares and distributed amongst n participants. The intriguing facet of this scheme is that the secret can only be reconstructed if all n shares are pooled together. If even a single share is missing, the original secret remains concealed, thereby ensuring maximum security.

On the other hand, a (t, n) secret sharing scheme, also referred to as a threshold scheme, adds a layer of flexibility. Here, the secret is again split into n portions, but the key difference is that the original secret can now be reconstructed by any subset of t or more shares. The advantage here is that even if some participants are absent or their shares are lost, the secret can still be retrieved as long as at least t shares are present. This adds resiliency to the secret sharing scheme while still maintaining a high level of security.

2.8.1 Share Distribution

Dealer takes as input a secret S and generates a set of shares $\{share_1, share_2, \ldots, share_n\}$, where each participant receives one share.

2.8.2 Secret Reconstruction

The process of reconstruction involves utilizing a set of shares, which are possessed by t participants. This set is used to reassemble the initial secret S.

For a secret sharing scheme to be secure, it should satisfy the following properties[70]:

- Correctness: The secret can be correctly reconstructed using the required threshold number of shares.
- Privacy: Any unauthorized subset of shares should reveal no information about the secret.
- Uniqueness: Different secrets should result in different sets of shares.

- Robustness: The scheme should be resilient to the presence of malicious participants or share errors.

2.9 Perfect Secret Sharing Scheme

A Perfect Secret Sharing (PSS)[70] scheme is an essential cryptographic protocol designed to safeguard a secret by distributing its pieces among a group of participants. The defining characteristic of a PSS scheme is that it offers complete information-theoretic security.

Mathematically, a PSS scheme operates with a group of n participants. The secret, denoted as S, is divided into n shares in such a way that any subset of k or more shares can reconstruct the secret, whereas any subset of less than k shares has no information about the secret.

More formally, a PSS scheme has the following properties:

- **Reconstructability:** Any group of k or more shares can combine their pieces to perfectly reconstruct the secret S.
- **Security:** Any group with less than k shares cannot glean any information about the secret S. In other words, for any subset of less than k shares, all possible values of the secret S are equally likely.

In essence, a PSS scheme allows a secret to be shared among participants in a secure and robust manner, such that the secret can only be accessed by groups of a certain size, while smaller groups gain no information about the secret. This ensures the confidentiality of the secret against any unauthorized access.

2.9.1 Shamir's Secret Sharing Scheme

Shamir's Secret Sharing Scheme [70] is a well-known and widely used threshold secret sharing scheme proposed by Adi Shamir in 1979. It is based on polynomial interpolation over a finite field. Let S be the secret to be shared, and let t be the threshold, indicating the minimum number of participants required to reconstruct the secret. The

scheme operates in a finite field $\mathbb{F}$ with a prime order q. Shamir's secret sharing scheme consists of the following steps:

2.9.1.1 Secret Distribution

- Choose a random polynomial $f(x)$ of degree at most $t-1$, where $f(0) = S$.
- Select n distinct non-zero elements $x_1, x_2, \ldots, x_n$ from $\mathbb{F}$ as evaluation points.
- Compute the shares as $share_i = (x_i, f(x_i))$ for $i = 1, 2, \ldots, n$, and distribute one share to each participant.

2.9.1.2 Secret Reconstruction

- Collect at least t shares $\{(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_t}, y_{i_t})\}$ held by t participants.
- Use Lagrange interpolation to reconstruct the polynomial $f(x)$, which yields the secret $S = f(0)$.

Shamir's Secret Sharing Scheme guarantees the security properties of a secret sharing scheme, such as correctness, privacy, uniqueness, and robustness. It allows for flexible threshold settings and provides a reliable method for distributing and reconstructing secret among a group of participants.

2.9.2 Asmuth Bloom Secret Sharing Scheme

The Asmuth Bloom Secret Sharing Scheme [4] is a type of threshold secret sharing scheme proposed by Asmuth and Bloom in 1983. It allows a secret to be divided into shares, where each share is an integer value. The scheme is based on the Chinese Remainder Theorem and requires a trusted dealer to distribute the shares. Shamir secret sharing scheme takes $O(t \log^2 t)$ to construct secret, but Asmuth Bloom secret sharing scheme takes only $O(t)$ to build the secret. This scheme is also of two phases, Share Distribution, and Secret Reconstruction.

2.9.2.1 Share Distribution

- Dealer D distributes the secret S into n shares between members of a set $P = \{p_i : 1 \leq i \leq n\}$, known as the participant set P, each p_i being an individual participant.
- A set of integers $\{p, l_1, l_2, ..., l_n\}$ such that $l_i \leq l_j$ for $i < j$ and $0 \leq S < p$, is chosen with the following conditions:

  $\gcd(l_i, l_j) = 1$ where $i \neq j$, $\gcd(p, l_i) = 1$ for every i, and

  $$\prod_{i=1}^{t} l_i > p \prod_{i=1}^{t-1} l_{n-i+1}$$

- Let $M = \prod_{i=1}^{t} l_i$. Dealer computes $sh = S + Ap$, where $A \in Z^+$ is generated randomly with $0 \leq sh < M$.
- The i^{th} participant share is $sh_i = sh \bmod l_i$, $i = 1, 2, \cdots, n$.

2.9.2.2 Secret Reconstruction

- Suppose that a coalition C of t participants want to reconstruct the secret. Let $M_C = \prod_{i \in C} l_i$, and $sh \equiv sh_i \bmod l_i$ for $i \in C$.
- Reconstruct the the secret as $S = sh \bmod p$. sh is calculated uniquely in $GF(M_C)$ by using CRT. Since $sh < M \leq M_C$, the solution is also unique in $GF(M)$.

The Asmuth Bloom Secret Sharing Scheme and Shamir Secret Sharing Scheme provide security properties such as correctness, privacy, uniqueness, and robustness. It ensures that any unauthorized subset of shares reveals no information about the secret. However, it requires a trusted dealer for share distribution, which is a limitation of both the schemes. Shamir's Secret Sharing Scheme takes $O(t \log^2 t)$ to construct a secret, but Asmuth Bloom Secret Sharing Scheme takes only $O(t)$ to build the secret. The Asmuth-Bloom and Shamir Secret Sharing Schemes are useful in scenarios where a secure and efficient threshold Secret sharing scheme is required, and the trusted dealer

can be relied upon for share distribution.

In Asmuth Bloom and Shamir SSS(Secret Sharing Scheme), participants have no mechanism to verify their respective share from dealer for consistency. To overcome this aspect,verifiable Secret Sharing Schemes are proposed by various authors like Qiong, et al. [66] in 2005, Iftene [45] in 2007.

Iftene proposed VSSS(Verifiable Secret Sharing Scheme) in 2007 as an extension for Asmuth Bloom SSS each and every participant can verify their respective share and other shares after receiving shares from the dealer. The security of the scheme depends on discrete logarithm problem and the time complexity of secret reconstruction will be of order of $O(t)$ where t is a threshold.

2.9.3 Iftene CRT-based VSSS

To overcome the verifiability aspect in Asmuth Bloom, Iften proposed VSSS in 2007. In this scheme, each and every participant can verify their respective share and other shares after receiving shares from the dealer. This scheme is also made of two phases, Share Distribution and Secret Reconstruction.

2.9.3.1 Share Distribution

Dealer D computes the following steps

- A set of integers $\{l_0, l_1, l_2, \cdots, l_n\}$, such that $l_i \leq l_j$ for $i < j$ is chosen with the following conditions:
  $\gcd(l_i, l_j) = 1$ where $i \neq j$, and $\gcd(l_0, l_i) = 1$ for every i,
  Also, Let $M = \prod_{i=1}^{t} l_i$.
- Choose secret $S \in Z_{l_0}$.
- Computes $sh = S + Al_0 < M$, where A is positive integer.

- Computes share $sh_i = sh \bmod l_i$ for $i = 1, 2, \cdots, n$.
- Choose l_i' s such that each $p_i = 2l_i + 1$ is also a prime.
- Let $g_i \in Z_{p_i}^*$ of order l_i . The dealer distributes share sh_i to the i^{th} participants secretly and computes $c_i = g_i^{sh} \bmod p_i$ for $1 \leq i \leq n$. Here c_i, p_i and g_i are public.
- The i^{th} participant checks whether the share is valid or not by

$$c_i = g_i^{sh_i} \bmod p_i.$$

2.9.3.2 Secret Reconstruction

Suppose that a coalition C of participants wants to reconstruct the secret.

- Other participants in C can verify the i^{th} participant share with

$$c_i = g_i^{sh_i} \bmod p_i.$$

- The coalition C can reconstruct the secret S if all shares are correct.

2.9.3.3 Merit

The primary advantage of the Iften Verifiable Secret Sharing Scheme lies in its robust self-verification feature. This scheme allows each participant to independently authenticate their shares, effectively detecting any inconsistencies or malicious shares distributed by the dealer. This validation mechanism enhances the overall trust and security in the secret sharing process.

2.9.3.4 Demerit

However, the dealer may be dishonest in a scenario where the dealer chooses $sh > M$ then coalition C cannot get the correct sh and secret S value.

2.9.4 Kamer Kaya CRT-based VSSS

In both Iften [45] and Qiong [66] Verifiable Secret Sharing Scheme, the dealer may be dishonest and may send wrong shares to participants. With those shares authorized participants will never be able to get the actual secret. But in Kamer Kaya et al. CRT-based VSSS, the dealer sends shares secretly to participants and participants verify their respective share and check the range of $sh < M$ with their share by using Boudot range proof technique [52]. The security of the scheme depends on discrete logarithm problem. This scheme is also made of two phases, namely, Share Distribution, and Secret Reconstruction.

Secret Sharing Schemes are commonly classified into two broad categories: unipartite and multipartite.

2.9.5 Unipartite Secret Sharing Scheme

In unipartite secret sharing, all participants are considered as a single group. The fundamental principle here is that the secret can be reconstructed only when a certain number of shares, known as the threshold, are combined. The most popular form of unipartite secret sharing scheme are Shamir's [70], Asmuth [4], Iften [45], Qiong et al.[66] , Kamer Kaya et al.[52], etc.

For instance, if we have a group of 10 participants and the threshold is set to 7, then any combination of 7 participants can reconstruct the secret. If there are fewer than 7, they will not be able to reconstruct the secret, adding a layer of protection to the information.

2.9.6 Multipartite Secret Sharing Scheme

Multipartite Secret Sharing is a more complex variant, where participants are divided into multiple groups. The secret can be reconstructed only when certain conditions are met regarding the involvement of different groups. In other words, a specified number of shares from each group is required to reconstruct the secret.

For example, consider we have three groups - A, B, and C. In a multipartite scheme, we can set conditions such as "the secret can be reconstructed when at least 2 shares from group A, 3 shares from group B, and 1 share from group C are combined." It offers a more granular level of control and is especially useful in scenarios where hierarchical or departmental separations are necessary.

2.10 Access Structure

A secret sharing scheme determines the access structure by determining the minimum number of shares needed and the specific participants allowed to rebuild the secret. The secret can be reconstructed by a subset of individuals out of a total of n participants in a (t, n)-threshold sharing scheme.

A group of participants who have the ability to collectively piece together the secret information is often referred to as an authorized subset.

In formal terms, let $S = [1, 2, ..., n]$ represent the set of participants. It consists of a subset of participants, $A = [A_1, A_2, ..., A_m]$, where A_i is a non-empty subset of S, and the union of all A_i is S. A Secret Sharing Scheme satisfies access structure if participants can only reconstruct the secret if their shares belong to one of the authorized subsets A_i. As a result, it is computationally infeasible to reconstruct the secret from the shares held by the participants in A for any subset A of S that does not belong to any A_i. The literature describes various types of access structures, such as Multipartite, Compartmented and Hierarchical, that are defined below:

2.10.1 Multipartite Access Structure

Let $\mathcal{P}(P)$ stand for the power set of P. Further let $\Omega = \{P_1, \ldots, P_r\}$ be a partition of the set P, this means $\bigcup_{i=1}^{r} P_i = P$ and $P_i \cap P_j = \emptyset$ for any $1 \leq i < j \leq r$. Again, let σ be a special kind of permutation on P. Special in the sense that σ must map each member P_i of Ω onto itself; mathematically speaking $\sigma(P_i) = P_i$, $\forall P_i \in \Omega$. Let β_Ω denote the collection of all such permutations σ. Let Λ be a collection of subsets of P, so indeed $\Lambda \subseteq \mathcal{P}(P)$. Let $\sigma(\Lambda) = \{\sigma(A) : A \in \Lambda\}$, such that $\sigma(\Lambda) \subset \mathcal{P}(P)$. The collection

Λ is Ω-Partite if and only if the following condition holds: $\sigma \in \beta_\Omega \Rightarrow \sigma(\Lambda) = \Lambda$. Λ is said to be r-partite for any positive integer r if it is Ω-partite for some partition Ω on P of cardinality r [44].

Consider the set $J_r = \{1, 2, \cdots, r\}$. Let Z_+^r denote the set of vectors $u = (u_1, \cdots, u_r) \in Z^r$ with $u_i \geq 0$ for every $i \in J_r$. For a partition $\Omega = \{P_1, \cdots, P_r\}$ of P and Subset $A \subseteq P$ and $i \in J_r$. Define $\Omega_i(A)$ is the number of participants in $A \cap P_i$ i.e., $|A \cap P_i|$ then define a map $\Omega : \mathcal{P}(P) \to Z_+^r$ as $\Omega(A) = (\Omega_1(A), \Omega_2(A), \cdots, \Omega_r(A))$.

2.10.2 Compartmented Access Structure

Compartmented access structure [35] is a type of access structure in secret sharing schemes where participants are from multiple compartments, and the recovery of the secret requires a specific number of participants from each compartment. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ denote the set of n participants, and let $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ represent the compartments, such that $\mathcal{P} = \bigcup_{i=1}^{k} C_i$.

In a compartmented access structure, we define a set of thresholds $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$, where t_i represents the minimum number of participants required from compartment C_i to recover the secret. The thresholds must satisfy $1 \leq t_i \leq |C_i|$ for each i, where $1 \leq i \leq k$.

Formally, the access structure Γ of a compartmented secret sharing scheme is defined as:

$$\Gamma = \{\mathcal{A} \subseteq \mathcal{P} : |\mathcal{A} \cap C_i| \geq t_i \text{ for all } i\},$$

where $\mathcal{A}$ is a subset of participants and $|\mathcal{A} \cap C_i|$ represents the number of participants in $\mathcal{A}$ belonging to compartment C_i. The access structure Γ specifies the sets of participants that satisfy the threshold requirements for each compartment.

In a compartmented access structure, the reconstruction of the secret involves having at least t_i participants from each compartment C_i. If the number of participants in a compartment falls below its threshold, the secret cannot be reconstructed. This structure allows for fine-grained control over access and enables more complex sharing

schemes.

The notion of a compartmented access structure provides flexibility in designing secret sharing schemes that require specific combinations of participants from different compartments to access the secret.

2.10.3 Hierarchical Access Structure

Simmons (1988) [75] introduced the concept of hierarchical threshold secret sharing (HTSS), which differs from simple threshold secret sharing (SSS) by assigning different roles to participants. In a HTSS scheme, participants are categorized into distinct security levels. The participants at higher levels can collaborate with lower level participants to reconstruct the secret. HTSS is also referred to as a multilevel threshold secret sharing (MTSS) scheme.

Let's define the parameters of a HTSS scheme. We have a set of participants denoted by $\mathcal{P} = p_1, p_2, \ldots, p_n$, which is divided into m security levels represented as $\mathcal{L} = L_1, L_2, \ldots, L_m$. In other words, $\mathcal{P}$ is the union of all the security levels, $\mathcal{P} = \bigcup_{i=1}^{m} L_i$. Additionally, we have a sequence of threshold values $\mathcal{T} = t_1, t_2, \ldots, t_m$, where $t_1 < t_2 < \cdots < t_m$. Each t_i indicates the minimum number of participants required to recover the secret at level L_i. These threshold values satisfy the condition $1 \leq t_i \leq |L_1| + |L_2| \cdots + |L_i|$, where $1 \leq i \leq m$.

We can formally define the access structure Γ of the hierarchical threshold secret sharing scheme as follows: $\Gamma = \{\mathcal{A} \subseteq \mathcal{P} : |\mathcal{A} \cap (\bigcup_{j=1}^{i} L_j)|$. where $\mathcal{A}$ is a subset of participants and i represents the level, with $1 \leq i \leq m$. The access structure Γ specifies the sets of participants that satisfy the threshold requirements for each level.

In a HTSS scheme, the participants are divided into m levels, forming a hierarchical structure. Each level L_i consists of n_i participants and a secret can be recovered at level L_i if t_i or more participants are present. If the number of participants in level L_i falls below t_i, say r_i, the remaining $t_i - r_i$ participants can be selected from higher levels to meet the threshold requirement. It is important to note that throughout the

thesis, we assume that level L_i is higher than level L_{i+1} for $1 \leq i < m$.

To illustrate, suppose we have threshold values $t_1 = 3$ at level L_1 and $t_2 = 4$ at level L_2. In this case, three participants from L_1 or four participants from L_2 can collectively recover the secret. Additionally, the secret can also be recovered if there is one participant from L_1 and three participants from L_2.

Secret Sharing Schemes can be categorized as either one-time use or multi-use schemes. In a one-time use scheme, the shares become invalid after reconstructing the corresponding secret. Conversely, in a multi-use scheme, the shares can be reused for multiple secrets. Since the distribution of shares is a meticulous and costly process, the multi-use property has become crucial for secret sharing schemes.

In all previously mentioned schemes, the dealer has a higher level of authority as they calculate the shares and distribute them to the participants through a safe communication medium. This method favors the dealer, who is a single point of contact in a centralized environment. Now we explore the scenario in a distributed environment. In this context, each participant or user holds an equivalent level of importance, fostering a more equitable distribution of power. While there is no exact replication of SSS in a distributed environment, the same is achieved through group key agreement protocols. Group Key Agreement Protocols are available in both Centralized and Distributed environments. The scope of the thesis is limited to exploring Group Key Agreement Protocol in a distributed environment using SSS.

2.11 Group Key Agreement Protocols

A base group key agreement protocol [63] enables a group of participants to establish a shared secret key collectively. The protocol ensures that only authorized participants can obtain the shared key. Let $\mathcal{G}$ represent the group of participants, and let G denote the group's generator.

The base group key agreement protocol can be mathematically described as follows:

Key Generation

- Each participant $i \in \mathcal{G}$ selects a private key x_i.
- The corresponding public key for participant i is calculated as $X_i = G^{x_i}$.

Key Exchange

- Participants in $\mathcal{G}$ broadcast their public keys $\{X_i\}$ to all group members.
- Each participant i computes the shared secret key K_i as $K_i = \prod_{j \in \mathcal{G}} X_j^{x_i}$.

Key Confirmation

- Participants verify the validity of the shared key K_i using a predefined criterion or a specific validation process.

The base group key agreement protocol provides a secure method for establishing a shared key within a group. It ensures that only participants with valid private keys can compute the shared secret key. The protocol can be implemented using various cryptographic primitives, such as elliptic curve cryptography or Diffie-Hellman (DH) key exchange.

In secure multiparty computing, a group DH (Diffie Hellman) key is computed for a group of n members, each with a private key k_i, and calculates a function $f(k_1, k_2, \cdots, k_n)$ [36]. Tzeng and Tzeng[83, 84] introduced a round-efficient conference key with $f(k1, k2, \cdots, k_n) = g^{k1+k2+,\cdots,+k_n}$. This is an extension to Burmester and Desmedt [15] two round protocol with $f(k1, k2, \cdots, k_n) = g^{k_1k_2+,\cdots,+k_nk_1}$ which is round efficient but has a malicious participant attack. The basic DH key agreement protocol was substantially generalized for many group DH key approaches. This technique was utilized by Ingemarsson et al. [47], Steer et al. [77], Burmester and Desmedt [15], and Steiner et al.[77] to exchange group DH public keys by arranging group members in a logic ring. Burmester achieved computational efficiency by reducing the number of rounds from $(n-1)$ to 3. In contrast, Lee et al. [57] and Kim et al. [53, 74] used a binary tree to arrange group members and exchange DH public keys. In 1996, Steiner et al.[77] introduced the GDH (Group DH) Key Exchange (KE) as an extension to DH protocol.

Bresson et al. [12] protocol was improved with authentication services in 2001 and is proven secure. In 2006, Bohli [10] developed a scheme for secure group key agreement protocols, which allows unverified point-to-point networks to be secure against internal and external attackers.

Later, in 2007, Bresson et al.[12] developed a secure generic authenticated Group DH key exchange protocol. In 2007 itself, Katz and Yung [51] developed the first provably secure constant-round and completely scalable GDH protocol in the standard model. Brecher et al. [11] added robustness to the GDH protocol's tree-DH method by making it resistant to system failures, network outages, and member misconduct. Jarecki et al.[49] developed a group key agreement protocol that can withstand up to t node failures out of n nodes. Secure digital signatures are used to provide authentication for DH public keys. The computational cost of each group member is a crucial concern when implementing these protocols, especially when the group size is large.

Joux [50] first proposed to use pairings in a one-round tripartite key exchange. Later [6, 22, 28, 30, 73] developed several versions of authenticated group key exchange protocols. However, most of the pairing-based group KE methods are inefficient, as they result in a rise in the number of rounds as the group size increases. Choi et al.[22] developed a pairing-based group KE protocol that required a fixed number of rounds so that each member will be calculating two pairings and $4n$ modular exponentiation where n is a message .

A tree-based pairing-based group KE was developed by Barua et al.[30]. Du et al.[28] proposed an authenticated ID-based group KE mechanism with a constant number of rounds. In 2008, Desmedt and Lange [26] created a constant round pairing-based authenticated group KE that had lower processing complexity per member than previous protocols. Wu et al.[88] and Zhao et al.[95] developed an asymmetric group key agreement model that establishes secure networks for group communications. In a hierarchical access control system, Gu et al. [37] represented a group key agreement technique to speed up rekeying time. Konstantinou [55] developed an ID-based group key agreement protocol with efficient constant rounds for adhoc networks. Haiyan Sun [80] developed a new key agreement mechanism with provable security that does not

require the use of a certificate.

In certain group key transfer protocols, secret sharing is used for group key communication to achieve efficiency. Laih et al.[56] introduced the first group key transmission technique using a (t, n) secret sharing method in 1989. Each scheme participant must enroll with a conference chairperson and reveal a secret to the chairperson. The conference chairperson will select a random conference key as the secret and uses the secret-sharing mechanism to distribute shares of the secret among members. However, the number of users that the key can be shared is limited to $(t-1)$ users only out of n users. Berkovits [7], Li et al. [58], Saze [68] took a similar approach to distribute group communications to a large number of people securely. Cao et al.[16] developed a constant-round group KE protocol based on secret sharing with universally composable security. Harn and Lin [40] developed an authenticated group key transfer protocol based on the secret sharing technique. However, members do not have equal priority in this arrangement. To overcome this problem, Harn et al. [39] designed an efficient group DH key agreement protocol using a secret sharing scheme.

Yang et al.[89] introduced a key agreement-based elliptic curve scheme. It has no perfect forward secrecy, impression attack, and provable security. Yoon et al. [90] designed a key agreement protocol that has provable security but suffers from perfect forward secrecy. Debio et al.[25] developed a key agreement protocol with perfect forward secret secrecy that is provable and secure. In all these three schemes, communication happens between KGC and users. Chen et al.[19] introduced a key agreement protocol based on an elliptic curve where communication happens between server and tag in multiple rounds. After that, Liu Y SUn [62] introduced a key agreement protocol based on ECC (Elliptic curve cryptography). This protocol has one round of communication between the server and the tag. After that, Shen et al.[72] introduced that ECC could be used to create secure session keys between authorized users and devices. Based on ECC, Islam et al.[48] developed an unpaired authentication group key protocol. It minimizes computational costs and gets rid of public key certificates. In literature, Alzahrani et al. [2], Cui et al.[23], Sun et al. [79] introduced group key agreement protocols based on an elliptic curve where communication happens between

KGC and users.

Petr Dzurenda et al.[31] introduced an authenticated key agreement protocol based on secret sharing schemes. However, communication happens between KGC and users. Cheng et al.[19] introduced a group key agreement protocol based on bilinear pairing, which is suitable for mobile environments. Qikun Zhang [94] introduced a group key agreement protocol based on bilinear pairing. It has been shown that this protocol can withstand harmful attacks, such as active and passive attacks. Although Cheng et al. [19] and Zhang Qikum et al. [94] developed key agreement protocols without KGC, the cost involved is more. Hence we proposed an Authenticated Distributed Key Agreement Protocol using Elliptic Curve SSS to improve efficiency .

K. R. RAGHUNANDAN et al. [67] introduced an encryption scheme. The keys for encrypting the data are generated by the chaotic maps using pseudo-random numbers. Unauthorized users have a lot of difficulty accessing or changing the data because of the encryption procedure. One advantage of chaotic-map-based encryption is that it can be used on edge devices with low computing power. This scheme can be applied for encryption in a distributed environment.

Liu et al. [61], Binu et al. [8], and Wang et al. [86] introduced centralized secret sharing schemes based on elliptic curve and pairing. However, these schemes are not suitable for distributed environments. Liu Y et al. [62], Sheikh et al.[71] send shares through a public channel in a centralized environment which is a general tendency. However, to the best of our knowledge, other than Harn et al.[39], no secret sharing scheme sends shares through the public channel in a distributed environment.

2.11.1 Authenticated DH Group Key Agreement Protocol

A group with n users $(U_1, U_2, \cdots, U_n)$, want to construct group key S collaboratively in a public channel. It consists of the following Set Up and Phases.

Set Up

- Any user choose two primes p and q such that $p = 2q + 1$, and choose generator $\alpha \in \mathbb{F}_q$.

- Every user $U_i, i = 1, 2, \cdots, n$, chooses private keys k_i and x_i , computes $r_i = \alpha^{k_i}, y_i = \alpha^{x_i}$ and makes r_i and y_i public .

Secret Distribution

- Every user U_i computes shares $y_{ij} = (r_j y_j)^{x_i + k_i}, j = 1, 2, \cdots, n, j \neq i$.
- Every user U_i constructs key $K_i = \prod_{j=1, j \neq i}^{n} (r_j y_j)^{k_i + x_i}$.
- U_i constructs polynomial $f_i(x)$ of degree $n-1$ from n points: $(0, K_i), (r_j, y_{i1}), \cdots, (r_j, y_{ij}), \cdots, (r_j, y_{in}), j = 1, 2, \cdots, n, j \neq i$.
- U_i computes $n-1$ shares from $f_i(x), x = 1, 2, \cdots n-1$ and makes them public.

Key Reconstruction

- U_i computes his corresponding share from U_j, $y_{ji} = (r_j y_j)^{x_i + k_i}, j = 1, 2, \cdots, n, j \neq i$.
- U_i reconstructs key $K_j, j = 1, 2, \cdots, n, j \neq i$ of U_j from $n-1$ public shares and his respective share by using Lagrange's interpolation formula.
- U_i reconstructs group key $S = \prod_{j=1,}^{n} K_j$.

Authentication

- Each user U_i computes $C_i = h(S, U_i, r_i)$ and makes it public.
- Each user U_i computes $C_j{}' = h(S, U_j, r_j)$, for $j = 1, 2, \cdots, n$, $j \neq i$. If $C_j{}' = C_j$ the group key S is valid.

In the literature, elliptic curves are a great resource constrained environments that give equal amount of security with smaller key sizes. They can offer the same level of security as other methods, but with smaller key sizes. If the Secret Sharing Scheme (SSS) mechanism is applied with Elliptic Curve Cryptography (ECC), it could offer significant benefits in these resource-constrained contexts.

Harn et al.[39] scheme cannot be extended using the elliptic curve for secret sharing as we cannot reconstruct the polynomial from the pairs of points in the Secret Distribution. Hence a novel Authenticated Distributed Group Key Agreement Protocol using Elliptic Curve Secret Sharing Scheme is proposed in chapter 5.

2.11.2 Summary

This chapter provides a thorough exploration of the key mathematical theories and a detailed survey of related scholarly works, laying the groundwork for our suggested approaches.

Chapter 3

Multipartite Verifiable Secret Sharing based on CRT

3.1 Introduction

As a fundamental concept in cryptography, secret sharing involves the distribution of a secret among a set of participants in such a way that only a selected number of participants can collaborate on reconstructing it, while any smaller number cannot retrieve it. The concept of information security is vital to the protection of sensitive information and to ensuring that only authorized parties are able to access the information. As a result, conventional secret sharing schemes often assume that the dealer responsible for the distribution of shares can be trusted. If the dealer is malicious, this assumption may expose the system to vulnerability.

3.1.1 Problem Identification and Motivation:

A Verifiable Secret Sharing Scheme (VSSS) is necessary because a malicious dealer may distribute incorrect shares, which would prevent the participants from reassembling the secret. Many existing VSSS, such as those proposed by Iften [45], Qiong et al.[66], and Kaya[52], are based on the Chinese Remainder Theorem (CRT) as its efficient in reducing computational cost. Various VSSS based on CRT are available in literature for Unipartite environments. However, no such mechanism is available in literature for Multipartite environments. As a generalization of threshold secret sharing, Multipartite

Secret Sharing involves dividing participants into disjoint partitions and performing the same action on each partition.

3.1.2 Contribution:

This chapter proposes two Multipartite Verifiable Secret Sharing based on CRT [44] for verifying the credibility of dealers. The first scheme, Multipartite Verifiable Secret Sharing based on CRT using Iften's verifiability scheme, and the second scheme, Multipartite Verifiable Secret Sharing based on CRT using Kaya Verifiability scheme. These two schemes have phases namely, Share Distribution, Commitments, Share Verification and Secret Reconstruction. In the Share Distribution stage, the Dealer generates shares and confidentially transmits these to the respective participants. Next, during the Commitment phase, the Dealer determines the commitment and publicly discloses it. During the Share Verification phase, each participant confirms their individual shares using their specific commitments. Finally, in the Secret Reconstruction phase, a coalition number of participants has the ability to reconstruct the secret.

3.1.3 Notations and Assumptions

In Multipartite Secret Sharing based on CRT [44], participants cannot verify their respective shares after receiving shares from the dealer. To overcome this problem a novel Multipartite Verifiable Secret Sharing based on CRT is proposed where the participants can verify their respective shares and other shares as well. The scheme is proposed in two variants, the first variant uses Iftene VSSS(Verifiable Secret Sharing Scheme) [45] and second one uses Kamer Kaya VSSS [52].

The following notations are used in the proposed schemes.

- Let $P = \{p_i : 1 \leq i \leq n\}$ be set of participants, Dealer D and S is secret.
- Let τ be set of subsets of P, $\tau \subseteq 2^P$, subsets in τ are called authorized subsets.
- $\Delta = 2^P \backslash \tau$ is called prohibited access structure. Subsets in Δ are called unauthorized subsets.
- Super set of authorized set is again an authorized subset if it satisfies monotone increasing property. i.e, if $B \in \tau$ and $B \subseteq C \subseteq P$, then $C \in \tau$.

- τ_0 is minimal authorized subset τ

$$\tau = \{C \subseteq P : B \subseteq C, B \in \tau_0\}$$

- Δ_1 is maximal unauthorized subset

$$\Delta = \{C \subseteq P : C \subseteq B, B \in \Delta_1\}$$

- u_i and v_i are positive integers.

3.2 Proposed Multipartite VSSS Based on CRT by using Iftene's Verifiability Scheme

For a r partition $\Omega = \{P_1, \cdots, P_r\}$ of $P = \{p_i : 1 \leq i \leq n\}$, we suppose that an access structure τ is an Ω-partite, where $|P_1| = n_1, \cdots, |P_r| = n_r$ and $n_1+, \cdots, +n_r = n$. Then the partition is a transformation $\Omega : \mathcal{P}(P) \longrightarrow Z_+^r$. Let the corresponding minimal access structure is τ_0 and maximal prohibited access structure is Δ_1, so that $\Omega(\tau_0) \subset Z_+^r$ and $\Omega(\Delta_1) \subset Z_+^r$ can be determined. This consists of four phases, namely Share distribution, Commitment, Share Verification and Secret Reconstruction.

3.2.1 Share Distribution

Dealer D distributes n shares between members of a set $P = \{p_i : 1 \leq i \leq n\}$, known as the participant set , each p_i being an individual participant.

The dealer does the following:

- A set of integers
  $\{l_0, l_1 <, \cdots, < l_{n_1}, l_{n_1+1} <, \cdots, < l_{n_1+n_2}, \cdots, l_{n-n_r+1} <, \cdots, < l_n\}$,
  where $0 \leq S \leq l_0$, is chosen such that

  $\gcd(l_i, l_j) = 1 \quad$ where $i \neq j$.

- Computes

$$M_3 = \min\left(\prod_{j=1}^{r}\prod_{i=1}^{u_j} l_{s_{j-1}+i},\ \text{for all } (u_1, u_2, \cdots, u_r) \in \Omega(\tau_0)\right)$$

where $s_i = \sum_{j=1}^{i} n_j$ and $0 = s_0 < s_1 < s_2 < s_3 < s_4, \cdots, < s_r = n$.

- Computes

$$M_4 = \max\left(\prod_{j=1}^{r}\prod_{i=1}^{v_j} l_{s_{j-1}+i-1},\ \text{for all}(v_1, v_2, \cdots, v_r) \in \Omega(\Delta_1)\right)$$

$$M_3 > l_0 M_4.$$

- Verify if $M_3 > l_0 M_4$, else choose different set of l_i's.
- Dealer computes $sh = S + Ap$, where $A \in Z^+$ is generated randomly with $0 \leq sh < M_3$.
- Computes

$$sh_i = sh \bmod l_i, \quad i = 1, \cdots, n_j, j = 1, 2, \cdots, r$$

The n_j shares are distributed to each participants in P_j randomly $f : \{sh_1, \cdots, sh_n\} \longrightarrow P$.

3.2.2 Commitment

- Let $g_i \in Z_{p_i}^*$ of order l_i. The dealer computes

$$c_i = g_i^{sh} \bmod p_i. \tag{3.1}$$

Here c_i, p_i and g_i are public.

- The dealer carries out the computation of commitments c_i and publicly discloses them to allow for the verification of each participant's share.

3.2.3 Share Verification

- The i^{th} participant validates his share by checking iff

$$c_i \equiv g_i^{sh_i} \bmod p_i. \tag{3.2}$$

- Other participants can verify the i^{th} participant share by checking iff

$$c_i \equiv g_i^{sh_i} \bmod p_i.$$

3.2.4 Secret Reconstruction

- Suppose that a coalition C of τ participants want to reconstruct the secret. Let $M_C = \prod_{f(sh_i)\in C} l_i$ and $sh \equiv sh_i \bmod l_i$, for $f(sh_i) \in C$. Solve sh in $GF(M_C)$ uniquely using the CRT.

- Reconstruct the secret as $S = sh \bmod l_0$.

3.2.5 Proof of Correctness for Verification

Every participant can verify his respective shares as follows, checking iff

$$c_i \stackrel{?}{\equiv} g_i^{sh_i} \bmod p_i.$$

The correctness of the above equation can be seen as

$$\begin{aligned} c_i &\equiv g_i^{sh} \bmod p_i, (\text{from } 3.2) \\ &\equiv g_i^{sh_i + l_i a} \bmod p_i, \text{ since } sh_i = sh \bmod l_i, \text{where } a \text{ is positive integer} \\ &\equiv g_i^{sh_i} (g_i^{l_i})^a \bmod p_i \\ &\equiv g_i^{sh_i} \bmod p_i, \text{ since order of } g_i \text{ is } l_i. \end{aligned}$$

3.2.6 Security Analysis

Lemma 3.2.1. *Commitment $c_i = g_i^{sh} \bmod p_i$ does not leak any information about sh.*

Proof. Let G be a cyclic group of order q and g_i generates G. Let e be the identity element of G. Given $c_i = g_i^{sh} \bmod p_i$. Choose a random integer $a \in Z_{p_i'}$. Then for any $sh \in Z_p$, we have

$$c_i = g_i^{qa+sh} \bmod p_i$$

$$= g_i^{qa} g_i^{sh} \bmod p_i$$

$$= e^a g_i^{sh} \bmod p_i, \text{since } e = g_i^q$$

$$= g_i^{sh} \bmod p_i$$

Hence, $c_i = g_i^{sh} \bmod p_i$ is uniformally shared in G, i.e., the information about sh is secure enough.

□

Theorem 3.2.2. *The proposed Multipartite iftene VSSS realizing multipartite access structures is a perfect SSS.*

Proof. In our multipartite scheme, we get that sh can be computed uniquely in $GF(M_C)$ using CRT. the solution is unique in $GF(M_3)$ as $sh < M_3 < M_C$. Hence it satisfies that $H(S|C) = 0$, where H is entropy, for all $C \in \tau$ (authorized participants can able to reconstruct the secret). We consider that a coalition $C^{'}$ unauthorized participants in Δ has assembled. Let $sh^{'}$ denote the unique solution for $sh \in GF(M_{C'})$, hence $sh' + jM_{C'} \bmod l_0$ is smaller than M_3, for $0 \leq j < l_0$. From $M_3 > l_0 M_4$ and $M_4 > M_{C'}$, we get $\frac{M_3}{M_{C'}} > l_0$. For $0 \leq j < l_0$ all $sh^{'} + jM_{C'} \bmod l_0$ are different since $\gcd(M_{C'}, l_0) = 1$, and there l_0 such values exits. That is $S \in GF(l_0)$, and coalition participants cannot get any information about the secret. Hence it satisfy that $H(S|C') = H(S)$, for all $C' \in \Delta$(Any unauthorized participants can not get any information about the secret.) □

3.3 Proposed Multipartite VSSS Based on CRT by using Kamer Kaya's Verifiability Scheme

For a r partition $\Omega = \{P_1, \cdots, P_r\}$ of $P = \{p_i : 1 \leq i \leq n\}$, we suppose that an access structure τ is an Ω-partite, where $|P_1| = n_1, \cdots, |P_r| = n_r$ and $n_1 +, \cdots, +n_r = n$. Then the partition is a transformation $\Omega : \mathcal{P}(P) \longrightarrow Z_+^r$. Let the corresponding minimal access structure is τ_0 and maximal prohibited access structure is Δ_1, so that $\Omega(\tau_0) \subset Z_+^r$ and $\Omega(\Delta_1) \subset Z_+^r$ can be determined. It consists of four phases, namely Share Distribution, Commitment, Share Verification and Secret Reconstruction.

3.3.1 Share Distribution

Dealer D distributes n shares between members of a set $P = \{p_i : 1 \leq i \leq n\}$, known as the participant set , each p_i being an individual participant.

The dealer does the following:

- A set of integers
  $\{l_0, l_1 <, \cdots, < l_{n_1}, l_{n_1+1} <, \cdots, < l_{n_1+n_2}, \cdots, l_{n-n_r+1} < \cdots < l_n\}$,
  where $0 \leq S \leq l_0$, is chosen,
  where, $\gcd(l_i, l_j) = 1$ where $i \neq j$.

- Computes

$$M_3 = \min\Bigg(\prod_{j=1}^{r} \prod_{i=1}^{u_j} l_{s_{j-1}+i}, \text{for all } (u_1, u_2, \cdots, u_r) \in \Omega(\tau_0) \Bigg)$$

  where $s_i = \sum_{j=1}^{i} n_j$ and $0 = s_0 < s_1 < s_2 < s_3 < s_4, \cdots, < s_r = n$.

- Computes

$$M_4 = \max\Bigg(\prod_{j=1}^{r} \prod_{i=1}^{v_j} l_{s_{j-1}+i-1}, \text{ for all}(v_1, v_2, \cdots, v_r) \in \Omega(\Delta_1) \Bigg)$$

- Verify if $M_3 > l_0 M_4$, else choose different set of l_i's.

- Let $g_i \in Z^*_{p_i'}$ of order l_i. Let $P' = \prod_{i=1}^{n} p_i{}'$,

  Here $p_i{}' = 2l_i' + 1$, and $l_i{}'$' s both are large primes for $1 \leq i \leq n$.

  Choose $g \in Z_{P'}$ that is unique satisfying

$$g \equiv g_i \bmod p_i{}'.$$

- Dealer computes $sh = S + Al_0$, where $A \in Z^+$ is generated randomly with $0 \leq sh < M_3$.

- Computes

$$sh_i = sh \bmod l_i, \quad i = 1, \cdots, n_j, j = 1, 2, \cdots, r$$

The n_j shares are distributed to each participants in P_j randomly $f : \{sh_1, \cdots, sh_n\} \longrightarrow P$.

3.3.2 Commitment

- Assume both dealer and participant do not know N prime factorization. Compute

$$E = g^{sh} \bmod P^{'}N. \tag{3.3}$$

3.3.3 Share Verification

- i^{th} participant checks whether the share is valid or not by checking iff $E \equiv g_i^{sh_i} \bmod p_i{'}$ to verify $sh_i = sh \bmod l_i$. Then participants can verify validity of the range proof by checking $sh < M_3$.

- Other participants can verify the i^{th} participant share with verification equation by checking iff

$$g_i^{sh_i} \equiv E \bmod p_i^{'}.$$

3.3.4 Secret Reconstruction

- Suppose that a coalition C of τ participants want to reconstruct the secret. Let $M_C = \prod_{f(sh_i) \epsilon C} l_i$, and $sh \equiv sh_i \bmod l_i$, for $f(sh_i) \in C$. Solve sh in $GF(M_C)$ uniquely using the CRT.

- Compute the secret as $S = sh \bmod l_0$.

3.3.5 Proof of Correctness for Verification

Every participant can verify his respective share checking iff

$$g_i^{sh_i} \stackrel{?}{\equiv} E \bmod \ {p_i}'.$$

In this equation, the correctness for share verification can be seen as follows:

$$\begin{aligned} E \bmod p_i^{'} &\equiv g^{sh} \bmod P^{'}N \bmod {p_i}' \text{ (from 3.3)} \\ &\equiv g_i^{sh} \bmod p_i^{'}, \text{ since } \ g_i \equiv g \bmod p_i^{'} \\ &\equiv g_i^{sh_i + l_i a} \bmod p_i^{'}, \text{ since } sh_i = sh \bmod l_i, a \text{ is positive integer} \\ &\equiv g_i^{sh_i}(g_i^{l_i})^a \bmod p_i^{'} \\ &\equiv g_i^{sh_i} \bmod {p_i}', \text{ since order of } g_i \text{ is } l_i. \end{aligned}$$

3.3.6 Security Analysis

Lemma 3.3.1. *Commitment $E = g^{sh}$ mod $P'N$ does not reveal any information about sh.*

Proof. Let G be a cyclic group of order q and g generates G. Let e be the identity element of G. Given $E = g^{sh} \ \bmod \ P'N$. Choose a random integer $a \in Z_{p_i'}$. Then for any $sh \in Z_p$. We have

$$E = g^{qa+sh}$$

$$= g^{qa} g^{sh}$$

$$= e^a g^{sh}$$

$$= g^{sh}$$

Hence, $E = g^{sh} \bmod P'N$ is uniformally shared in G, i.e., the information about sh is secure enough.

□

Theorem 3.3.2. *The proposed Multipartite Kamer Kaya, et al. VSSS realizing multipartite access structures is a perfect SSS .*

Proof. Proof is same as theorem 3.2.2. □

3.4 Comparison with Existing Schemes

The presented table, as illustrated in Table 3.1, offers a detailed comparison of six distinctive secret sharing schemes, all of which are underpinned by the Chinese Remainder Theorem. The main characteristics scrutinized in these schemes include their partitioning nature (unipartite or multipartite), verifiability, and resilience against malicious dealer behavior.

The Asmuth-Bloom SSS underpinned by CRT is noted as a unipartite scheme, however, it lacks both share verifiability and resistance against dealer malevolence. The Verifiable Secret Sharing Scheme (VSSS) proposed by Iften , albeit still unipartite, introduces share verifiability. Yet, similar to the Asmuth-Bloom SSS, it fails to counteract malicious dealer activities.

In contrast, the unipartite VSSS scheme presented by Kameer Kaya et al. embeds both share verifiability and the ability to withstand malevolent dealers, demonstrating an advancement over the previous models. The Multipartite SSS leveraging CRT diverges from the previous three schemes as it operates in a multipartite fashion, and while it can resist dealer malevolence, it lacks the verifiability feature.

A similar absence of protection against malicious dealers is observed in the Multipartite VSSS as per Iften , although this multipartite scheme does offer verifiability. Finally, the Multipartite VSSS as per Kameer Kaya et al. combines multipartiteness, share verifiability, and resilience against malicious dealers, culminating in a comprehensive solution.

In essence, Table 3.1 delineates the strengths and limitations of each scheme, indicating that the choice of a scheme would greatly depend on the specific requirements of partitioning nature, verifiability, and resistance to malicious dealer activities.

Scheme Name	Unipartite	Multipartite	Verifiability	Resistant to malicious dealer
Asmuth Bloom SSS based on CRT	Yes	No	No	No
Iften VSSS based on CRT	Yes	No	Yes	No
Kameer kaya et al. VSSS based on CRT	Yes	No	Yes	Yes
Multipartite SSS Based on CRT	No	Yes	No	Yes
Multipartite VSSS Based on CRT by using Iften, et al.	No	Yes	Yes	No
Multipartite VSSS Based on CRT by using Kameer Kaya et al.	No	Yes	Yes	Yes

Table 3.1: Comparision table

3.5 Summary

Two schemes namely Multipartite VSSS Based on CRT by using Iften verifiable scheme, and Multipartite VSSS Based on CRT by using Kamer Kaya verifiable scheme are proposed. In the first scheme, the dealer may be malicious because if participants can not choose his share $sh < M_3$. However, in the second scheme, the participant can overcome this demerit by verifying if his share $sh < M_3$. The first scheme do not have that provision. Both schemes are perfectly secure, and the security of the schemes depends on discrete logarithm problem.

In both Multipartite VSSS Based on CRT by using Iften's scheme and Multipartite VSSS Based on CRT by using Kamer Kaya's schemes, works for centralized environments where dealer is the centralized person who generates shares and distributes them. Our future work is to extend this scheme to Share Renewal in Multipartite VSSS based on CRT.

Chapter 4

Compartmented Proactive Secret Sharing based on Polynomial

4.1 Introduction

Secret Sharing Schemes (SSS) play an important role in ensuring that sensitive information is accessible only to authorized parties in secure information storage and sharing systems. A Compartmented Secret Sharing Scheme (CSSS) [35] divides participants into distinct compartments, each compartment has its own threshold that must be met in order to reconstruct the secret. However, conventional CSSS methodologies introduce a set of limitations and vulnerabilities that can pose significant threats to the privacy and security of participants. A key concern is the requirement for the Dealer, who is responsible for distributing shares to participants, to publicly disclose the compartment number ℓ_i. This information can provide valuable insights into the distribution and structure of the secret sharing scheme, potentially enabling unauthorized entities to mount successful attacks against the system.

Moreover, the inability of participants to verify or renew their shares in a conventional CSSS is another considerable drawback. This lack of self-verification and renewal procedures allows for scenarios where a malicious or compromised Dealer can distribute incorrect or manipulated shares without detection. Consequently, it introduces opportunities for exploitation and increases the likelihood of successful attacks against the system.

Motivated by these challenges, this chapter proposes an enhanced Compartmented Proactive Secret Sharing Scheme. This advanced framework aims to address the identified issues, specifically focusing on improving participant privacy, enabling share verification, and facilitating share renewal. The proposed scheme provides a comprehensive solution to the noted vulnerabilities of traditional CSSS, thus promising a significantly more secure, reliable, and robust secret sharing mechanism.

4.1.1 Problem Identification and Motivation

The central challenge in contemporary Compartmented Secret Sharing Schemes (CSSS) resides in their design and operational shortcomings that can undermine the privacy of participants and expose the system to a range of security vulnerabilities. One of the primary concerns stems from the requirement for the Dealer to publicly reveal the compartment number ℓ_i. This practice, although necessary in traditional CSSS, could inadvertently expose sensitive structural information about the secret sharing scheme. As a result, it could potentially offer an advantage to unauthorized entities seeking to compromise the system.

Additionally, participants in a standard CSSS lack the ability to verify their individual shares. This absence of verification mechanisms means participants must inherently trust the Dealer. Unfortunately, if the Dealer is compromised or acts maliciously, the participants are at risk of receiving tampered or incorrect shares, severely threatening the integrity of the system.

Further amplifying these issues is the absence of a provision for participants to renew their shares. The lack of a share renewal mechanism is problematic as it could lead to stagnation of the share distribution. This static distribution could be exploited over time by an adversary, given sufficient computational power and time.

These combined weaknesses present compelling motivation to revisit and revise the traditional CSSS. It is imperative to introduce new mechanisms to ensure participant privacy, allow share verification, and provide a mechanism for share renewal. By addressing these issues, the proposed Compartmented Proactive Secret Sharing

Scheme strives to significantly enhance the security and robustness of the secret sharing paradigm. Through this, it aims to offer a more resilient and reliable solution for secure information storage and sharing systems.

4.1.2 Contribution

In order to address the issues namely Privacy of participants, Share Verification and Share Renewal associated with existing CSSS schemes, this chapter proposes a Compartmented Proactive Secret Sharing Scheme, that has the following characteristics:

1. The compartment number is made private in order to ensure the privacy of participants.
2. The compartment number is further protected by revealing it only through the share of each participant.
3. Participants can verify their shares and can assure that their shares are consistent.
4. Shares of participants are periodically renewed through share renewals.

The proposed scheme can be used in the following circumstances: Let's assume a Company CEO(Dealer) has a top-secret S, and his company has ℓ department's. The CEO chooses the secret s, and then he computes partial secret s_i for all departments using that secret. Every department partial secret s_i is computed by the CEO and he distributes shares of partial secret secretly to all participants in that compartment i. Every participant receives a share from the CEO and then uses that share to calculate his department number ℓ_i. After that, a threshold number of participants t_i come together to reconstruct the partial secret in each department. Finally, the secret can be reconstructed by a global threshold number of participants. Next, CSSS proposed scheme is explained in section 4.2.

4.2 Proposed Compartmented Proactive Secret Sharing Scheme

In Compartmented Secret Sharing Scheme, the compartment number is made public, and because of this other participant knows the compartment number of the participant.

So, participant has no Privacy in compartmented SSS. Hence we proposed Compartmented Proactive SSS to address this issue. In addition, we add Share Verification and Share Renewal features to our proposed scheme. The scheme is explained below.

Assume that there is a group $P = \{p_1, p_2, \cdots, p_N\}$ of N participants. The participants are disjointly partitioned into ℓ compartments, such as $\ell_1, \ell_2, \cdots, \ell_\ell$. Let t_i be the threshold and n_i be the total number of participants in ℓ_i. Then $N = \sum_{i=1}^{\ell} n_i$ and we denote $t = \sum_{i=1}^{\ell} t_i$ as global threshold . However, ℓ_i (compartment number) is private and n_i is public in this scheme.

This scheme comprises six phases: Share Distribution, Commitments, Share Verification, Compartment Number Computation, Share Renewal, and Secret Reconstruction.

During Share Distribution phase, Dealer chooses $t-1$ degree polynomial, from which shares of level secret s_i are computed, and then distributes them through secure channels to participants in level ℓ_i. Next, the Dealer computes a public value. In Commitment phase, the Dealer computes commitments and makes them public. In Share Verification phase, each participant verifies their respective share whether it is valid or not by using commitments. In Compartment Number Computation phase, each participant computes his level number from public value and his corresponding share. In Share Renewal phase, each participant renewed his share after some interval time without changing the secret. In Secret Reconstruction phase, the threshold number of participants get to reassemble the secret. The above phases are explained below in detail.

4.2.1 Share Distribution

- The Dealer chooses $\ell - 1$ degree polynomial.

$$g(x) = S + a_1 x + a_2 x^2 + \cdots + a_{\ell-1} x^{\ell-1} \bmod q \tag{4.1}$$

  where $a_i \in \mathbb{F}_q$, secret $S = g(0)$ and q is prime .

- Each participant has $id_{ij},\ 1 \leq i \leq \ell,\ 1 \leq j \leq n_i$.

- Dealer computes compartment secret s_i, $s_i = g(i) \bmod q,\ 1 \leq i \leq \ell$.

- Dealer chooses a polynomial of degree $t_i - 1$ for each compartment $\ell_i,\ 1 \leq i \leq \ell$

$$h_i(x) = s_i + c_{i,1}x + c_{i,2}x^2 + c_{i,3}x^3 + \cdots + c_{i,t_i-1}x^{t_i-1} \bmod q, c_{i,1}, c_{i,2}, \cdots, c_{i,t_i-1} \in \mathbb{F}_q. \tag{4.2}$$

- Dealer computes shares(sh_{ij})for participants in every compartment and sends via secure channel.

$$sh_{ij} = h_i(id_{ij}) \bmod q,\ 1 \leq i \leq \ell,\ 1 \leq j \leq n_i. \tag{4.3}$$

- Dealer computes a public value

$$k_{ij} = sh_{ij} + sh_{ij}^{-1} + \ell_i id_{ij} \bmod q. \tag{4.4}$$

4.2.2 Commitments

- Dealer chooses a large prime p such that $q \mid (p-1)$ and primitive root a of subgroup of $\mathbb{Z}_p^*$ of order q and computes commitments cm_{i0}, cm_{ij} as

$$cm_{i0} = a^{s_i} \bmod p \text{ for } 1 \leq i \leq \ell. \tag{4.5}$$

$$cm_{ij} = a^{c_{ik}} \bmod p \text{ for } 1 \leq i \leq \ell,\ 1 \leq j \leq n_i,\ 1 \leq k \leq t-1. \tag{4.6}$$

- Dealer makes cm_{i0}, cm_{ij} values public.

4.2.3 Share Verification

- Participants can verify validity of their respective share sh_{ij} by checking iff

$$a^{sh_{ij}} = (cm_{i0})(cm_{i1})^{id_{ij}} \cdots (cm_{i(t-1)})^{{id_{ij}}^{t-1}} \bmod p,$$

where $1 \leq i \leq \ell,\ 1 \leq j \leq n_i$.

4.2.4 Compartment Number Computation

- Each participant receives his share sh_{ij} secretly and public information k_{ij}.

- Each participant computes his compartment number ℓ_i as
  $\ell_i = id_{ij}^{-1}(k_{ij} - sh_{ij} - sh_{ij}^{-1}) \bmod q$.

4.2.5 Share Renewal

- Each participant chooses a polynomial of degree $t_i - 1$ in every compartment

$$d_{ij}(x) = c_{ij,1}x + c_{ij,2}x^2 + c_{ij,3}x^3 + \cdots + c_{ij,t_i-1}x^{t_i-1} \bmod q \text{ ,where}$$
$$1 \le i \le \ell,\ 1 \le j \le n_i,\ d_{ij}(0) = 0 \text{ and } c_{ij,1}, c_{ij,2}, \cdots, c_{ij,t_i-1} \in \mathbb{F}_q.$$

- Each participant computes $r_{i,k,j}$ and distribute to all participants secretly

$$r_{i,k,j} = d_{ij}(id_{ik}) \bmod q,\ 1 \le i \le \ell,\ 1 \le j \le n_i,\ 1 \le k \le n_i. \tag{4.7}$$

- Each participant computes sh'_{ij} as

$$sh'_{ij} = \sum_{k=1}^{n_i} r_{i,j,k} \bmod q,\ 1 \le i \le \ell,\ 1 \le j \le n_i. \tag{4.8}$$

- Each participant gets his new share nsh_{ij} as

$$nsh_{ij} = sh_{ij} + sh'_{ij} \bmod q,\ 1 \le i \le \ell,\ 1 \le j \le n_i. \tag{4.9}$$

4.2.6 Secret Reconstruction

- In every compartment (t_i, n_i) participant can recover their compartment secret s_i with help of their respective shares by using Lagrange's interpolation formula 2.1.

- Then participants can get back their polynomial $g(x)$ by using Lagrange's interpolation formula , hence secret $S = g(0)$.

4.3 Numerical Problem

Suppose there is a Dealer who wants to share a secret S among a group P of $N = 7$ participants, disjointly partitioned into $\ell = 2$ compartments: Compartment 1 with $t_1 = 2$ participants, $n_1 = 3$ and Compartment 2 with $t_2 = 3$ participants, $n_2 = 4$. The Dealer has chosen prime numbers $q = 17$, $p = 53$, and $a = 3$, which is a primitive root of $\mathbb{Z}_p^*$ of order q.

The Dealer chooses a degree $\ell - 1 = 1$ polynomial for secret S:

$g(x) = S + a_1 x \bmod q$

Let's say that $S = 5$ and $a_1 = 3$. So, the polynomial is $g(x) = 5 + 3x \bmod 17$.

Each participant is assigned an identification number id_{ij}. For simplicity, let's assume that for Compartment 1, $id_{11} = 1, id_{12} = 2, id_{13} = 3$ and for Compartment 2, $id_{21} = 8, id_{22} = 5, id_{23} = 6, id_{24} = 7$

The Dealer computes the compartment secret s_i for each compartment:

$s_1 = g(1) = 5 + 3 * 1 \bmod 17 = 8 \bmod 17 = 8$,

$s_2 = g(2) = 5 + 3 * 2 \bmod 17 = 11 \bmod 17 = 11$.

The Dealer chooses a polynomial of degree $t_i - 1$ for each compartment ℓ_i.

For ℓ_1, we have a degree $t_1 - 1 = 1$ polynomial:

$h_1(x) = s_1 + c_{1,1}x \bmod q$.

Let's say $c_{1,1} = 2$. So, the polynomial is $h_1(x) = 8 + 2x \bmod 17$.

For ℓ_2, we have a degree $t_2 - 1 = 2$ polynomial:

$h_2(x) = s_2 + c_{2,1}x + c_{2,2}x^2 \bmod q$.

Let's say $c_{2,1} = 3$ and $c_{2,2} = 4$. So, the polynomial is $h_2(x) = 11 + 3x + 4x^2 \bmod 17$.

The Dealer computes the shares (sh_{ij}) for participants in each compartment and sends them securely:

For Compartment 1 Shares: $sh_{11} = h_1(1) = 8 + 2(1) \bmod 17 = 10$, $sh_{12} = h_1(2) = 8 + 2(2) \bmod 17 = 12$, $sh_{13} = h_1(3) = 8 + 2(3) \bmod 17 = 14$.

For Compartment 2 Shares: $sh_{21} = h_2(8) = 11 + 3(8) + 4(8^2) \bmod 17 = 2$, $sh_{22} = h_2(5) = 11 + 3(5) + 4(5^2) \bmod 17 = 7$, $sh_{23} = h_2(6) = 11 + 3(6) + 4(6^2) \bmod 17 = 3$, $sh_{24} = h_2(7) = 11 + 3(7) + 4(7^2) \bmod 17 = 7$.

The Dealer computes a public value for each participant k_{ij}:

For Compartment 1:

$$\begin{aligned} k_{11} &= sh_{11} + sh_{11}^{-1} + \ell_1 \cdot id_{11} \mod 17 \\ &= 10 + 10^{-1} + 2 \cdot 1 \bmod 17 = 7. \end{aligned}$$

$$\begin{aligned} k_{12} &= sh_{12} + sh_{12}^{-1} + \ell_1 \cdot id_{12} \mod 17 \\ &= 12 + 12^{-1} + 2 \cdot 2 \bmod 17 = 9. \end{aligned}$$

$$\begin{aligned} k_{13} &= sh_{13} + sh_{13}^{-1} + \ell_1 \cdot id_{13} \mod 17 \\ &= 14 + 14^{-1} + 2 \cdot 3 \bmod 17 = 1. \end{aligned}$$

For Compartment 2:

$$\begin{aligned} k_{21} &= sh_{21} + sh_{21}^{-1} + \ell_2 \cdot id_{21} \mod 17 \\ &= 2 + 2^{-1} + 2 \cdot 8 \mod 17 = 12. \end{aligned}$$

$$\begin{aligned} k_{22} &= sh_2 + sh_{21}^{-1} + \ell_2 \cdot id_{22} \mod 17 \\ &= 7 + 7^{-1} + 2 \cdot 5 \mod 17 = 5. \end{aligned}$$

$$\begin{aligned} k_{23} &= sh_{23} + sh_{23}^{-1} + \ell_2 \cdot id_{23} \mod 17 \\ &= 3 + 3^{-1} + 2 \cdot 6 \bmod 17 = 4. \end{aligned}$$

$$k_{24} = sh_{24} + sh_{24}^{-1} + \ell_2 \cdot id_{24} \mod 17$$
$$= 7 + 7^{-1} + 2 \cdot 7 \bmod 17 = 9.$$

For Compartment 1: $k_{11} = 7, k_{12} = 9, k_{13} = 1$.
For Compartment 2: $k_{21} = 12, k_{12} = 5, k_{23} = 4, k_{24} = 9$.

4.3.1 Commitments

For Compartment 1 (ℓ_1), the commitments from equations 4.5 and 4.6 are computed based on the provided equations. Let's compute them:
$cm_{10} = a^{s_1} \bmod p = 3^8 \bmod 53 = 8$
$cm_{11} = a^{c_{1,1}} \bmod p = 3^2 \bmod 53 = 9$
So, for Compartment 1, the commitments are $cm_{10} = 8$ and $cm_{11} = 9$.

For Compartment 2 (ℓ_2), again, we compute the commitments based on the equations:
$cm_{20} = a^{s_2} \bmod p = 3^{11} \bmod 53 = 47$
$cm_{21} = a^{c_{2,1}} \bmod p = 3^3 \bmod 53 = 27$
$cm_{22} = a^{c_{2,2}} \bmod p = 3^4 \bmod 53 = 14$
So, for Compartment 2, the commitments are $cm_{20} = 47$, $cm_{21} = 27$, and $cm_{22} = 14$.

The dealer will make public the following commitments:
For compartment 1: $cm_{10} = 8$, $cm_{11} = 9$
For compartment 2: $cm_{20} = 47$, $cm_{21} = 27$, $cm_{22} = 14$

4.3.2 Compartment Number Computation

Let's consider a scenario where we have participant 2, who belongs to compartment 1, and they wish to determine their corresponding compartment number. The participant possesses the following information:

Share: $sh_{12} = 12$. Public information: $k_{12} = 9$

To calculate the participant's compartment number, we can utilize the given formula:

$\ell_i = id_{ij}^{-1}(k_{ij} - sh_{ij} - sh_{ij}^{-1}) \bmod q$

By substituting the provided values into the formula, we obtain:

$\ell_i = 1^{-1}(9 - 12 - 12^{-1}) \bmod 17 = 1$

Thus, participant 2, who belongs to compartment 1, can determine their respective compartment number by using their share value of 12 and the public information value of 9. In this case, the calculated compartment number is 1.

Similarly, each participant calculates their own respective compartment number.

4.3.3 Share Verification

To verify the validity of participant 2's share, denoted as $sh_{12} = 12$, we can utilize the following equation:

$a^{sh_{ij}} = (cm_{i0})(cm_{i1})^{id_{ij}} \cdots (cm_{i(t-1)})^{{id_{ij}}^{t-1}} \bmod p,$

where the given values are as follows:

Participant id: $id_{ij} = 2$

Compartment number: $i = 1$

Share value: $sh_{ij} = 12$

$p = 53, a = 3$

We can substitute these values into the equation and check its validity:

$a^{12} = (cm_{10})(cm_{11})^2 \bmod 53$

Commitments: $(cm_{10}) = 8, (cm_{11}) = 9$

$a^{12} = 3^{12} \equiv 10648 \bmod 53 \equiv 35 \bmod 53$

Now, compare both sides of the equation:

$35 \equiv (8)(9^2) \bmod 53$

$\equiv 8 \cdot 81 \bmod 53$

$\equiv 648 \bmod 53$

$\equiv 35 \bmod 53$

Since both sides of the equation are equal, the share $sh_{12} = 12$ is verified as valid.

Similarly, other participants should confirm whether their respective shares are valid or not.

4.3.4 Share Renewal

For Compartment 1 Shares:

$sh_{11} = 10$, $sh_{12} = 12$, $sh_{13} = 14$.

For Compartment 2 Shares: $sh_{21} = 2$, $sh_{22} = 7$, $sh_{23} = 3$, $sh_{24} = 7$.

For Compartment 1:

$d_{11}(x) = 9x$

$d_{12}(x) = 3x$

$d_{13}(x) = 4x$

For Compartment 2:

$d_{21}(x) = 5x$

$d_{22}(x) = 6x$

$d_{23}(x) = 9x$

$d_{24}(x) = 7x$

Share renewal values:

For Compartment 1:

$d_{11}(1) = 9 * 1 = 9 \mod 17 = 9$

$d_{11}(2) = 9 * 2 = 18 \mod 17 = 1$

$d_{11}(3) = 9 * 3 = 27 \mod 17 = 10$

$d_{11}(8) = 9 * 8 = 72 \mod 17 = 4$

$d_{11}(5) = 9 * 5 = 45 \mod 17 = 11$

$d_{11}(6) = 9 * 6 = 54 \mod 17 = 3$

$d_{11}(7) = 9 * 7 = 63 \mod 17 = 12.$

$d_{12}(1) = 3 * 1 = 3 \mod 17 = 3$

$d_{12}(2) = 3 * 2 = 6 \mod 17 = 6$

$d_{12}(3) = 3 * 3 = 9 \mod 17 = 9$

$d_{12}(8) = 3 * 8 = 24 \mod 17 = 7$

$d_{12}(5) = 3 * 5 = 15 \mod 17 = 15$

$d_{12}(6) = 3 * 6 = 18 \mod 17 = 1$
$d_{12}(7) = 3 * 7 = 21 \mod 17 = 4$

$d_{13}(1) = 4 * 1 = 4 \mod 17 = 4$
$d_{13}(2) = 4 * 2 = 8 \mod 17 = 8$
$d_{13}(3) = 4 * 3 = 12 \mod 17 = 12$
$d_{13}(8) = 4 * 8 = 32 \mod 17 = 15$
$d_{13}(5) = 4 * 5 = 20 \mod 17 = 3$
$d_{13}(6) = 4 * 6 = 24 \mod 17 = 7$
$d_{13}(7) = 4 * 7 = 28 \mod 17 = 11$

For Compartment 2:
$d_{21}(1) = 5 * 1 = 5 \mod 17 = 5$
$d_{21}(2) = 5 * 2 = 10 \mod 17 = 10$
$d_{21}(3) = 5 * 3 = 15 \mod 17 = 15$
$d_{21}(8) = 5 * 8 = 40 \mod 17 = 6$
$d_{21}(5) = 5 * 5 = 25 \mod 17 = 8$
$d_{21}(6) = 5 * 6 = 30 \mod 17 = 13$
$d_{21}(7) = 5 * 7 = 35 \mod 17 = 1$

$d_{22}(1) = 6 * 1 = 6 \mod 17 = 6$
$d_{22}(2) = 6 * 2 = 12 \mod 17 = 12$
$d_{22}(3) = 6 * 3 = 18 \mod 17 = 1$
$d_{22}(8) = 6 * 8 = 48 \mod 17 = 14$
$d_{22}(5) = 6 * 5 = 30 \mod 17 = 13$
$d_{22}(6) = 6 * 6 = 36 \mod 17 = 2$
$d_{22}(7) = 6 * 7 = 42 \mod 17 = 8$

$d_{22}(1) = 6 * 1 = 6 \mod 17 = 6$
$d_{22}(2) = 6 * 2 = 12 \mod 17 = 12$
$d_{22}(3) = 6 * 3 = 18 \mod 17 = 1$
$d_{22}(8) = 6 * 8 = 48 \mod 17 = 14$
$d_{22}(5) = 6 * 5 = 30 \mod 17 = 13$

$d_{22}(6) = 6 * 6 = 36 \mod 17 = 2$

$d_{22}(7) = 6 * 7 = 42 \mod 17 = 8$

$d_{24}(1) = 7 * 1 = 7 \mod 17 = 7$

$d_{24}(2) = 7 * 2 = 14 \mod 17 = 14$

$d_{24}(3) = 7 * 3 = 21 \mod 17 = 4$

$d_{24}(8) = 7 * 8 = 56 \mod 17 = 5$

$d_{24}(5) = 7 * 5 = 35 \mod 17 = 1$

$d_{24}(6) = 7 * 6 = 42 \mod 17 = 8$

$d_{24}(7) = 7 * 7 = 49 \mod 17 = 15$

sh'_{11}: $(9 + 3 + 4 + 5 + 6 + 9 + 7) \mod 17 = 43 \mod 17 = 9$

sh'_{12}: $(1 + 6 + 8 + 10 + 12 + 1 + 14) \mod 17 = 52 \mod 17 = 1$

sh'_{13}: $(10 + 9 + 12 + 15 + 1 + 10 + 4) \mod 17 = 61 \mod 17 = 10$

sh'_{21}: $(4 + 7 + 15 + 6 + 14 + 4 + 5) \mod 17 = 55 \mod 17 = 4$

sh'_{22}: $(11 + 15 + 3 + 8 + 13 + 11 + 1) \mod 17 = 62 \mod 17 = 11$

sh'_{23}: $(3 + 1 + 7 + 13 + 2 + 3 + 8) \mod 17 = 37 \mod 17 = 3$

sh'_{24}: $(12 + 4 + 11 + 1 + 8 + 12 + 15) \mod 17 = 63 \mod 17 = 12$

For Compartment 1 new shares: $nsh_{11} = 2$, $nsh_{12} = 13$, $nsh_{13} = 7$.

For Compartment 2 new shares: $nsh_{21} = 6$, $nsh_{22} = 1$, $nsh_{23} = 3$, $nsh_{24} = 2$.

4.3.5 Secret Reconstruction

Using Lagrange's interpolation 2.1, the threshold number of participants($t = t_1 + t_2 = 2 + 3$) can reconstruct the secret by combining their old and new shares.

Old threshold number of shares: $sh_{11} = 10, sh_{12} = 12, sh_{22} = 7, sh_{23} = 3, sh_{24} = 7$.

The secret $S = 5$.

New threshold number of shares:

$nsh_{11} = 2, nsh_{12} = 13, nsh_{22} = 1, nsh_{23} = 3, nsh_{24} = 2$. The secret $S = 5$.

4.4 Correctness and Security Analysis of the scheme

This section explains the correctness and security analysis of Compartment Number Computation, Share Verification and Share Renewal.

4.4.1 Correctness and Security Analysis for Compartment Number

Each participant calculates their compartment number as follows:

$\ell_i = id_{ij}^{-1}(k_{ij} - sh_{ij} - sh_{ij}^{-1}) \bmod q.$

$$\begin{aligned} & id_{ij}^{-1}(k_{ij} - sh_{ij} - sh_{ij}^{-1}) \bmod q \\ \equiv\ & id_{ij}^{-1}(sh_{ij} + sh_{ij}^{-1} + \ell_i id_{ij} - sh_{ij} - sh_{ij}^{-1}) \bmod q, \text{ from equation 4.4} \\ \equiv\ & id_{ij}^{-1}\ell_i id_{ij} \bmod q \\ \equiv\ & \ell_i \bmod q. \end{aligned}$$

The probability of choosing correct sh_{ij} is $\frac{1}{q}$. As q is large prime, the probability is very less/negligible. From knowing k_{ij} and q getting ℓ_i is difficult without knowledge of sh_{ij} for an adversary.

4.4.2 Correctness and Security Analysis for share verification

Participants verify their respective share by using cm_{i0}, cm_{ij} as below:

$$a^{sh_{ij}} = (cm_{i0})(cm_{i1})^{id_{ij}} \cdots (cm_{i(t-1)})^{{id_{ij}}^{t-1}} \bmod p,$$

where $1 \leq i \leq \ell,\ 1 \leq j \leq n_i$.

$$\begin{aligned} & (cm_{i0})(cm_{i1})^{id_{ij}}(cm_{i2})^{{id_{ij}}^2} \cdots (cm_{i(t-1)})^{{id_{ij}}^{t-1}} \bmod p \\ \equiv\ & (a^{s_i})(a^{c_{i1}})^{id_{ij}}(a^{c_{i2}})^{{id_{ij}}^2} \cdots (a^{c_{i(t-1)}})^{{id_{ij}}^{t-1}} \bmod p, \text{from equation 4.5 and equation 4.6} \\ \equiv\ & a^{s_i + c_{i1}id_{ij} + c_{i2}{id_{ij}}^2 + \cdots + c_{i(t-1)}{id_{ij}}^{t-1}} \bmod p \\ \equiv\ & a^{g_i(id_{ij})} \bmod p, \text{ from equation 4.2} \\ \equiv\ & a^{sh_{ij}} \bmod p, \text{ from equation 4.3}. \end{aligned}$$

Here cm_{i0} and cm_{ij} values alone don't reveal any information about secret because revealing those values down to solving discrete logarithm problem.

4.4.3 Correctness and Security Analysis for Share renewal

The new share nsh_{ij} of each participant in compartment ℓ_i is

$$\begin{aligned}
nsh_{ij} &= sh_{ij} + sh'_{ij} \bmod q, \text{ from equation } 4.9 \\
&= sh_{ij} + \sum_{k=1}^{n_i} r_{i,j,k} \bmod q,\ 1 \le i \le \ell,\ 1 \le j \le n_i, \text{from equation } 4.7 \\
&= h_i(id_{ij}) + \sum_{k=1}^{n_i} d_{ik}(id_{ij}) \bmod q, \text{from equation } 4.7 \\
&= s_i + \sum_{k=1}^{t_i-1} c_{i,k} {id_{ij}}^k + \sum_{k=1}^{t_i-1} c_{i1,k} {id_{ij}}^k + \cdots + \sum_{k=1}^{t_i-1} c_{in_i,k} {id_{ij}}^k \bmod q, \text{from equation } 4.2 \\
&= s_i + \Big(c_{i,1} + \sum_{k=1}^{n_i} c_{ik,1}\Big) id_{ij} \cdots + \Big(c_{i,t_i-1} + \sum_{k=1}^{n_i} c_{ik,t_i-1}\Big) {id_{ij}}^{t_i-1} \bmod q.
\end{aligned}$$

From each compartment ℓ_i, t_i or more participants can combine to rebuild the compartment secret s_i using the Lagrange's interpolation formula

Thus the compartment secret s_i will be revealed. The threshold number of participants or more can reconstruct the secret. While less than threshold number of participants or more cannot get the secret.

The probability of getting the correct secret S is $\frac{1}{q}$ as $S \in \mathbb{F}_q$ is random.

4.5 Conclusion

This chapter proposed a Compartmented Proactive Secret Sharing Scheme. In this scheme, a threshold number of participants are necessary to reconstruct the secret in each compartment. As part of this proposal, each user can verify his own share consistency. Moreover, these shares are updated periodically, making it difficult for unauthorized parties to recreate the hidden information. The secret cannot be reconstructed in any compartment if there is less than the threshold number of participants.

This scheme is offered to improve participant privacy because the compartment number is known only from his respective share and Share Verifiability and Share Renewal features were added to enhance the security of the scheme.

Chapter 5

Authenticated Distributed Group Key Agreement Protocol using Elliptic Curve SSS

5.1 Introduction

In a centralised system, we encounter a significant challenge known as the single point of failure. This issue arises because everything relies on one central element, and if that part fails, the whole system can collapse. To avoid this risk, we often shift to a distributed environment. In a distributed framework, many different parts or individuals contribute to the system, making it less reliant on one central part. This shared responsibility greatly reduces the chance of the entire system breaking down if one part has a problem.

An important aspect of these distributed systems is the consensus on a group key agreement protocol. This is a collective decision made by all members of the system to use a shared key. This key is critical for maintaining secure communication within the system, allowing everyone to work together efficiently and securely. By moving from a centralised to a distributed system and agreeing on a group key, we are able to avoid the single point of failure issue, resulting in a more reliable system.

In distributed environments where participants are located in different locations, group

key agreement protocols are crucial for securing communication within groups. A common group key is agreed upon by group members, and this key is then used to facilitate secure communication among the members of the group. In resource-constrained environments, the need for efficient and secure key agreement protocols has become more pressing than ever due to the rapid expansion of Internet of Things (IoT) applications. A popular choice for these environments is elliptic curve cryptography (ECC), which provides sufficient security with smaller key sizes.

Problem Identification and Motivation

There have been numerous techniques used in the literature to distribute group key-related information among group members, including polynomials, bilinear pairings, and secret sharing schemes (SSSs). These methods have demonstrated considerable utility in fostering a secure environment for communication by distributing keys efficiently among users in a network. However, while these approaches are undoubtedly effective, they present certain challenges when deployed in Internet of Things (IoT) or other resource-constrained environments. The constraints posed by IoT devices, such as limited computing power, storage capacity, and energy supply, make traditional Secret Sharing Scheme less efficient. These environments require a balance between high-level security and the reduction of computational overhead, storage requirements, and power consumption. In other words, resource-constrained environments demand highly efficient security protocols that maintain robust security while consuming minimal resources.

With the aim to address this challenge, there has been a substantial enhancement in the efficiency of these cryptographic techniques. Notably, the advent and use of secret sharing schemes have greatly improved key distribution in terms of computational and communication overheads.

Contributions:

The following are our contributions in this chapter:

- A novel Elliptic Curve Secret Sharing Scheme (ECSSS) is proposed for resource constrained environments.
- Authenticated Distributed Group Key Agreement Protocol using Elliptic Curve Secret Sharing Scheme (ADGKAP) is proposed to achieve equal security with relatively smaller key size, storage, fast computation.
- To the best of our knowledge, no Authenticated Distributed Group Key Agreement Protocol has used Elliptic Curve Secret Sharing Scheme for share distribution till date.

The salient features of the ADGKAP are as below:

- A novel Elliptic Curve Secret Sharing Scheme is designed to generate points that are used as shares by the users.
- All the shares of the scheme are shared through a public channel in a distributed environment.
- After the group key is reconstructed by each individual user, the user can verify the authentication of the group key by comparing the hash of his reconstructed key with the hash of the other user's reconstructed key.
- ADGKAP is proven secure enough provided the elliptic curve discrete logarithm problem is intractable. Proposed ADGKAP provides equal security with a smaller key size, less storage, faster computation, and less computational cost without compromising on the number of rounds. In this ADGKAP, each group user can reconstruct the key individually, but an attacker cannot reconstruct the key.

5.2 Notations

- Users, $(U_1, U_2, \cdots, U_n)$, where n is number of users.
- Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{F}_q$, q is prime power.
- Point $P \in E(\mathbb{F}_q)$ and

  $G =< P >$ be a group of order ℓ, where ℓ is prime.

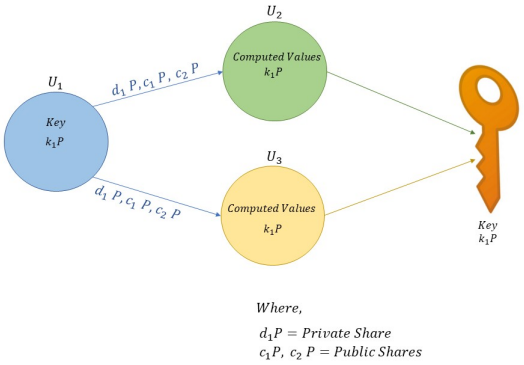

Figure 5.1: Framework of ECSSS

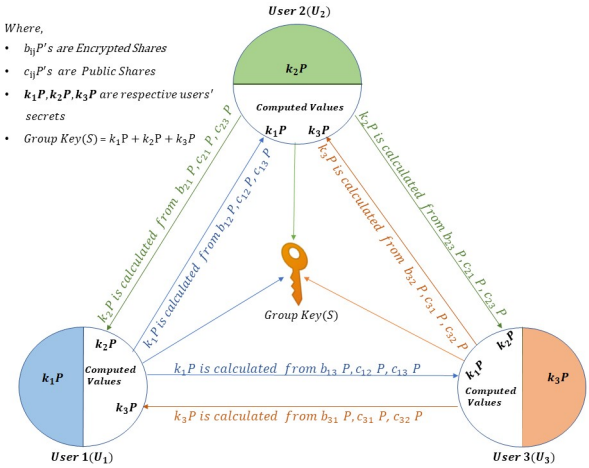

Figure 5.2: Framework of ADGKAP

- Any User U_i makes values E, q, P, and ℓ public.
- $r_1, r_2, \cdots, r_{(n-1)}, k_1 \in [1, \ell - 1]$ are random values.
- User U_1 key is k_1P, User U_2 is key is k_2P, and User U_n key is k_nP.
- $b_{ij}P$ are encrypted shares and $c_{ij}P$ are public shares, for i^{th} user and $j = 1, 2 \cdots, n, j \neq i$.
- h is SHA 256 hashing function.

5.3 Proposed Elliptic Curve Secret Sharing Scheme (EC-SSS)

ECSSS has two steps namely , Secret Distribution and Key Reconstruction. In Secret Distribution, user U_1 computes n shares of his key k_1P generated from elliptic curve E. Out of n shares, $n-1$ shares are made public, and 1 is kept as a private share. The private share is sent to $n-1$ users through a secure channel. In Key Reconstruction, U_2, U_3, U_4 reconstructs key k_1P. The ECSSS and the Correctness of the key reconstruction are explained in sections 5.3 and 5.3.3, respectively.

Assume that user U_1 wants to secretly send the shares of key, k_1P to $n-1$ users $(U_2, U_3, U_4, \cdots, U_n)$.

5.3.1 Secret Distribution

- Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{F}_q$, q is prime power.
- Point $P \in E(\mathbb{F}_q)$ and

  $G =< P >$ be a group of order ℓ, where ℓ is prime.
- User U_1 makes values E, q, P, and ℓ public.
- User U_1 chooses a $n \times n$ vandermonde matrix A_1 and makes it public

$$A_1 = \begin{bmatrix} 1 & z_1 & z_1^2 & \cdots & z_1^{n-1} \\ 1 & z_2 & z_2^2 & \cdots & z_2^{n-1} \\ 1 & z_3 & z_3^2 & \cdots & z_3^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_n & z_n^2 & \cdots & z_n^{n-1} \end{bmatrix}_{n \times n}$$

- U_1 selects values, $r_1, r_2, \cdots, r_{(n-1)}, k_1 \in [1, \ell - 1]$ randomly.
- U_1 calculates $d_1, c_1, \cdots, c_{(n-1)}$ as follows,
  $(d_1, c_1, c_2, \cdots, c_{(n-1)})^T = A_1(k_1, r_1, r_2 \cdots, r_{(n-1)})^T$.
- U_1 calculates $(d_1P, c_1P, c_2P, \cdots, c_{(n-1)}P)^T$.
- U_1 sends share d_1P secretly to $n-1$ users through secure channel and makes the remaining $n-1$ shares $c_1P, c_2P, \cdots, c_{(n-1)}P$ as public .

5.3.2 Key Reconstruction

- Each user $U_i, i = 2, 3, \cdots, n$, can reconstruct the key k_1P by computing ${A_1}^{-1}(d_1P, c_1P,\ c_2P, \cdots, c_{(n-1)}P)^T$, which is equivalent to $(k_1P, r_1P, r_2P, \cdots, r_{(n-1)}P)^T$.

Note 1:

Each user U_i has shares $d_1P, c_1P,\ c_2P, \cdots, c_{(n-1)}P$.

That is, $A_1(k_1P, r_1P, r_2P, \cdots, r_{(n-1)}P)^T = (d_1P, c_1P, c_2P, \cdots, c_{(n-1)}P)^T$.

Figure 5.1 describes ECSSS as follows: Suppose there are three users in the scheme. User U_1 chooses a key k_1P, and computes shares: one private share d_1P and public shares c_1P and c_2P. User U_1 sends private share to U_2, U_3 through a secure channel. Finally, U_2, U_3, reconstruct the key k_1P from d_1P and c_1P and c_2P.

5.3.3 Correctness of the Key Reconstruction

Each user $U_i,\ i = 2, 3, \cdots, n$ has n shares: one private share d_1P,and $n-1$ public shares $c_1P,\ c_2P, \cdots, c_{(n-1)}P$. The user U_i computes ${A_1}^{-1}$ and multiply that with shares $(d_1P, c_1P, c_2P, \cdots, c_{(n-1)}P)^T$. i.e

${A_1}^{-1}(d_1P, c_1P,\ c_2P, \cdots, c_{(n-1)}P)^T$.

$$\begin{aligned} &= {A_1}^{-1}(A_1(k_1P, r_1P, r_2P, \cdots, r_{(n-1)}P)^T) \\ &= I(k_1P, r_1P, r_2P, \cdots, r_{(n-1)}P)^T, \text{ from Note 1 of 5.3.2} \\ &= (k_1P, r_1P, r_2P, \cdots, r_{(n-1)}P)^T \end{aligned}$$

where $I = {A_1}^{-1}(A_1)$ is identity matrix of order n. Therefore each user $U_i, i = 2, 3, \cdots, n$ can reconstruct the key.

Next, we propose an Authenticated Distributed Group Key Agreement Protocol using Elliptic Curve Secret Sharing Scheme using (ECSSS), which uses public channel for share distribution and gives authentication to the group key.

5.4 Proposed Authenticated Distributed Group Key Agreement Protocol Using Elliptic Curve Secret Sharing Scheme (ADGKAP)

We propose ADGKAP by using ECSSS. ADGKAP has three steps namely, Secret Distribution, Key Reconstruction and Authentication. In Secret Distribution , each user $U_i, i = 1, 2, \cdots, n$ computes $n-1$ private shares and $n-1$ public shares of point k_iP. The private shares are sent in an encrypted manner via a public channel. Then, each user reconstructs his respective share using his private key and encrypted share. In Key Reconstruction, every user $U_i, i = 1, 2, \cdots, n$ reconstructs the key k_jP of $U_j, j = 1, 2, \cdots, n, j \neq i$ by using his respective share and $n-1$ public shares . Then, each user U_i reconstructs the group key $S = \sum_{j=1}^{n} k_jP$. In Authentication, the group key is reconstructed by each individual user. Also, the user authenticates of group key by comparing the hash of each user's reconstructed key with the hash of the other user's reconstructed key. This ADGKAP and the Correctness of the group key reconstruction are explained in sections 5.4 and 5.4.4, and the numerical example of the ADGKAP is explained in section 5.4.5, respectively.

The goal of the scheme is to create a group key S among n users, $U_1, U_2, \cdots, U_n$, collaboratively using public channel. This scheme consists of three steps: Secret Distribution, Key Reconstruction, and Authentication.

5.4.1 Secret Distribution

- Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{F}_q$, q is prime power.
- Point $P \in E(\mathbb{F}_q)$ and $G = \langle P \rangle$ be a group of order ℓ, where ℓ is prime.
- Any user U_i, $i = 1, 2, \cdots, n$, can make E, q, P and ℓ as public.
- Each user U_i chooses a $(2n-2) \times n$ matrix

$$A_i = \begin{bmatrix} 1 & z_{i1} & z_{i1}^2 & \cdots & z_{i1}^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_{i(n-1)} & z_{i(n-1)}^2 & \cdots & z_{i(n-1)}^{n-1} \\ 1 & z_{in} & z_{in}^2 & \cdots & z_{in}^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_{i(2n-2)} & z_{i(2n-2)}^2 & \cdots & z_{i(2n-2)}^{n-1} \end{bmatrix}$$

where $z_{im} \neq z_{ik}$ for $m \neq k$, $z_{im}, z_{ik} \in [1, \ell - 1]$,
for $1 \leq i \leq n$ and $1 \leq m, k \leq (2n-2)$,
and U_i makes A_i public.

- Each U_i chooses $r_{ij}, k_i \in [1, \ell - 1]$ randomly, $j = 1, 2, \cdots, n-1$.

- U_i computes $d_{i1}, d_{i2}, \cdots, d_{i(i-1)}, d_{i(i+1)}, \cdots, d_{in}$, and
$c_{i1}, c_{i2}, \cdots, c_{i(i-1)}, c_{i(i+1)}, \cdots, c_{in}$ as $(d_{i1}, d_{i2}, \cdots, d_{i(i-1)}, d_{i(i+1)}, \cdots, d_{in}, c_{i1}, c_{i2}, \cdots, c_{i(i-1)}, c_{i(i+1)}, \cdots, c_{in})^T = A_i(k_i, r_{i1}, \cdots, r_{i(n-1)})^T$.

- U_i computes key $k_i P$.

- U_i computes $d_{i1}P, d_{i2}P, \cdots, d_{i(i-1)}P, d_{i(i+1)}P, \cdots, d_{in}P$, and
$c_{i1}P, c_{i2}P, \cdots, c_{i(i-1)}P, c_{i(i+1)}P, \cdots, c_{in}P$.

- U_i makes $c_{i1}P, c_{i2}P, \cdots, c_{i(i-1)}P, c_{i(i+1)}P, \cdots, c_{in}P$ as public.

- Each user U_i chooses a private key $v_i \in [1, \ell - 1]$ and makes $v_i P$ public.

- User U_i computes encrypted share $b_{ij}P$ as
$b_{ij}P = d_{ij}P + v_i v_j P$
and sends $b_{ij}P$ publicly to the user U_j, $j = 1, 2, \cdots n$, $j \neq i$.

- User U_j will get the private share $d_{ij}P$ of U_i by computing $d_{ij}P = b_{ij}P - v_j v_i P$.

Note 2: $b_{ij}P$ are encrypted public shares and $c_{ij}P$ are public shares.

5.4.2 Key Reconstruction

- Each user U_i reconstructs the key of U_j, $j = 1, 2, \cdots, n$ and $j \neq i$, as ${M_j}^{-1}(d_{ji}P, c_{j1}P,\ c_{j2}P, \cdots, c_{jn}P)$, which is equivalent to, $(k_jP, r_{j1}P, r_{j2}P, \cdots, r_{j(n-1)}P)$ where

$$M_j = \begin{bmatrix} 1 & z_{ji} & {z_{ji}}^2 & \cdots & {z_{ji}}^{n-1} \\ 1 & z_{jn} & z_{jn}^2 & \cdots & z_{jn}^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & z_{j(2n-2)} & z_{j(2n-2)}^2 & \cdots & z_{j(2n-2)}^{n-1} \end{bmatrix}_{n \times n}$$

  is a submatrix of A_j corresponding to the shares $d_{ji}P, c_{j1}P,\ c_{j2}P, \cdots, c_{jn}P$. Note that M_j is a Vandermonde matrix.

- Finally, each user U_i reconstructs the group key $S = \sum_{j=1}^{n} k_jP$.

5.4.3 Authentication

- Each user U_i computes $C_i = h(S)$ and makes it public, where $S = \sum_{j=1}^{n} k_jP$.

- The user will check the authenticity of the key by checking iff $C_1 = C_2 = \cdots = C_n$. If the above holds true, then the group key S is valid.

Note 3:
$M_j(k_j, r_{j1}, \cdots, r_{j(n-1)})T$ represents corresponding rows in $A_i(k_i, r_{i1}, \cdots, r_{i(n-1)})^T$.

Figure 5.2 describes ADGKAP as follows: Assume three users are in the scheme. In Secret Distribution, user U_1 chooses a key k_1P and computes encrypted shares, $b_{12}P$ for U_2, $b_{13}P$ for U_3 and also computes public shares $c_{12}P, c_{13}P$ and sends them to users, U_2, and U_3 respectively. Similarly, users, U_2 and U_3 chooses keys k_2P and k_3P respectively. User U_2 computes encrypted shares $b_{21}P$ for U_1 and $b_{23}P$ for U_3 and U_3 computes encrypted shares, $b_{31}P$ for U_1 and $b_{32}P$ for U_2. User U_2 computes public shares $c_{21}P, c_{23}P$ sends them to both users U_1 and U_3. User U_3 computes public shares $c_{31}P, c_{32}P$ sends them to both users U_1 and U_2. In Key Reconstruction: User U_1 computes k_2P using $b_{21}P$ and public shares $c_{21}P, c_{23}P$. And also computes k_3P using $b_{31}P$ and public shares $c_{31}P, c_{32}P$. Similarly U_2 computes k_1P and k_3P, and U_3

computes k_1P and k_2P. Finally, users , U_1, U_2, and ,U_3 can compute the group key as $S = k_1P + k_2P + k_3P$.

5.4.4 Correctness of the Group Key Reconstruction

Each user U_i has $(n-1)$ private shares $d_{i1}P, d_{i2}P, \cdots, d_{i(i-1)}P, d_{i(i+1)}P, \cdots, d_{in}P$, and $(n-1)$ public shares $c_{j1}P, c_{j2}P, \cdots, c_{j(i-1)}P, c_{j(i+1)}P, \cdots, c_{jn}P$, for $j = 1, 2, \cdots, n$, $j \neq i$.

User U_i first computes ${M_j}^{-1}$, $j = 1, 2, \cdots, n$ and $j \neq i$. Then the user computes

$${M_j}^{-1}(d_{ji}P, c_{j1}P, \cdots, c_{j(i-1)}P, c_{j(i+1)}P, \cdots, c_{jn}P)^T.$$

$$\begin{aligned} &= {M_j}^{-1}(M_j(k_jP, r_{j1}P, \cdots, r_{j(n-1)}P)^T), \text{ from Note 3 of 5.4.3} \\ &= I(k_jP, r_{j1}P, \cdots, r_{j(n-1)}P)^T \\ &= (k_jP, r_{j1}P, \cdots, r_{j(n-1)}P)^T \end{aligned}$$

where $I = {M_j}^{-1}(M_j)$ is the identity matrix of order n.

Finally, user U_i computes group key $S = \sum_{j=1}^{n} k_jP$.

5.4.5 Numerical Example

- Let $E : y^2 = x^3 + 11x + 12$ be an elliptic curve over $\mathbb{F}_{467}$.
- Point $P = (360, 185) \in E(\mathbb{F}_{467})$ and $G = \langle P \rangle$ be a group of order $\ell = 79$.
- Any user U_i, $i = 1, 2, 3$, can make E, q, P and ℓ public.
- We discuss in detail the Secret Distribution , Key Reconstruction, and Authentication in Tables 5.1, 5.2, and 5.3, respectively.

Table 5.1 is described as follows: In step 1, each user U_i, $i = 1, 2, 3$ chooses a matrix of order 4×3 and makes it public. In step 2, each user U_i, $i = 1, 2, 3$, chooses three random integers $k_i, r_{i1}, r_{i2} \in [1, 78]$ randomly. In step 3, user U_1 computes d_{12}, d_{13}, c_{12} and c_{13}, user U_2 computes d_{21}, d_{23}, c_{21} and c_{23} , user U_3 computes

d_{31}, d_{32}, c_{31} and c_{32} . In step 4, user U_i computes key k_iP for $i = 1, 2, 3$. In step 5, user U_1 computes $d_{12}P, d_{13}P, c_{12}P$, $c_{13}P$ and makes $c_{12}P$ and $c_{13}P$ public, user U_2 computes $d_{21}P, d_{23}P, c_{21}P$, $c_{23}P$ and makes $c_{21}P$ and $c_{23}P$ public, user U_3 computes $d_{31}P, d_{32}P, c_{31}P$, $c_{32}P$ and makes $c_{31}P$ and $c_{32}P$ public . In step 6, user $U_i, i = 1, 2, 3$ chooses private key v_i and computes public key v_iP . In step 7, user U_1 computes $b_{12}P$ and $b_{13}P$, user U_2 computes $b_{21}P$ and $b_{23}P$, user U_3 computes $b_{31}P$ and $b_{32}P$. Then the users make $b_{12}P, b_{13}Pb_{21}P, b_{23}P, b_{31}P, b_{32}P$ public. In step 8, user U_1 computes his shares $d_{21}P$ and $d_{31}P$ from v_2P and v_3P respectively, user U_2 computes his from shares $d_{12}P$ and $d_{32}P$ from v_1P and v_3P respectively, user U_3 computes his shares $d_{13}P$ and $d_{23}P$ from v_1P and v_2P respectively.

Table 5.2 is described as follows: In step 1, user U_1 computes $k_2P, r_{21}P, r_{22}P$ from M_2 and $d_{21}P, c_{21}P, c_{23}P$, user U_2 computes $k_1P, r_{11}P, r_{12}P$ from M_1 and $d_{12}P, c_{12}P, c_{13}P$, user U_3 computes $k_1P, r_{11}P, r_{12}P$ from M_1 and $d_{13}P, c_{12}P, c_{13}P$. In step 2, user U_1 computes $k_3P, r_{31}P, r_{32}P$ from M_3 and $d_{31}P, c_{31}P, c_{32}P$, user U_2 computes $k_3P, r_{31}P, r_{32}P$ from M_3 and $d_{32}P, c_{31}P, c_{32}P$. user U_3 computes $k_2P, r_{21}P, r_{22}P$ from M_2 and $d_{23}P, c_{21}P$, $c_{22}P$. In step 3 ,user $U_i, i = 1, 2, 3$ computes group key $S = k_1P + k_2P + k_3P$.

Table 5.3 is described as follows: In step 1, user U_i computes $C_i, i = 1, 2, 3$ and makes it public. In step 2, user U_1 verify $C_1 = C_2 = C_3$, then if they are equal, group key is valid . Similarly, U_2 and U_3 verify if their group key is valid or not.

5.5 Security Analysis

In this section, we discussed in detail that a valid user could reconstruct the group key, but an attacker can not reconstruct the group key in detail.

5.5.1 Every User can reconstruct group key by using public shares

Suppose the user U_i wants to compute k_jP, $j = 1, 2, \cdots, n$, $j \neq i$, the secret point of U_j with the help of the private share $d_{ji}P$ and $n - 1$ public shares. The user U_i can form a system of equations as follows

Table 5.1: Secret distribution

Steps	User U_1	User U_2	User U_3
1	U_1 chooses 4×3 matrix $A_1 = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 5 & 25 \\ 1 & 3 & 9 \\ 1 & 6 & 36 \end{bmatrix}$ A_1 is public	U_2 chooses 4×3 matrix $A_2 = \begin{bmatrix} 1 & 3 & 9 \\ 1 & 7 & 49 \\ 1 & 4 & 16 \\ 1 & 2 & 4 \end{bmatrix}$ A_2 is public	U_3 chooses 4×3 matrix $A_3 = \begin{bmatrix} 1 & 8 & 64 \\ 1 & 4 & 16 \\ 1 & 9 & 81 \\ 1 & 10 & 100 \end{bmatrix}$ A_3 is public
2	U_1 chooses randomly $k_1 = 14, r_{11} = 32, r_{12} = 27$	U_2 chooses randomly $k_2 = 19, r_{21} = 21, r_{22} = 35$	U_3 chooses randomly $k_3 = 48, r_{31} = 61, r_{32} = 12$
3	U_1 computes $\begin{bmatrix} d_{12} \\ d_{13} \\ c_{12} \\ c_{13} \end{bmatrix} = A_1 \begin{bmatrix} 14 \\ 32 \\ 27 \end{bmatrix} = \begin{bmatrix} 186 \\ 849 \\ 353 \\ 1178 \end{bmatrix}$	U_2 computes $\begin{bmatrix} d_{21} \\ d_{23} \\ c_{21} \\ c_{23} \end{bmatrix} = A_2 \begin{bmatrix} 19 \\ 21 \\ 35 \end{bmatrix} = \begin{bmatrix} 397 \\ 1881 \\ 663 \\ 201 \end{bmatrix}$	U_3 computes $\begin{bmatrix} d_{31} \\ d_{32} \\ c_{31} \\ c_{32} \end{bmatrix} = A_3 \begin{bmatrix} 48 \\ 61 \\ 12 \end{bmatrix} = \begin{bmatrix} 1304 \\ 484 \\ 1569 \\ 1858 \end{bmatrix}$
4	U_1 computes Key $k_1P = (99, 290)$	U_2 computes Key $k_2P = (116, 97)$	U_3 computes Key $k_3P = (410, 375)$
5	$\begin{bmatrix} d_{12}P \\ d_{13}P \\ c_{12}P \\ c_{13}P \end{bmatrix} = \begin{bmatrix} (424, 193) \\ (21, 346) \\ (108, 135) \\ (288, 416) \end{bmatrix}$ $c_{12}P, c_{13}P$ make public	$\begin{bmatrix} d_{21}P \\ d_{23}P \\ c_{21}P \\ c_{23}P \end{bmatrix} = \begin{bmatrix} (435, 7) \\ (11, 159) \\ (410, 92) \\ (275, 427) \end{bmatrix}$ $c_{21}P, c_{23}P$ make public	$\begin{bmatrix} d_{31}P \\ d_{32}P \\ c_{31}P \\ c_{32}P \end{bmatrix} = \begin{bmatrix} (88, 50) \\ (316, 252) \\ (221, 195) \\ (387, 157) \end{bmatrix}$ $c_{31}P, c_{32}P$ make public
6	U_1 choose Private key $v_1 = 24$ Computes public key $v_1P = (261, 186)$	U_2 chooses Private key $v_2 = 57$ Computes public key $v_2P = (133, 124)$	U_3 chooses Private key $v_3 = 63$ Computes public key $v_3P = (121, 391)$
7	U_1 computes encrypted shares $b_{12}P = d_{12}P + v_1v_2P$ $\implies b_{12}P = (394, 229)$ $b_{13}P = d_{13}P + v_1v_3P$ $\implies b_{13}P = (24, 435)$ $b_{12}P, b_{13}P$ made public	U_2 computes encrypted shares $b_{21}P = d_{21}P + v_2v_1P$ $\implies b_{21}P = (220, 253)$ $b_{23}P = d_{23}P + v_2v_3P$ $\implies b_{23}P = (99, 177)$ $b_{21}P, b_{23}P$ made public	U_3 computes encrypted shares $b_{31}P = d_{31}P + v_3v_1P$ $\implies b_{31}P = (424, 274)$ $b_{32}P = d_{32}P + v_3v_2P$ $\implies b_{32}P = (250, 169)$ $b_{31}P, b_{32}P$ made public
8	U_1 computes shares $d_{21}P = b_{21}P - v_1v_2P$ $\implies d_{21}P = (435, 7)$ $d_{31}P = b_{31}P - v_1v_3P$ $\implies d_{31}P = (88, 50)$	U_2 computes shares $d_{12}P = b_{12}P - v_2v_1P$ $\implies d_{12}P = (424, 193)$ $d_{32}P = b_{32}P - v_2v_3P$ $\implies d_{32}P = (316, 252)$	U_3 computes shares $d_{13}P = b_{13}P - v_3v_1P$ $\implies d_{13}P = (21, 346)$ $d_{23}P = b_{23}P - v_3v_2P$ $\implies d_{23}P = (11, 159)$

Table 5.2: Key Reconstruction

Steps	U_1	U_2	U_3
1	$M_2 = \begin{bmatrix} 1 & 3 & 9 \\ 1 & 4 & 16 \\ 1 & 2 & 4 \end{bmatrix}$ ${M_2}^{-1}(d_{21}P, c_{21}P, c_{23}P)^T$ $= \begin{bmatrix} (116, 97) \\ (203, 437) \\ (285, 426) \end{bmatrix} = \begin{bmatrix} k_2P \\ r_{21}P \\ r_{22}P \end{bmatrix}$	$M_1 = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 6 & 36 \end{bmatrix}$ ${M_1}^{-1}(d_{12}P, c_{12}P, c_{13}P)^T$ $= \begin{bmatrix} (99, 290) \\ (140, 153) \\ (220, 253) \end{bmatrix} = \begin{bmatrix} k_1P \\ r_{11}P \\ r_{12}P \end{bmatrix}$	$M_1 = \begin{bmatrix} 1 & 5 & 25 \\ 1 & 3 & 9 \\ 1 & 6 & 36 \end{bmatrix}$ ${M_1}^{-1}(d_{13}P, c_{12}P, c_{13}P)^T$ $= \begin{bmatrix} (99, 290) \\ (140, 153) \\ (220, 253) \end{bmatrix} = \begin{bmatrix} k_1P \\ r_{11}P \\ r_{12}P \end{bmatrix}$
2	$M_3 = \begin{bmatrix} 1 & 8 & 64 \\ 1 & 9 & 81 \\ 1 & 10 & 100 \end{bmatrix}$ ${M_3}^{-1}(d_{31}P, c_{31}P, c_{32}P)^T$ $= \begin{bmatrix} (410, 375) \\ (121, 391) \\ (163, 367) \end{bmatrix} = \begin{bmatrix} k_3P \\ r_{31}P \\ r_{32}P \end{bmatrix}$	$M_3 = \begin{bmatrix} 1 & 4 & 16 \\ 1 & 9 & 81 \\ 1 & 10 & 100 \end{bmatrix}$ ${M_3}^{-1}(d_{32}P, c_{31}P, c_{32}P)^T$ $= \begin{bmatrix} (410, 375) \\ (121, 391) \\ (163, 367) \end{bmatrix} = \begin{bmatrix} k_3P \\ r_{31}P \\ r_{32}P \end{bmatrix}$	$M_2 = \begin{bmatrix} 1 & 7 & 49 \\ 1 & 4 & 16 \\ 1 & 2 & 4 \end{bmatrix}$ ${M_2}^{-1}(d_{23}P, c_{21}P, c_{23}P)^T$ $= \begin{bmatrix} (116, 97) \\ (203, 437) \\ (285, 426) \end{bmatrix} = \begin{bmatrix} k_2P \\ r_{21}P \\ r_{22}P \end{bmatrix}$
3	Group key $S = \sum_{j=1}^{3} k_jP = (435, 7)$	Group key $S = \sum_{j=1}^{3} k_jP = (435, 7)$	Group key $S = \sum_{j=1}^{3} k_jP = (435, 7)$

Table 5.3: Key Authentication

Steps	U_1	U_2	U_3
1	$h(S) = 34$ $C_1 = 34$ is public	$h(S) = 34$ $C_2 = 34$ is public	$h(S) = 34$ $C_3 = 34$ is public
2	$C_1 = C_2 = C_3$ Group key S is Valid	$C_1 = C_2 = C_3$ Group key S is Valid	$C_1 = C_2 = C_3$ Group key S is Valid

$$
\begin{aligned}
k_jP + r_{j1}z_{ji}P + \cdots + r_{j(n-1)}z_{ji}^{n-1}P &= d_{ji}P \\
k_jP + r_{j1}z_{jn}P + \cdots + r_{j(n-1)}z_{jn}^{n-1}P &= c_{j1}P \\
k_jP + r_{j1}z_{j(n+1)}P + \cdots + r_{j(n-1)}z_{j(n+1)}^{n-1}P &= c_{j2}P \\
&\vdots \\
k_jP + r_{j1}z_{j(2n-2)}P + \cdots + r_{j(n-1)}z_{j(2n-2)}^{n-1}P &= c_{jn}P
\end{aligned}
$$

Here, $c_{i1}, c_{i2} \cdots, c_{in}$ are public values, and k_ip and $r_{i1}p, \cdots, r_{in-1}p$ and can be calculated by $z_{in}, \cdots, z_{i2n-2}$.

The same matrix form can be represented as

$$
\begin{bmatrix}
1 & z_{ji} & z_{ji}^2 & \cdots & z_{ji}^{n-1} \\
1 & z_{jn} & z_{jn}^2 & \cdots & z_{jn}^{n-1} \\
1 & z_{j(n+1)} & z_{j(n+1)}^2 & \cdots & z_{j(n+1)}^{n-1} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
1 & z_{j(2n-2)} & z_{j(2n-2)}^2 & \cdots & z_{j(2n-2)}^{n-1}
\end{bmatrix}
\begin{bmatrix}
k_jP \\ r_{j1}P \\ r_{j2}P \\ \vdots \\ r_{jn}P
\end{bmatrix}
=
\begin{bmatrix}
d_{ji}P \\ c_{j1}P \\ c_{j2}P \\ \vdots \\ c_{jn}P
\end{bmatrix}
$$

The above coefficient matrix of the system has n unknowns and n equations, as the rank of the coefficient matrix is n. Thus the system of equations has a unique solution [3]. Hence a user can get key k_jP, $j = 1, 2 \cdots n$, by inverting the coefficient matrix and multiplying it with the share matrix $(d_{ji}P, c_{j1}P, c_{j2}P, \cdots, c_{jn}P)^T$. Finally, the user U_i computes the group key $S = \sum_{j=1}^{n} k_jP$, using his secret key k_iP.

5.5.2 An attacker cannot reconstruct the group key by using public shares

Suppose an attacker wants to get k_iP with $n-1$ public shares. The attacker can form a system of equations as follows.

$$
\begin{aligned}
k_P + r_{i1} z_{in} P + \cdots + r_{i(n-1)} {z_{in}}^{n-1} P &= c_{i1} P \\
k_i P + r_{i1} z_{i(n+1)} P + \cdots + r_{i(n-1)} z_{i(n+1)}^{n-1} P &= c_{i2} P \\
&\vdots \\
k_i P + r_{i1} z_{i(n+i-2)} P + \cdots + r_{i(n-1)} z_{i(n+i-2)}^{n-1} P &= c_{i(i-1)} P \\
k_i P + r_{i1} z_{i(n+i)} P + \cdots + r_{i(n-1)} z_{i(n+i)}^{n-1} P &= c_{i(i+1)} P \\
&\vdots \\
k_i P + r_{i1} z_{i(2n-2)} P + \cdots + r_{i(n-1)} z_{i(2n-2)}^{n-1} P &= c_{in} P
\end{aligned}
$$

The same matrix form can be represented as

$$
\begin{bmatrix}
1 & z_{in} & \cdots & z_{in}^{n-1} \\
1 & z_{i(n+1)} & \cdots & z_{i(n+1)}^{n-1} \\
\vdots & \vdots & \vdots & \vdots \\
1 & z_{i(n+i-2)} & \cdots & z_{i(n+i-2)}^{n-1} \\
1 & z_{i(n+i)} & \cdots & z_{i(n+i)}^{n-1} \\
\vdots & \vdots & \vdots & \vdots \\
1 & z_{i(2n-2)} & \cdots & z_{i(2n-2)}^{n-1}
\end{bmatrix}
\begin{bmatrix}
k_i P \\ r_{i1} P \\ \vdots \\ r_{i(n-1)} P
\end{bmatrix}
=
\begin{bmatrix}
c_{i1} P \\ c_{i2} P \\ \vdots \\ c_{i(i-1)} P \\ c_{i(i+1)} P \\ \vdots \\ c_{in} P
\end{bmatrix}
$$

The above coefficient matrix of the system has n unknowns and only $n-1$ equations. Thus the system of equations has finitely many solutions[3], and each solution is of equal probability of $\frac{1}{\ell^n}$. For large ℓ, $\frac{1}{\ell^n}$ is negligible. Thus an attacker cannot get key $k_i P$ and hence $S = \sum_{i=1}^{n} k_i P$.

5.6 Comparisons

This section compares our proposed scheme, Authenticated Distributed Group Key Agreement Protocol using Elliptic Curve Secret Sharing Scheme (ADGKAP) with Harn et al.[39] on various parameters such as Key size, a technique used, no of rounds and so on. In literature, Shanks' and Index Calculus Algorithms [78] are popular for solving discrete logarithm algorithms in sub exponentiation time. However, till date, there are no such algorithms for ECDLP [38].

Our ADGKAP scheme's security relies on ECDLP(Elliptic Curve Discrete Loga-

Table 5.4: Comparison of computation parameters between Harn Scheme and ADGKAP scheme

Parameters	Harn et al.[39]	Our ADGKAP
Key Size(in bits)	1024,2048,3072	160,224,256
Hard problem	DLP	ECDLP
Crypto technique	Polynomial SSS	Elliptic Curve SSS
Attack	DLP attacks(Index calculus,Shanks)	No ECDLP attack

rithm Problem), but Harn et al. rely only on DLP (Discrete Logarithm Problem). It is widely known that ECDLP is more secure compared to DLP, hence our scheme is more secure comparatively. Harn et al. worked on the finite field $\mathbb{F}_q$, but our ADGKAP scheme worked on the elliptic curve over a finite field $E(\mathbb{F}_q)$. This ensures that our ADGKAP scheme, key size, and storage space are significantly less and have fast computation compared to Harn et al. [39]. A comparison of Harn et al. with our proposed ADGKAP scheme with respect to various computational efficiency parameters is listed in Table 5.4.

In Cheng et al. [19] and Zhang Qikum et al. [94] schemes, the group key is distributed to participants using an encryption mechanism. Decryption relies on a single key making it less secure because of a single point of failure. However, our proposed ADGKAP scheme is more secure because shares of the key are distributed to a group of participants instead of a key. Also, all of these use a secure channel for communication among users. But our ADGKAP uses public channel for communication among participants. Li J et al. [59] uses $8n + 6$ exponentiation, where n is operations, Zhang Q et al. [94] uses $(2n + 10)$ exponentiation, Cao et al. [16] uses 2 bilinear parings and Zhang L et. al [93] uses 16 exponentiation, where n is number of users. All these four schemes use a secure channel for communication between KGC and users. Cheng et al. [19] uses $(2n + 2)$ bilinear operations, and Zhang Qikum et al. [94] uses 5 bilinear operations. Both Cheng et al. [19], and Zhang Qikum et al. [94] protocols were designed without KGC. Harn et al. used $2n$ exponentiation operations and a secret sharing scheme without KGC. However, our scheme uses n^2 scalar multiplications only. The cost of bilinear parings and exponentiation are costly operations compared to scalar multiplications[94]. Hence ADGKAP has less computational cost compared to existing schemes. Cost comparisons among various schemes are listed in table 6.1.

In our proposed scheme, the matrix A_i and inverse of the matrix M_j of user $U_j, j = 1, 2, \cdots, n$ are pre computed and made public by user U_i. Harn et al. require, $(n^2 + n - 2)$ additions, multiplications $(6n^2 - 2n - 4)$, $(2n)$ divisions , $(2n)$ expo-

Table 5.5: Cost comparisons among various schemes

Author	KGC	SSS	Channel	Total cost
Li J et al. [59]	Yes	No	Secure	$(8n+6)$ exponentiation
Zhang L et. al [93]	Yes	No	Secure	16 exponentiation
Zhang Q et al. [94]	Yes	No	Secure	$(2n+10)$ exponentiation
Cheng et al. [19]	No	No	Secure	$(2n+2)$ bilinear pairings
Qikum et al. [95]	No	No	Secure	5 bilinear pairings
Cao et al. [16]	Yes	Yes	Secure	2 bilinear pairings
Harn et al. [39]	No	Yes	Public	$2n$ exponentiation
ADGKAP scheme	No	Yes	Public	n^2 scalar multiplications

Table 5.6: Cost comparison between Harn Scheme and ADGKAP scheme

Operations	Harn Scheme[39]	ADGKAP scheme
Additions	$O(n^2)$	Nil
Multiplications	$O(n^2)$	$O(n^2)$
Divisions	$O(n)$	Nil
Exponentiation	$O(n)$	Nil
Point additions	Nil	$O(n^2)$
Scalar multiplications	Nil	$O(n^2)$
Hashing	$O(n)$	$O(n)$

nentiation and $(3n)$ hashing operations . However, our proposed scheme (ADGKAP) requires $(2n^2-2n)$ multiplications, $(3n^2-3n)$ point additions , $(3n^2+n-1)$ scalar multiplications and (n) hashing operations. A comparison of the computational cost between the Harn scheme and our proposed ADGKAP scheme is given in table 5.6. The observation from the table is that the costlier operation exponentiation is avoided in our scheme and replaced by point additions and scalar multiplications that deals with smaller key size with an equal level of security. This reduces the computational cost of our scheme very much compared to Harn's scheme. This makes our scheme efficient in terms of computational cost and key size, which makes it a better choice for resource constrained environments. Hence our proposed ADGKAP scheme gives similar security with a smaller key size.

5.7 Summary

We proposed a novel Elliptic curve secret sharing scheme (ECSSS) for share distribution that is secure enough with a relatively smaller key size and storage. Then, an Authenticated Distributed Group Key Agreement Protocol using Elliptic Curve Secret

Sharing Scheme (ECSSS) is proposed. The proposed scheme can be used efficiently in a distributed environment for a group key agreement. In comparison to existing schemes, our proposed ADGKAP offers equal security with much smaller key sizes, less storage space, and less computational cost without compromising on the number of rounds. Every valid user in this scheme can reconstruct the group key, but the attacker cannot do so. ECDLP is a major aspect of this scheme's security. Compared to Authenticated group Diffie–Hellman key agreement protocol [39] our proposed scheme ADGKAP is more appropriate for resource constrained devices in a distributed environment.

In the future, it would be worthwhile to explore the dynamics of participant interactions, especially the aspects of joining and leaving, within the framework of the Authenticated Distributed Group Key Agreement Protocol implemented using ECSSS.

Chapter 6

Multi Group Key Agreement Protocol using SSS

6.1 Introduction

While numerous key agreement protocols are designed for single groups, their efficiency wanes when applied to multi-group situations. In this digital age, where remote work and online collaboration have become the norm, it is vital to ensure secure group communication across various network applications. Leveraging group key agreement protocols allows multiple participants to create a shared key, thereby ensuring the security of their communication over networks. However, these protocols often fall short in scenarios involving multiple groups that need to establish a common key for secure interaction.

Now, consider the scenario where we try to adapt single group key agreement protocols for multi-group scenarios. Here, the protocols typically encounter efficiency issues, as they are not originally designed for such complex situations. Specifically, applying these single group key agreement protocols for multi-group key agreement incurs significant overhead. This is because multiple iterations and re-computations are required to establish a common key among various groups, which increases computational resources and time, hence undermining the performance of the network. Therefore, it's crucial to develop specialized key agreement protocols tailored for multi-group scenarios to maintain efficient and secure communication.

Problem Identification and Motivation

In today's rapidly evolving digital landscape, the rise of remote work and the spatial distribution of teams has led to an increased demand for efficient and secure multi-group communication methods. Traditionally, many key decision-making bodies function as a collective rather than as individuals, forming a series of interlinked committees. To facilitate effective, secure communication among these committees, it becomes essential to implement key agreement protocols explicitly tailored for multi-group scenarios.

However, the problem arises when we notice that most existing key agreement protocols such as Diffie-Hellman and its various extensions, are primarily built on the principle of pairwise key agreements or single group key agreements. These protocols, while effective for their designed purposes, fail to address the unique set of challenges posed by multi-group communications. For instance, they do not account for the increased computational overhead that comes with scaling from single to multiple groups.

Moreover, when single group key agreement protocols are stretched to accommodate multi-group situations, they suffer from a significant increase in computational overhead, as they require additional iterations and calculations to establish a common key amongst different groups. This extra overhead negatively impacts the network's performance by consuming more resources and increasing latency.

The motivation driving this study arises from the urgent need to establish secure, efficient communication across geographically dispersed groups in an era defined by online collaboration and remote work. It becomes paramount to explore and develop new key agreement protocols specifically engineered for multi-group scenarios. These protocols should not only ensure secure communication by generating a common key across multiple groups, but also optimize the process to minimize computational overhead and resource consumption. By achieving this, we can improve the overall performance of networks and foster seamless, secure communication across multiple groups.

Contribution

To address the limitations of existing protocols in multi-group scenarios, we propose a novel multi-group key agreement protocol based on a secret sharing scheme. In our opinion, our proposed protocol represents the first of its kind. This paper contributes the following main contributions:

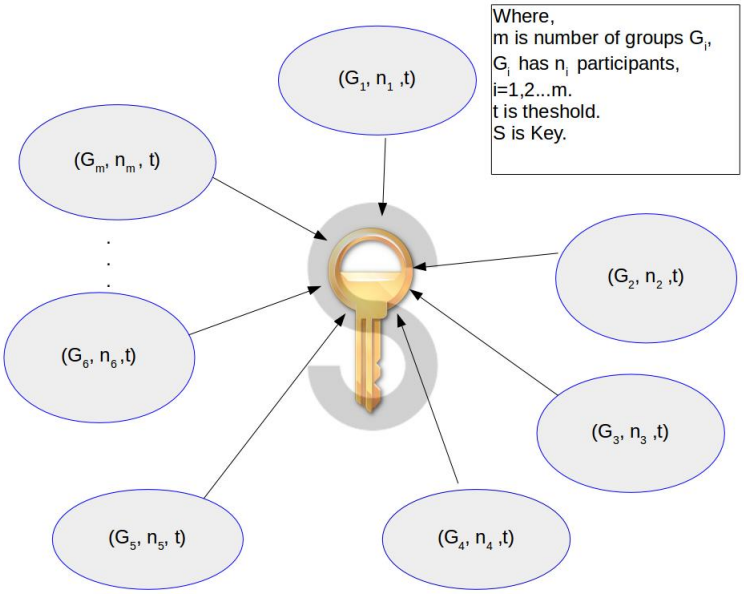

Figure 6.1: Multi Group Key Agreement Protocol Using Secret Sharing Scheme

- Using a secret sharing scheme, we propose a multi-group key agreement protocol that fills a gap in the existing literature on multi-group key agreements.
- In every group, the key can be rebuilt by any t or more participants, where t represents the threshold number of participants.
- Any group will not be able to reconstruct the key with $t-1$ or fewer participants.
- The safety of a scheme relies on the discrete logarithm problem.

6.2 Proposed Multi Group Key Agreement Protocol Using Secret Sharing Scheme (MGKAP)

Participants from multiple groups can reconstruct the same key for having multi group communication in a collaborative application. The entire setup is divided into m groups, each with n_i participants and threshold t for each group. Each participant can reconstruct the key S iff at least a threshold number of individuals from his group come together.

Figure 6.1 represents our multi group key agreement protocol containing m groups, each with n_i participants. Each group has threshold t, i.e., the secret S, which is the group key, can be reconstructed if and only if atleast threshold t number of participants from each group come together.

The idea of the scheme is to work in a distributed environment, where any participant from any group who wants to initiate the communication can select the threshold t and the prime numbers q and p. All these three values are made public to all the participants. Each participant chooses a random key polynomial and computes its subshares, and distributes them secretly to other participants and himself in every group. Every participant must verify whether the respective subshare received from other participants is valid or not. After that, each participant calculates his respective share from the received subshares and his subshare. A threshold number of participants from each group need to come together to reconstruct the key S. After reconstructing the key S at each group, any participant of the multiple groups can verify whether the reconstructed key is correct or not. If the key is valid, participants can start their secure group communication; otherwise, they cannot, which is discussed in theorem 6.4.3.

Proposed MGKAP involves six steps: Setup, Subshare Distribution, Commitments, Subshare Verification, Original Share Construction, and Key Reconstruction.

6.2.1 Setup

- Assume that there are n total participants in m groups, each group with n_i participants such that $\sum_{i=1}^{m} n_i = n$.
- The threshold value for each group is t.
- Each group has n_i participants, where $n_i \geq t$.
- Each participant p_i has id_i, $i = 1, 2, \cdots, n$.
- Any participant from any group initiate the communication, by choosing a large prime p such that $q \mid (p-1)$ and primitive root a of subgroup of $\mathbb{Z}_p^*$ of order q.
- All the above values namely p, q, a, t are made public by the respective participant.

6.2.2 Subshare Distribution

- Each participant p_i, chooses a key s_i and a polynomial of degree $t-1$,

$$g_i(x) = s_i + c_{i1}x + c_{i2}x^2 + c_{i3}x^3 + \cdots + c_{i(t-1)}x^{t-1}, i = 1, \cdots, n, \tag{6.1}$$

  where $c_{i1}, c_{i2}, \cdots, c_{i(t-1)} \in \mathbb{F}_q$.

- And computes subshare $y_{ij} = g_i(id_j) \bmod q,\ 1 \leq i, j \leq n$.

- Later, subshare y_{ij} is communicated to all other participants and himself secretly.

6.2.3 Commitments

- Each participant p_i computes Feldman commitments [34] $cm_{i0},\ cm_{ik}$ and makes them public where

$$cm_{i0} = a^{s_i} \bmod p,\ 1 \leq i \leq n. \tag{6.2}$$

$$cm_{ik} = a^{c_{ik}} \bmod p,\ 1 \leq i \leq n,\ 1 \leq k \leq t-1. \tag{6.3}$$

6.2.4 Subshare Verification

- Each participant p_i can verify his respective subshare y_{ij} by using commitments from section 6.2.3. His respective subshare is valid iff

$$a^{y_{ij}} = (cm_{i0})(cm_{i1})^{id_j}(cm_{i2})^{{id_j}^2} \cdots (cm_{i(t-1)})^{{id_j}^{t-1}} \bmod p,\ 1 \leq i, j \leq n.$$

6.2.5 Original Share Construction

- Each participant p_i has $n-1$ subshares and 1 subshare of himself , a total of n subshares with him as y_{ij}, where $1 \leq j \leq n$.

- And computes his original share y_i as

$$y_i = \sum_{j=1}^{n} y_{ji} \bmod q. \tag{6.4}$$

6.2.6 Key Reconstruction

- Applying Lagrange's interpolation equation, a threshold number of participants from each group will come together to get the secret polynomial

$$g(x) = \sum_{i=1}^{t} L_i(x) y_i \bmod q. \tag{6.5}$$

$$g(x) = \sum_{i=1}^{n} g_i(x) \bmod q. \tag{6.6}$$

- Then, threshold number of participants compute key $S = g(0) = \sum_{i=1}^{n} g_i(0) \bmod q = \sum_{i=1}^{n} s_i \bmod q$.

- Final key S can now be used as a session key for multi group communication.

6.3 Numerical Example

A numerical example on how each participant computes his Subshares, Commitments, Subshare verification and Key Reconstruction is explained below step wise:

6.3.1 Setup

Assume total participants $n = 24$

Number of groups $m = 4$

Group 1 participant $id's = [1, 2, 3, 4, 5, 6, 7]$

Group 2 participant $id's = [8, 9, 10, 11, 12, 13]$

Group 3 participant $id's = [14, 15, 16, 17, 18]$

Group 4 participant $id's = [19, 20, 21, 22, 23, 24]$

Any participant of any group can initiate a group communication. Lets us say a participant chooses for each group threshold, the number of participants $t = 4$, prime numbers $q = 233$, and $p = 467$. $a = 4$ is the primitive root of subgroup $\mathbb{Z}^*_{467}$ values p, q, t and a are made public.

Tuple $[x, y]$ represents x as id and y as subshare y_{ij}.

Can be be selected by any participant and made public.

6.3.2 Subshare Distribution

Participant 1 chooses a three degree polynomial
$g_1(x) = 39 + 56x + 133x^2 + 211x^3$.
Participant 1 computes his subshares as

[[1, 206], [2, 41], [3, 111], [4, 51], [5, 195], [6, 178], [7, 101], [8, 65], [9, 171], [10, 54], [11, 48], [12, 21], [13, 74], [14, 75], [15, 125], [16, 92], [17, 77], [18, 181], [19, 39], [20, 218], [21, 120], [22, 79], [23, 196], [24, 106]].

Participant 2 chooses three degree polynomial $g_2(x) = 87 + 13x + 222x^2 + 173x^3$.
Participant 2 computed sub shares

[[1, 29], [2, 55], [3, 38], [4, 84], [5, 66], [6, 90], [7, 29], [8, 222], [9, 76], [10, 163], [11, 123], [12, 62], [13, 86], [14, 68], [15, 114], [16, 97], [17, 123], [18, 65], [19, 29], [20, 121], [21, 214], [22, 181], [23, 128], [24, 161]].

Participant 3 chooses three degree polynomial $g_3(x) = 5 + 183x + 52x^2 + 30x^3$.
Participant 3 computed sub shares

[[1, 37], [2, 120], [3, 201], [4, 227], [5, 145], [6, 135], [7, 144], [8, 119], [9, 7], [10, 221], [11, 9], [12, 17], [13, 192], [14, 15], [15, 132], [16, 24], [17, 104], [18, 86], [19, 150], [20, 10], [21, 79], [22, 71], [23, 166], [24, 78]].

Participant 4 chooses three degree polynomial $g_4(x) = 190 + 89x + 207x^2 + 63x^3$.
Participant 4 computed sub shares

[[1, 83], [2, 69], [3, 60], [4, 201], [5, 171], [6, 115], [7, 178], [8, 39], [9, 76], [10, 201], [11, 93], [12, 130], [13, 224], [14, 54], [15, 231], [16, 201], [17, 109], [18, 100], [19, 86], [20, 212], [21, 157], [22, 66], [23, 84], [24, 123]].

Participant 5 chooses three degree polynomial $g_5(x) = 188 + 3x + 138x^2 + 150x^3$.
Participant 5 computed subshares

[[1, 13], [2, 82], [3, 130], [4, 125], [5, 35], [6, 61], [7, 171], [8, 100], [9, 49], [10, 219], [11, 112], [12, 162], [13, 104], [14, 139], [15, 2], [16, 127], [17, 16], [18, 103], [19, 123], [20, 44], [21, 67], [22, 160], [23, 58], [24, 195]].

Participant 6 chooses three degree polynomial $g_6(x) = 132 + 65x + 143x^2 + 115x^3$.
Participant 6 computed subshares

[[1, 222], [2, 123], [3, 59], [4, 21], [5, 0], [6, 220], [7, 206], [8, 182], [9, 139], [10, 68], [11, 193], [12, 39], [13, 63], [14, 23], [15, 143], [16, 181], [17, 128], [18, 208], [19, 179], [20, 32], [21, 224], [22, 47], [23, 191], [24, 181]].

Participant 7 chooses three degree polynomial $g_7(x) = 157 + 202x + 211x^2 + 189x^3$
Participant 7 computed sub shares

[[1, 60], [2, 121], [3, 76], [4, 127], [5, 10], [6, 160], [7, 80], [8, 205], [9, 38], [10, 14], [11, 102], [12, 38], [13, 24], [14, 29], [15, 22], [16, 205], [17, 81], [18, 85], [19, 186], [20, 120], [21, 89], [22, 62], [23, 8], [24, 129]]

Participant 8 chooses three degree polynomial $g_8(x) = 167 + 221x + 69x^2 + 29x^3$
Participant 8 computed subshares

[[1, 20], [2, 185], [3, 137], [4, 50], [5, 98], [6, 222], [7, 130], [8, 229], [9, 227], [10, 65], [11, 150], [12, 190], [13, 126], [14, 132], [15, 149], [16, 118], [17, 213], [18, 142], [19, 79], [20, 198], [21, 207], [22, 47], [23, 125], [24, 149]]

Participant 9 chooses three degree polynomial $g_9(x) = 97 + 74x + 53x^2 + 72x^3$.
Participant 9 computed subshares

[[1, 63], [2, 101], [3, 177], [4, 24], [5, 74], [6, 60], [7, 181], [8, 170], [9, 226], [10, 82], [11, 170], [12, 223], [13, 207], [14, 88], [15, 65], [16, 104], [17, 171], [18, 232], [19, 20], [20, 200], [21, 39], [22, 202], [23, 189], [24, 199]].

Participant 10 chooses three degree polynomial $g_{10}(x) = 211 + 212x + 61x^2 + 105x^3$
Participant 10 computed subshares

[[1, 123], [2, 88], [3, 37], [4, 134], [5, 77], [6, 30], [7, 157], [8, 156], [9, 191], [10, 193], [11, 93], [12, 55], [13, 10], [14, 122], [15, 89], [16, 75], [17, 11], [18, 61], [19, 156], [20, 227], [21, 205], [22, 21], [23, 72], [24, 56]]

Participant 11 chooses three degree polynomial $g_{11}(x) = 148 + 116x + 37x^2 + 44x^3$.

Participant 11 computed subshares

[[1, 112], [2, 181], [3, 153], [4, 59], [5, 163], [6, 30], [7, 157], [8, 109], [9, 150], [10, 78], [11, 157], [12, 185], [13, 193], [14, 212], [15, 40], [16, 174], [17, 179], [18, 86], [19, 159], [20, 196], [21, 228], [22, 53], [23, 168], [24, 138]].

Participant 12 chooses three degree polynomial $g_{12}(x) = 55 + 77x + 35x^2 + 1x^3$.
Participant 12 computed subshares

[[1, 168], [2, 124], [3, 162], [4, 55], [5, 42], [6, 129], [7, 89], [8, 161], [9, 118], [10, 199], [11, 177], [12, 58], [13, 81], [14, 19], [15, 111], [16, 130], [17, 82], [18, 206], [19, 42], [20, 62], [21, 39], [22, 212], [23, 121], [24, 5]]

Participant 13 chooses three degree polynomial $g_{13}(x) = 122 + 161x + 220x^2 + 97x^3$
Participant 13 computed subshares

[[1, 134], [2, 3], [3, 78], [4, 9], [5, 145], [6, 136], [7, 98], [8, 147], [9, 166], [10, 38], [11, 112], [12, 38], [13, 165], [14, 143], [15, 88], [16, 116], [17, 110], [18, 186], [19, 227], [20, 116], [21, 202], [22, 135], [23, 31], [24, 6]].

Participant 14 chooses three degree polynomial $g_{14}(x) = 137 + 164x + 94x^2 + 2x^3$.
Participant 14 computed subshares

[[1, 164], [2, 158], [3, 131], [4, 95], [5, 62], [6, 44], [7, 53], [8, 101], [9, 200], [10, 129], [11, 133], [12, 224], [13, 181], [14, 16], [15, 207], [16, 67], [17, 74], [18, 7], [19, 111], [20, 165], [21, 181], [22, 171], [23, 147], [24, 121]] .

Participant 15 chooses three degree polynomial $g_{15}(x) = 204 + 223x + 172x^2 + 61x^3$.
Participant 15 computed subshares

[[1, 194], [2, 195], [3, 107], [4, 63], [5, 196], [6, 173], [7, 127], [8, 191], [9, 32], [10, 16], [11, 43], [12, 13], [13, 59], [14, 81], [15, 212], [16, 119], [17, 168], [18, 26], [19, 59], [20, 167], [21, 17], [22, 208], [23, 174], [24, 48]].

Participant 16 chooses three degree polynomial
$g_{16}(x) = 172 + 140x + 211x^2 + 213x^3$.

Participant 16 computed subshares

[[1, 37], [2, 204], [3, 87], [4, 32], [5, 152], [6, 94], [7, 204], [8, 129], [9, 215], [10, 109], [11, 157], [12, 6], [13, 2], [14, 25], [15, 188], [16, 138], [17, 221], [18, 84], [19, 73], [20, 68], [21, 182], [22, 62], [23, 54], [24, 38]].

Participant 17 chooses three degree polynomial $g_{17}(x) = 54 + 176x + 17x^2 + 14x^3$.
Participant 17 computed subshares

[[1, 28], [2, 120], [3, 181], [4, 62], [5, 80], [6, 86], [7, 164], [8, 165], [9, 173], [10, 39], [11, 80], [12, 147], [13, 91], [14, 229], [15, 179], [16, 25], [17, 84], [18, 207], [19, 12], [20, 49], [21, 169], [22, 223], [23, 62], [24, 3]]

Participant 18 chooses three degree polynomial
$g_{18}(x) = 103 + 11x + 180x^2 + 95x^3$
Participant 18 computed subshares

[[1, 156], [2, 207], [3, 127], [4, 20], [5, 223], [6, 141], [7, 111], [8, 4], [9, 157], [10, 208], [11, 28], [12, 187], [13, 90], [14, 74], [15, 10], [16, 2], [17, 154], [18, 104], [19, 189], [20, 47], [21, 15], [22, 197], [23, 231], [24, 221]].

Participant 19 chooses three degree polynomial $g_{19}(x) = 194 + 99x + 47x^2 + 132x^3$.
Participant 19 computed subshares

[[1, 6], [2, 5], [3, 51], [4, 4], [5, 190], [6, 3], [7, 2], [8, 47], [9, 231], [10, 181], [11, 223], [12, 217], [13, 23], [14, 200], [15, 142], [16, 175], [17, 159], [18, 187], [19, 119], [20, 48], [21, 67], [22, 36], [23, 48], [24, 196]].

Participant 20 chooses three degree polynomial $g_{20}(x) = 55 + 118x + 159x^2 + 98x^3$
Participant 20 computed subshares

[[1, 197], [2, 80], [3, 59], [4, 23], [5, 94], [6, 161], [7, 113], [8, 72], [9, 160], [10, 33], [11, 46], [12, 88], [13, 48], [14, 48], [15, 210], [16, 190], [17, 110], [18, 92], [19, 25], [20, 31], [21, 232], [22, 51], [23, 76], [24, 196]]

Participant 21 chooses three degree polynomial
$g_{21}(x) = 133 + 108x + 105x^2 + 96x^3$.

Participant 21 computed subshares

[[1, 209], [2, 139], [3, 33], [4, 1], [5, 153], [6, 133], [7, 51], [8, 17], [9, 141], [10, 67], [11, 138], [12, 231], [13, 223], [14, 224], [15, 111], [16, 227], [17, 216], [18, 188], [19, 20], [20, 55], [21, 170], [22, 9], [23, 148], [24, 231]].

Participant 22 chooses three degree polynomial $g_{22}(x) = 177 + 79x + 109x^2 + 213x^3$.

Participant 22 computed subshares

[[1, 112], [2, 145], [3, 156], [4, 25], [5, 98], [6, 22], [7, 143], [8, 108], [9, 30], [10, 22], [11, 197], [12, 202], [13, 150], [14, 154], [15, 94], [16, 83], [17, 1], [18, 194], [19, 76], [20, 226], [21, 58], [22, 151], [23, 152], [24, 174]].

Participant 23 chooses three degree polynomial

$g_{23}(x) = 120 + 210x + 30x^2 + 110x^3$.

Participant 23 computed subshares

[[1, 4], [2, 142], [3, 29], [4, 92], [5, 59], [6, 124], [7, 15], [8, 159], [9, 51], [10, 118], [11, 88], [12, 155], [13, 47], [14, 191], [15, 82], [16, 147], [17, 114], [18, 177], [19, 64], [20, 202], [21, 86], [22, 143], [23, 101], [24, 154]].

Participant 24 chooses three degree polynomial

$g_{24}(x) = 62 + 167x + 119x^2 + 187x^3$.

Participant 24 computed subshares

[[1, 69], [2, 38], [3, 159], [4, 156], [5, 219], [6, 72], [7, 138], [8, 141], [9, 38], [10, 19], [11, 41], [12, 61], [13, 36], [14, 156], [15, 145], [16, 193], [17, 24], [18, 61], [19, 28], [20, 115], [21, 46], [22, 11], [23, 200], [24, 104]].

After computing their respective subshares, each participant communicates them secretly to every other participant and himself.

6.3.3 Commitments

Group 1 computes commitments using equation 6.2 and 6.3 and makes them public

$$cm_{10} = 381, cm_{11} = 325, cm_{12} = 363, cm_{13} = 432$$

$$cm_{20} = 210, cm_{21} = 30, cm_{22} = 343, cm_{23} = 284$$

$$cm_{30} = 90, cm_{31} = 425, cm_{32} = 222, cm_{33} = 169$$

$$cm_{40} = 230, cm_{41} = 91, cm_{42} = 206, cm_{43} = 66$$

$$cm_{50} = 423, cm_{51} = 64, cm_{52} = 447, cm_{53} = 317$$

$$cm_{60} = 441, cm_{61} = 122, cm_{62} = 68, cm_{63} = 175$$

$$cm_{70} = 221, cm_{71} = 324, cm_{72} = 432, cm_{73} = 291$$

Similarly, group $2, 3, 4$ computes their respective commitments and makes them public.

6.3.4 Subshare Verification

Suppose participant 6 computes subshare $[3, 59]$ and sends secretly to participant 3. By using participant 6 commitments, participant 3 can verify his respective subshare as shown below:

$$a^{y_{63}} = cm_{60} \times ((cm_{61})^3)^1 \times ((cm_{62})^3)^2 \times ((cm_{63})^3)^3 \mod p$$
$$4^{59} = 441 \times ((122)^3)^1 \times ((68)^3)^2 \times ((175)^3)^3 \mod 467$$
$$252 = 252 \mod 467$$

Similarly, all subshares can be verified by using commitments.

6.3.5 Original Share Construction and Key Reconstruction

All participants compute their respective share from equation 6.4. Now, the threshold number of participants can reconstruct the key by using Lagranges interpolation 6.5.

Group 1 shares:
[1, 116] [2, 163] [3, 209] [4, 109] [5, 184] [6, 56] [7, 46]

The threshold number of shares: 4
[1, 116] [4, 109] [5, 184] [6, 56] [7, 46]
Key Reconstruction: 213

Group 2 shares:
[8, 9] [9, 33] [10, 206] [11, 150] [12, 186] [13, 169]
The threshold number of shares: 4
[8, 9] [9, 33] [12, 186] [13, 169]
Key Reconstruction: 213

Group 3 shares:
[14, 187] [15, 95] [16, 214] [17, 166] [18, 39]
The threshold number of shares: 4
[14, 187] [16, 214] [17, 166] [18, 39]
Key Reconstruction: 213

Group 4 shares:
[19, 154] [20, 133] [21, 64] [22, 35] [23, 134] [24, 216]
The threshold number of shares: 4
[19, 154] [21, 64] [22, 35] [23, 134]
Key reconstruction: 213

6.4 Correctness and Security Analysis

In this section, the correctness of Subshare Verification, the correctness of the Key Reconstruction, and Security Analysis are discussed .

6.4.1 Correctness of Subshare Verification

Each participant verifies his respective subshare by using commitments from section 6.2.3.

The verifiability of the subshare can be checked iff

$$a^{y_{ij}} = (cm_{i0})(cm_{i1})^{id_j}(cm_{i2})^{{id_j}^2} \cdots (cm_{i(t-1)})^{{id_j}^{t-1}} \bmod p.$$

$$
\begin{aligned}
a^{y_{ij}} &= (a^{s_i})(a^{c_{i1}})^{id_j}(a^{c_{i2}})^{{id_j}^2}\cdots(a^{c_{i(t-1)}})^{{id_j}^{t-1}} \bmod p, \text{from equation 6.2, and 6.3} \\
&= a^{s_i+c_{i1}id_j+c_{i2}{id_j}^2+\cdots+c_{i(t-1)}{id_j}^{t-1}} \bmod p \\
&= a^{g_i(id_j)} \bmod p, \text{from equation 6.1} \\
&= a^{y_{ij}} \bmod p.
\end{aligned}
$$

Commitments cm_{i0}, cm_{ij} do not reveal any information about secret [34] as the same reduces to solving discrete logarithm problem. From equation 6.2 given cm_{i0}, a and p, getting s_i is difficult. Similarly from 6.3 given cm_{ij}, a and p, finding out c_{ik} is difficult.

6.4.2 Correctness of the Key Reconstruction

Without loss of generality, assume that $p_1, p_2, \cdots, p_t$ desire to rebuild the key, say S'. The secret polynomial $g(x)$ can be computed by using Lagranges interpolation formula $g(x)$ as

$$
\begin{aligned}
g(x) &= \sum_{i=1}^{t} L_i(x)(y_i) \bmod q \text{ (from equation 6.5)} \\
&= \sum_{i=1}^{t} L_i(x) \sum_{j=1}^{n}(y_{ji}) \bmod q \text{ (from equation 6.4)} \\
&= \sum_{i=1}^{t}\sum_{j=1}^{n} L_i(x)y_{ji} \bmod q \\
&= \sum_{j=1}^{n}\sum_{i=1}^{t} L_i(x)y_{ji} \bmod q \\
&= \sum_{j=1}^{n} g_j(x) \bmod q
\end{aligned}
$$

where $L_i(x) = \frac{(x-1)\cdots(x-(i-1))(x-(i+1))\cdots(x-t)}{(i-1)\cdots(i-(i-1))(i-(i+1))\cdots(i-t)}$.

The key $S' = g(0) \bmod q = \sum_{j=1}^{n} g_j(0) \bmod q = \sum_{i=1}^{n} s_i \bmod q = S$.

6.4.3 Security Analysis

Theorem 6.4.1. *Any t or more participants can rebuild the key in every group.*

Proof. Without loss of generality, suppose t participants $p_1, p_2 \cdots p_t$ cooperate with their shares $y_1, y_2 \cdots y_t$ to reconstruct the key. The threshold number of participants computes the Lagrange polynomial $g(x)$ with their public information $id_1, id_2 \cdots id_t$. Thus, the polynomial $g(x)$ can be computed as

$$g(x) = \sum_{i=1}^{t} L_i(x) y_i \bmod q.$$

and hence the key can be recovered as

$$g(0) = S = \sum_{i=1}^{n} s_i \bmod q.$$

as proved in section 6.4.2

Atleast t or more participants produce polynomial $g(x) = S + c_1 x + c_2 x^2 + \cdots + c_{t-1} x^{t-1}$ of degree $t-1$ to share S, each of $S, c_1, \cdots, c_{t-1}$ is choosen with probability $\frac{1}{q}$.

Using basic linear algebra we can see the $g(x)$ is the only solution. There must be definitely q^t possible tuples(t values in F_q), and reconstruction with Lagrange polynomials that gives a matching polynomial of degree $t-1$.

However, only q^t polynomials exist with a degree of $t-1$. As a result, there can only be one matching polynomial per tuple; otherwise, we would run out of polynomials.

□

Theorem 6.4.2. *In every group, lesser than t participants will not be able to reconstruct the key.*

Proof. Without loss of generality, suppose that $t-1$ or less participants $p_1, p_2 \cdots p_{t-1}$ want to reconstruct the key. They use Lagranges interpolation method to reconstruct the polynomial, say $g_1(x)$. However $g_1(x)$ is a polynomial of atmost degree $t-2$. Hence $g_1(x) \neq g(x)$ and so the secret cannot be recovered. Suppose that they choose random share $y_t{}'$ and follow the same procedure as above. They can find a polynomial, say $g_2(x)$ of degree $t-1$. But the probability of choosing a correct share $y_t{}'$ for which $g_2(x) = g(x)$ is $\frac{1}{q}$.

Given the shares $y_1, y_2, y_3, \cdots, y_{t-1}$ there is just one value of the missing share y_t that would give that S for each possible value of S. As a result, the $t-1$ shares $y_1, y_2, y_3, \cdots, y_{t-1}$ provide no information on the key because all feasible key values are still feasible and equiprobable at that moment.

Take note of the word "equiprobability" in this sentence. For a given key S, the counting argument states that there exist precisely q^{t-1} polynomials $g(x)$ with a maximum degree of $t-1$, such that $S = g(0)$. We wouldn't realize that much about $t-1$ shares, i.e., knowledge of $t-1$ values $y_i = g(id_i)$, to identify a single polynomial $g(x)$; in reality, we have exactly q matching polynomials, for each feasible value of the key S. Hence, in every group, lesser than t participants will not be able to reconstruct the key.

□

Theorem 6.4.3. *Any participant can verify the correctness of the secret s by using their share after reconstructing it.*

Proof. By using shares y_i, at least the threshold number of the participants can reconstruct the secret polynomial

$$g(x) = \sum_{i=1}^{n} g_i(x) \bmod q.$$

Assuming that any participant want to know if the secret is correct, the participant computes $k_i = g(id_i) \bmod q$. If both k_i and his share y_i are equal, then the secret is valid; otherwise, not.

□

6.5 Comparison

Various group key agreement protocols are available in the literature that can generate a group key among users of a single group. However, to the best of our knowledge, our proposed MGKAP is the first of its kind, which can generate a group key among users of multigroup. Table 1 compares the first five rows that correspond to schemes proposed by Li et al.[59], Alzahrani et al.[2], Cui et al. [23], Sun et al.[79], and Zhang et al.[93], where key generation happens in a distributed environment but still need a key generation center for information distribution. Also, they use techniques such as bilinear pairings, whereas most other schemes use SSS. However, none of these

Table 6.1: Comparisons among various schemes

S.No	Scheme	Distributed	KGC	SSS	Multi Group
1	Li et al. [59]	Yes	Yes	No	No
2	Alzahrani et al. [2]	Yes	Yes	No	No
3	Cui et al. [23]	Yes	Yes	No	No
4	Sun et al.[79]	Yes	Yes	No	No
5	Zhang et al. [93]	Yes	Yes	No	No
6	Q Zhang et al.[94]	Yes	No	No	No
7	Cheng et al.[19]	Yes	No	No	No
8	Cao et al. [16]	Yes	Yes	Yes	No
9	Harn et al.[39]	Yes	No	Yes	No
10	Proposed MGKAP	Yes	No	Yes	Yes

SSS: Secret Sharing Scheme, KGC: Key Generation Center,
MGKAP: Multi Group Key Agreement Protocol using a Secret Sharing Scheme.

schemes are suitable for multigroup communication. Schemes proposed by [94] and Cheng et al.[19] function in a distributed environment without the need for KGC. This is relatively better compared to the above schemes, as the problems that arise with KGC can be avoided. It uses techniques such as bilinear pairings, but still not suitable for a multigroup environment. Cao et al.[16] and Harn et al.[39] schemes are suitable for a distributed environment and use SSS to distribute group key related information. While Cao et al.[16] uses a KGC, Harn et al.[39] do not need a KGC, and both are not suitable for multigroup environments. Our proposed MGKAP scheme is very much suitable for key agreement protocols in a distributed multigroup environment and does not need a KGC, as the users themselves share the information required for group key generation in the form of subshares using SSS.

6.6 Summary

A novel Multi Group Key Agreement Protocol using a Secret Sharing Scheme in a distributed environment is proposed in this chapter. Participants from multiple groups generate subshares for every participant and communicate them secretly. Feldman commitments allow each participant to check the validity of their subshare after receiving subshares from all other participants. Any threshold t or more participants in every group can reconstruct the secret, and participants across multiple groups can communicate securely using this one time session key. The secret cannot be reconstructed in a group if there are $t-1$ or fewer participants. This scheme would be more useful in collaborative applications where people from multiple groups must communicate securely.

The correctness of the scheme and security analysis are discussed.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis contributes significantly to the sphere of cryptography, focusing primarily on Centralized Secret Sharing Schemes and Distributed Group Key Agreement Protocols. The aim has been to enhance three essential facets of the cryptographic protocols - security, privacy, and computational efficiency. These improvements have been devised with a specific intent to be applicable across both centralized and distributed computational platforms.

For centralized secret sharing schemes, a Multipartite Verifiable Secret Sharing Scheme has been introduced. This approach utilizes principles from the Chinese Remainder Theorem (CRT), which subsequently empower participants to verify their individual shares. This authentication acts as a preventive measure, negating any possible maliciousness from the dealer. In this scheme, Commitments ensure that information is not leaked to reveal secrets and it is a perfect secret sharing scheme.

Additionally, to counter the challenges of privacy, verification, and share renewal in the existing Compartmented Secret Sharing Schemes (CSSS), we put forth a Compartmented Proactive Secret Sharing Scheme. This mechanism ensures the privacy of participants while enabling the renewal and validation of shares. In this scheme, at least a threshold number of participants can reconstruct the secret. Less than the threshold number of participants cannot reconstruct the secret.

Despite these advancements, a common vulnerability in centralized secret sharing

schemes is the risk of a single point of failure. Addressing this concern, our research has ventured into the realm of distributed group key agreement protocols. When it comes to distributed systems, we put forth a novel scheme known as the Elliptic Curve Secret Sharing Scheme (ECSSS). This scheme serves as a foundation upon which we developed a unique Authenticated Distributed Group Key Agreement Protocol, integrating the ECSSS into its design. This protocol is highly efficient in managing key distribution, particularly in resource constrained environments. It is achieved through the use of smaller key sizes, rapid computation, and minimal storage requirements. Security analysis of the scheme ensures that only valid user can reconstruct the key.

We further contribute to this domain by formulating a Multi-Group Key Agreement Protocol. This protocol, grounded in a secret sharing scheme, enables several groups to agree upon a common key for secure communication. In this scheme, at least a threshold number of participants can reconstruct the group key in every group. Less than the threshold number of participants cannot reconstruct the group key.

Throughout our research, we have placed significant emphasis on the correctness and security analysis of the proposed schemes. The security of centralized secret sharing schemes is predicated on the discrete logarithm problem, while distributed group key agreement protocols leverage the Elliptic Curve Discrete Logarithm Problem (ECDLP). We firmly believe that the schemes we have proposed serve as secure and effective mechanisms to protect confidential data and generate group communication keys.

In conclusion, this thesis significantly enriches the field of cryptography by proposing innovative techniques and protocols in both Centralized Secret Sharing and Distributed Group Key Agreement. These developments are crucial in improving security, privacy, and computational efficiency across numerous practical applications.

7.2 Future Work

For future research, it would be insightful to delve into share renewal within the Multipartite Verifiable Secret Sharing Scheme and share recovery in the Compartmented Proactive Secret Sharing Scheme. Furthermore, it would be beneficial to investigate in dynamic environment where participants join and leave the group dynamically. Authenticated Distributed Group Key Agreement Protocol using ECSSS. Lastly, an exciting research avenue would be to develop solutions for instances where multiple groups meet

the threshold in the Multi-Group Key Agreement Protocol, which could potentially result in unintended secret reconstruction.

References

[1] Noura Al Ebri, Joonsang Baek, and Chan Yeob Yeun. **Study on Secret Sharing Schemes (SSS) and their applications**. In *2011 International Conference for Internet Technology and Secured Transactions*, pages 40–45. IEEE, 2011. ()

[2] Bander A Alzahrani, Shehzad Ashraf Chaudhry, Ahmed Barnawi, Abdullah Al-Barakati, and Mohammed H Alsharif. **A privacy preserving authentication scheme for roaming in IoT-based wireless mobile networks**. *Symmetry*, **12**(2):287, 2020. (5, 29, 93, 94)

[3] Howard Anton and Chris Rorres. *Elementary linear algebra: applications version*. John Wiley & Sons, 2013. (73, 74)

[4] Charles Asmuth and John Bloom. **A modular approach to key safeguarding**. *IEEE transactions on information theory*, **29**(2):208–210, 1983. (3, 18, 22)

[5] Li Bai and XuKai Zou. **A proactive secret sharing scheme in matrix projection method**. *International Journal of Security and Networks*, **4**(4):201–209, 2009. ()

[6] Rana Barua, Ratna Dutta, and Palash Sarkar. **Extending Joux's protocol to multi party key agreement**. In *Progress in Cryptology-INDOCRYPT 2003: 4th International Conference on Cryptology in India, New Delhi, India, December 8-10, 2003. Proceedings 4*, pages 205–217. Springer, 2003. (28)

[7] Shimshon Berkovits. **How to broadcast a secret**. In *Advances in Cryptology—EUROCRYPT'91: Workshop on the Theory and Application of Cryptographic Techniques Brighton, UK, April 8–11, 1991 Proceedings 10*, pages 535–541. Springer, 1991. (29)

[8] VP Binu and A Sreekumar. **Threshold multi secret sharing using elliptic curve and pairing**. *arXiv preprint arXiv:1603.09524*, 2016. (30)

[9] George Robert Blakley. **Safeguarding cryptographic keys**. *Proc. of the National Computer Conference1979*, **48**:313–317, 1979. (3)

[10] Jens-Matthias Bohli. **A framework for robust group key agreement**. In *Computational Science and Its Applications-ICCSA 2006: International Conference, Glasgow, UK, May 8-11, 2006, Proceedings, Part III 6*, pages 355–364. Springer, 2006. (28)

[11] Timo Brecher, Emmanuel Bresson, and Mark Manulis. **Fully Robust Tree-Diffie-Hellman Group Key Exchange.** In *CANS*, **9**, pages 478–497. Springer, 2009. (28)

[12] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. **Provably authenticated group Diffie-Hellman key exchange—the dynamic case**. In *Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings*, pages 290–309. Springer, 2001. (28)

[13] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. **Provably secure authenticated group Diffie-Hellman key exchange**. *ACM Transactions on Information and System Security (TISSEC)*, **10**(3):10–es, 2007. ()

[14] Ernest F Brickell. **Some ideal secret sharing schemes**. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 468–475. Springer, 1989. ()

[15] Mike Burmester and Yvo Desmedt. **A secure and efficient conference key distribution system**. In *Advances in Cryptology—EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9–12, 1994 Proceedings 13*, pages 275–286. Springer, 1995. (27)

[16] Chunjie Cao, Chao Yang, Jianfeng Ma, and Sangjae Moon. **Constructing UC secure and constant-round group key exchange protocols via secret sharing**. *EURASIP Journal on Wireless Communications and Networking*, **2008**:1–9, 2008. (5, 29, 75, 76, 94)

[17] Ting-Yi Chang, Min-Shiang Hwang, and Wei-Pang Yang. **A new multi-stage secret sharing scheme using one-way function**. *ACM SIGOPS Operating Systems Review*, **39**(1):48–55, 2005. ()

[18] Oinam Bidyapati Chanu, Appala Naidu Tentu, and V Ch Venkaiah. **Multi-Stage Multi-Secret Sharing Schemes Based on Chinese Remainder Theorem**. In *Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)*, page 17. ACM, 2015. ()

[19] Qing-Feng Cheng, Chuan-Gui Ma, and Fu-Shan Wei. **Analysis and improvement of a new authenticated group key agreement in a mobile environment**. *annals of telecommunications-annales des télécommunications*, **66**(5-6):331–337, 2011. (5, 29, 30, 75, 76, 94)

[20] Hung-Yu Chien. **Elliptic curve cryptography-based RFID authentication resisting active tracking**. *Wireless Personal Communications*, **94**(4):2925–2936, 2017. ()

[21] 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**(12):2762–2765, 2000. ()

[22] Kyu Young Choi, Jung Yeon Hwang, and Dong Hoon Lee. **Efficient ID-based group key agreement with bilinear maps**. In *Public Key Cryptography–PKC 2004: 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004. Proceedings 7*, pages 130–144. Springer, 2004. (28)

[23] Wenchao Cui, Rui Cheng, Kehe Wu, Yuling Su, and Yuqing Lei. **A Certificateless Authenticated Key Agreement Scheme for the Power IoT**. *Energies*, **14**(19):6317, 2021. (5, 29, 93, 94)

[24] Ed Dawson and Diane Donovan. **The breadth of Shamir's secret-sharing scheme**. *Computers & Security*, **13**(1):69–78, 1994. ()

[25] He Debiao, Chen Jianhua, and Hu Jin. **An ID-based client authentication with key agreement protocol for mobile client–server environment on ECC with provable security**. *Information Fusion*, **13**(3):223–230, 2012. (29)

[26] Yvo Desmedt and Tanja Lange. **Revisiting pairing based group key exchange**. In *Financial Cryptography and Data Security: 12th International Conference, FC 2008, Cozumel, Mexico, January 28-31, 2008. Revised Selected Papers 12*, pages 53–68. Springer, 2008. (28)

[27] Whitfield Diffie. **New direction in cryptography**. *IEEE Trans. Inform. Theory*, **22**:472–492, 1976. (5)

[28] Xinjun Du, Ying Wang, Jianhua Ge, and Yumin Wang. **An improved ID-based authenticated group key agreement scheme**. *Cryptology ePrint Archive*, 2003. (28)

[29] Barun Duari and Debasis Giri. **An ideal and perfect (t, n) Multi-secret sharing scheme based on finite geometry**. In *Information Technology and Applied Mathematics*, pages 85–94. Springer, 2019. ()

[30] Ratna Dutta, Rana Barua, and Palash Sarkar. **Provably secure authenticated tree based group key agreement**. In *Information and Communications Security: 6th International Conference, ICICS 2004, Malaga, Spain, October 27-29, 2004. Proceedings 6*, pages 92–104. Springer, 2004. (28)

[31] Petr Dzurenda, Sara Ricci, Raúl Casanova Marqués, Jan Hajny, and Petr Cika. **Secret sharing-based authenticated key agreement protocol**. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, pages 1–10, 2021. (30)

[32] Oriol Farràs, Jaume Martí-Farré, and Carles Padró. **Ideal multipartite secret sharing schemes**. *Journal of cryptology*, **25**(3):434–463, 2012. (3)

[33] Oriol Farras, Carles Padró, Chaoping Xing, and An Yang. **Natural generalizations of threshold secret sharing**. *IEEE transactions on information theory*, **60**(3):1652–1664, 2014. ()

[34] Paul Feldman. **A practical scheme for non-interactive verifiable secret sharing**. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 427–438. IEEE, 1987. (3, 82, 91)

[35] Hossein Ghodosi, Josef Pieprzyk, and Rei Safavi-Naini. **Secret sharing in multilevel and compartmented groups**. In *Australasian Conference on Information Security and Privacy*, pages 367–378. Springer, 1998. (4, 24, 44)

[36] S Goldwasser, M Ben-Or, and A Wigderson. **Completeness theorems for non-cryptographic fault-tolerant distributed computing**. In *Proc. of the 20th STOC*, pages 1–10, 1988. (27)

[37] Xiaozhuo Gu, Youjian Zhao, and Jianzu Yang. **Reducing rekeying time using an integrated group key agreement scheme**. *Journal of Communications and Networks*, **14**(4):418–428, 2012. (28)

[38] Rakel Haakegaard and Joanna Lang. **The elliptic curve diffie-hellman (ecdh)**. *Online at https://koclab. cs. ucsb. edu/teaching/ecc/project/2015Projects/Haakegaard+ Lang. pdf*, 2015. (5, 74)

[39] Lein Harn and Changlu Lin. **Efficient group Diffie–Hellman key agreement protocols**. *Computers & Electrical Engineering*, **40**(6):1972–1980, 2014. (4, 5, 29, 30, 32, 74, 75, 76, 77, 94)

[40] Lien Harn and Changlu Lin. **Authenticated group key transfer protocol based on secret sharing**. *IEEE transactions on computers*, **59**(6):842–846, 2010. (29)

[41] Jingmin He and Edward Dawson. **Multistage secret sharing based on one-way function**. *Electronics Letters*, **30**(19):1591–1592, 1994. ()

[42] Javier Herranz and Germán Sáez. **New results on multipartite access structures**. *IEE Proceedings-Information Security*, **153**(4):153–162, 2006. ()

[43] Amir Herzberg, Stanisław Jarecki, Hugo Krawczyk, and Moti Yung. **Proactive secret sharing or: How to cope with perpetual leakage**. In *annual international cryptology conference*, pages 339–352. Springer, 1995. (3)

[44] Ching-Fang Hsu and Lein Harn. **Multipartite secret sharing based on CRT**. *Wireless personal communications*, **78**(1):271–282, 2014. (4, 24, 34)

[45] Sorin Iftene. **Secret Sharing Schemes with Applications in Security Protocols.** *Sci. Ann. Cuza Univ.*, **16**:63–96, 2006. (3, 20, 22, 33, 34)

[46] Sorin Iftene. **General secret sharing based on the chinese remainder theorem with applications in e-voting**. *Electronic Notes in Theoretical Computer Science*, **186**:67–84, 2007. ()

[47] Ingemar Ingemarsson, Donald Tang, and C Wong. **A conference key distribution system**. *IEEE Transactions on Information theory*, **28**(5):714–720, 1982. (27)

[48] SK Hafizul Islam and GP Biswas. **An improved pairing-free identity-based authenticated key agreement protocol based on ECC**. *Procedia Engineering*, **30**:499–507, 2012. (29)

[49] Stanislaw Jarecki, Jihye Kim, and Gene Tsudik. **Flexible robust group key agreement**. *IEEE Transactions on Parallel and Distributed Systems*, **22**(5):879–886, 2010. (28)

[50] Antoine Joux. **A one round protocol for tripartite Diffie–Hellman**. In *Algorithmic Number Theory: 4th International Symposium, ANTS-IV Leiden, The Netherlands, July 2-7, 2000. Proceedings 4*, pages 385–393. Springer, 2000. (28)

[51] Jonathan Katz and Moti Yung. **Scalable protocols for authenticated group key exchange**. *Journal of Cryptology*, **20**:85–113, 2007. (28)

[52] Kamer Kaya and Ali Aydin Selçuk. **A verifiable secret sharing scheme based on the chinese remainder theorem**. In *International Conference on Cryptology in India*, pages 414–425. Springer, 2008. (3, 22, 33, 34)

[53] Yongdae Kim, Adrian Perrig, and Gene Tsudik. **Group key agreement efficient in communication**. *IEEE transactions on computers*, **53**(7):905–921, 2004. (27)

[54] Allen Klinger. **The vandermonde matrix**. *The American Mathematical Monthly*, **74**(5):571–574, 1967. (15)

[55] Elisavet Konstantinou. **An efficient constant round id-based group key agreement protocol for ad hoc networks**. In *Network and System Security: 7th International Conference, NSS 2013, Madrid, Spain, June 3-4, 2013. Proceedings 7*, pages 563–574. Springer, 2013. (28)

[56] Chi Sung Laih, Jau Yien Lee, and Lein Harn. **A new threshold scheme and its application in designing the conference key distribution cryptosystem**. *Information Processing Letters*, **32**(3):95–99, 1989. (29)

[57] Patrick PC Lee, John CS Lui, and David KY Yau. **Distributed collaborative key agreement and authentication protocols for dynamic peer groups**. *IEEE/ACM Transactions On Networking*, **14**(2):263–276, 2006. (27)

[58] Chih-Hung Li and Josef Pieprzyk. **Conference key agreement from secret sharing**. In *Information Security and Privacy: 4th Australasian Conference, ACISP'99 Wollongong, NSW, Australia, April 7–9, 1999 Proceedings 4*, pages 64–76. Springer, 1999. (29)

[59] Juyan Li, Zhiqi Qiao, and Jialiang Peng. **Asymmetric group key agreement protocol based on blockchain and attribute for industrial internet of things**. *IEEE Transactions on Industrial Informatics*, **18**(11):8326–8335, 2022. (5, 75, 76, 93, 94)

[60] Changlu Lin, L Harn, and Dingfeng Yea. **Ideal Hierarchical (t, n) Secret Sharing Schemes**. In *Proceedings of the Fifth International Conference on Information Assurance and Security (IAS 09), Xian, China*. Citeseer, 2009. ()

[61] Duo Liu, Dongping Huang, Ping Luo, and Yiqi Dai. **New schemes for sharing points on an elliptic curve**. *Computers & Mathematics with Applications*, **56**(6):1556–1561, 2008. (30)

[62] Yanxiao Liu, Qindong Sun, Yichuan Wang, Lei Zhu, and Wenjiang Ji. **Efficient group authentication in RFID using secret sharing scheme**. *Cluster Computing*, **22**:8605–8611, 2019. (29, 30)

[63] A Menezes, P Oorschot, and S Vanstone. **, Handbook of Applied Cryptography, CRC Press, Boca Raton**. 1996. (26)

[64] Keju Meng, Fuyou Miao, Yu Ning, Wenchao Huang, Yan Xiong, and Chin-Chen Chang. **A proactive secret sharing scheme based on Chinese remainder theorem**. *Frontiers of Computer Science*, **15**:1–10, 2021. ()

[65] Torben Pryds Pedersen. **Non-interactive and information-theoretic secure verifiable secret sharing**. In *Annual International Cryptology Conference*, pages 129–140. Springer, 1991. ()

[66] Li Qiong, Wang Zhifang, Niu Xiamu, and Sun Shenghe. **A non-interactive modular verifiable secret sharing scheme**. In *Communications, Circuits and Systems, 2005. Proceedings. 2005 International Conference on*, **1**, pages 84–87. IEEE, 2005. (3, 20, 22, 33)

[67] KR Raghunandan, Radhakrishna Dodmane, K Bhavya, NS Krishnaraj Rao, and Aditya Kumar Sahu. **Chaotic-Map Based Encryption for 3D**

Point and 3D Mesh Fog Data in Edge Computing. *IEEE Access*, **11**:3545–3554, 2022. (30)

[68] Germán Sáez. **Generation of key predistribution schemes using secret sharing schemes**. *Discrete Applied Mathematics*, **128**(1):239–249, 2003. (29)

[69] David A Schultz, Barbara Liskov, and Moses Liskov. **Mobile proactive secret sharing**. In *Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*, pages 458–458, 2008. ()

[70] Adi Shamir. **How to share a secret**. *Communications of the ACM*, **22**(11):612–613, 1979. (3, 16, 17, 22)

[71] Maryam Sheikhi-Garjan, Mojtaba Bahramian, and Christophe Doche. **Threshold verifiable multi-secret sharing based on elliptic curves and Chinese remainder theorem**. *IET Information Security*, **13**(3):278–284, 2019. (30)

[72] Jian Shen, Shaohua Chang, Jun Shen, Qi Liu, and Xingming Sun. **A lightweight multi-layer authentication protocol for wireless body area networks**. *Future generation computer systems*, **78**:956–963, 2018. (29)

[73] Kyung-Ah Shim. **A round-optimal three-party ID-based authenticated key agreement protocol**. *Information Sciences*, **186**(1):239–248, 2012. (28)

[74] Abdulhadi Shoufan and Sorin A Huss. **High-performance rekeying processor architecture for group key management**. *ieee transactions on computers*, **58**(10):1421–1434, 2009. (27)

[75] Gustavus J Simmons. **How to (really) share a secret**. In *Conference on the Theory and Application of Cryptography*, pages 390–448. Springer, 1988. (25)

[76] David G Steer, Leo Strawczynski, Whitfield Diffie, and M Wiener. **A secure audio teleconference system**. In *Advances in Cryptology—CRYPTO'88: Proceedings 8*, pages 520–528. Springer, 1990. ()

[77] Michael Steiner, Gene Tsudik, and Michael Waidner. **Diffie-Hellman key distribution extended to group communication**. In *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, pages 31–37, 1996. (27)

[78] Douglas Robert Stinson and Maura Paterson. *Cryptography: theory and practice*. CRC press, 2018. (74)

[79] Da-Zhi Sun. **Security and Privacy Analysis of Vinoth et al.'s Authenticated Key Agreement Scheme for Industrial IoT**. *Symmetry*, **13**(10):1952, 2021. (5, 29, 93, 94)

[80] Haiyan Sun, Qiaoyan Wen, Hua Zhang, and Zhengping Jin. **A novel pairing-free certificateless authenticated key agreement protocol with provable security**. *Frontiers of Computer Science*, **7**(4):544–557, 2013. (28)

[81] Tamir Tassa. **Hierarchical threshold secret sharing**. *Journal of Cryptology*, **20**(2):237–264, 2007. ()

[82] Tamir Tassa and Nira Dyn. **Multipartite secret sharing by bivariate interpolation**. *Journal of Cryptology*, **22**(2):227–258, 2009. ()

[83] Wen-Guey Tzeng. **A secure fault-tolerant conference-key agreement protocol**. *IEEE Transactions on Computers*, **51**(4):373–379, 2002. (27)

[84] Wen-Guey Tzeng and Zhi-Jia Tzeng. **Round-efficient conference key agreement protocols with provable security**. In *Advances in Cryptology—ASIACRYPT 2000: 6th International Conference on the Theory and Application of Cryptology and Information Security Kyoto, Japan, December 3–7, 2000 Proceedings*, pages 614–627. Springer, 2000. (27)

[85] Kai Wang, Xukai Zou, and Yan Sui. **A multiple secret sharing scheme based on matrix projection**. In *2009 33rd Annual IEEE International Computer Software and Applications Conference*, **1**, pages 400–405. IEEE, 2009. ()

[86] Shiuh-Jeng Wang, Yuh-Ren Tsai, and Jian-Jhih Shen. **Dynamic threshold multi-secret sharing scheme using elliptic curve and bilinear maps**. In *2008 Second International Conference on Future Generation Communication and Networking*, **2**, pages 405–410. IEEE, 2008. (30)

[87] Avi Wigderson, MB Or, and S Goldwasser. **Completeness theorems for noncryptographic fault-tolerant distributed computations**. In *Proceedings of the 20th Annual Symposium on the Theory of Computing (STOC'88)*, pages 1–10, 1988. ()

[88] Qianhong Wu, Yi Mu, Willy Susilo, Bo Qin, and Josep Domingo-Ferrer. **Asymmetric group key agreement**. In *Advances in Cryptology-EUROCRYPT 2009: 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings 28*, pages 153–170. Springer, 2009. (28)

[89] Jen-Ho Yang and Chin-Chen Chang. **An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem**. *Computers & security*, **28**(3-4):138–143, 2009. (29)

[90] Eun-Jun Yoon and Kee-Young Yoo. **Robust id-based remote mutual authentication with key agreement scheme for mobile devices on ecc**. In *2009 International conference on computational science and engineering*, **2**, pages 633–640. IEEE, 2009. (29)

[91] Jiangtao Yuan, Jing Yang, Chenyu Wang, Xingxing Jia, Fang-Wei Fu, and Guoai Xu. **A new efficient hierarchical multi-secret sharing scheme based on linear homogeneous recurrence relations**. *Information Sciences*, **592**:36–49, 2022. ()

[92] S Rao YV and Chakravarthy Bhagvati. **CRT based threshold multi secret sharing scheme**. *International Journal of Network Security*, **16**(3):194–200, 2014. ()

[93] Lei Zhang, Qianhong Wu, Josep Domingo-Ferrer, Bo Qin, and Zheming Dong. **Round-efficient and sender-unrestricted dynamic group key agreement protocol for secure group communications**. *IEEE Transactions on Information Forensics and Security*, **10**(11):2352–2364, 2015. (5, 75, 76, 93, 94)

[94] Qikun Zhang, Yong Gan, Lu Liu, Xianmin Wang, Xiangyang Luo, and Yuanzhang Li. **An authenticated asymmetric group key agreement based on attribute encryption**. *Journal of Network and Computer Applications*, **123**:1–10, 2018. (5, 30, 75, 76, 94)

[95] Qikun Zhang, Liang Zhu, Ruifang Wang, Jianyong Li, Junling Yuan, Tiancai Liang, and Jun Zheng. **Group key agreement protocol among terminals of the intelligent information system for mobile edge computing**. *International Journal of Intelligent Systems*. (5, 28, 76)

[96] Lidong Zhou, Fred B Schneider, and Robbert Van Renesse. **APSS: Proactive secret sharing in asynchronous systems**. *ACM transactions on information and system security (TISSEC)*, **8**(3):259–286, 2005. ()

CENTRALIZED SECRET SHARING SCHEMES AND DISTRIBUTED GROUP KEY AGREEMENT PROTOCOLS

by Rolla Subrahmanyam

Submission date: 24-Jul-2023 03:29PM (UTC+0530)
Submission ID: 2136016835
File name: PhD_Thesis__Plagiarism_17MCPC06.pdf (743.06K)
Word count: 25643
Character count: 123591

CENTRALIZED SECRET SHARING SCHEMES AND DISTRIBUTED GROUP KEY AGREEMENT PROTOCOLS

ORIGINALITY REPORT

35%	13%	35%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES Overall Similarity Index = 35 - (14+5+4+3+3) = 6 %

1	Rolla Subrahmanyam, N Rukma Rekha, Y V Subba Rao. "Authenticated Distributed Group Key Agreement Protocol Using Elliptic Curve Secret Sharing Scheme", IEEE Access, 2023 Publication This publication is by the student.	14%
2	"Computer Networks and Inventive Communication Technologies", Springer Science and Business Media LLC, 2022 Publication This publication is by the student	5%
3	"Intelligent Sustainable Systems", Springer Science and Business Media LLC, 2023 Publication This publication is by the student.	4%
4	Rolla Subrahmanyam, N. Rukma Rekha, Y. V. Subba Rao. "Authenticated Distributed Group Key Agreement Protocol Using Elliptic Curve Secret Sharing Scheme", IEEE Access, 2023 Publication This publication is by the student.	3%
5	link.springer.com Internet Source This publication is by the student.	3%

Associate Professor
School of CIS
Prof. C.R. Rao Road,
Central University
Hyderabad-46. (India)
N. R. Rekha

6	Submitted to University of Hyderabad, Hyderabad Student Paper	1 %
7	doaj.org Internet Source	<1 %
8	www.maths.uq.edu.au Internet Source	<1 %
9	Lecture Notes in Computer Science, 2015. Publication	<1 %
10	repository.bilkent.edu.tr Internet Source	<1 %
11	Lecture Notes in Computer Science, 2000. Publication	<1 %
12	Atanu Basu, Indranil Sengupta, Jamuna Kanta Sing. "Secured hierarchical secret sharing using ECC based signcryption", Security and Communication Networks, 2012 Publication	<1 %
13	P. Thiyagarajan, Prasanth Kumar Thandra, J. Rajan, S. A. V. Satyamurthy, G. Aghila. "Shamir Secret Sharing Scheme with Dynamic Access Structure (SSSDAS): case study on nuclear power plant", Kerntechnik, 2015 Publication	<1 %
14	Na Wang, Yuanyuan Cai, Junsong Fu, Xiqi Chen. "Information Privacy Protection Based	<1 %

	on Verifiable (t, n)-Threshold Multi-Secret Sharing Scheme", IEEE Access, 2020 Publication	
15	onlinelibrary.wiley.com Internet Source	<1 %
16	"Information Security and Privacy", Springer Science and Business Media LLC, 2018 Publication	<1 %
17	Lecture Notes in Computer Science, 2009. Publication	<1 %
18	Schridde, Christian. "Secure Session Framework: An Identity-based Cryptographic Key Agreement and Signature Protocol", Philipps-Universität Marburg, 2010. Publication	<1 %
19	9pdf.net Internet Source	<1 %
20	Alireza Hodjat. "Network Security", Sensor Network Operations, 04/04/2006 Publication	<1 %
21	C. Blundo, A. De Santis, A.G. Gaggia, U. Vaccaro. "New bounds on the information rate of secret sharing schemes", IEEE Transactions on Information Theory, 1995 Publication	<1 %
22	Lecture Notes in Computer Science, 2008. Publication	<1 %

23	archive.org Internet Source	<1%
24	Huang, XiaoFang, Qi Tao, BaoDong Qin, and ZhiQin Liu. "Multi-Authority Attribute Based Encryption Scheme with Revocation", 2015 24th International Conference on Computer Communication and Networks (ICCCN), 2015. Publication	<1%
25	ebin.pub Internet Source	<1%
26	JiKai Teng, ChuanKun Wu, ChunMing Tang. "An ID-based authenticated dynamic group key agreement with optimal round", Science China Information Sciences, 2011 Publication	<1%
27	"Advances in Cryptology - EUROCRYPT 2007", Springer Science and Business Media LLC, 2007 Publication	<1%
28	L. Jani Anbarasi, G. S. Anandha Mala. "Survey and Analysis of Visual Secret Sharing Techniques", International Review on Computers and Software (IRECOS), 2014 Publication	<1%
29	Submitted to United World College of South East Asia Student Paper	<1%

30	Zhan Zhang, Xinghua Li, Yunwei Wang, Yinbin Miao, Ximeng Liu, Jian Weng, Robert H. Deng. "TAGKA: Threshold Authenticated Group Key Agreement Protocol Against Member Disconnect for UANET", IEEE Transactions on Vehicular Technology, 2023 Publication	<1 %
31	fr.slideserve.com Internet Source	<1 %
32	Samaneh Mashhadi, Zahra Saeedi. "A (t,n)-Secret image sharing with steganography based on Rook polynomial and LWE problem", Multimedia Tools and Applications, 2023 Publication	<1 %
33	Submitted to University of Exeter Student Paper	<1 %
34	"Artificial Intelligence and Security", Springer Science and Business Media LLC, 2020 Publication	<1 %
35	Abdul Basit, V. Ch. Venkaiah, Salman Abdul Moiz. "Chapter 31 Multi-secret Sharing Scheme Using Modular Inverse for Compartmented Access Structure", Springer Science and Business Media LLC, 2020 Publication	<1 %
36	Amos Beimel. "On Matroids and Nonideal Secret Sharing", IEEE Transactions on	<1 %

	Information Theory, 06/2008 Publication	
37	D'Arco, P.. "Properties and constraints of cheating-immune secret sharing schemes", Discrete Applied Mathematics, 20060201 Publication	<1 %
38	Submitted to Kensington College of Business - Brunei Student Paper	<1 %
39	h.web.umkc.edu Internet Source	<1 %
40	silo.pub Internet Source	<1 %
41	downloads.hindawi.com Internet Source	<1 %
42	hdl.handle.net Internet Source	<1 %
43	www.academypublisher.com Internet Source	<1 %
44	"Information Technology and Applied Mathematics", Springer Science and Business Media LLC, 2019 Publication	<1 %
45	"Coding and Cryptology", Springer Science and Business Media LLC, 2011 Publication	<1 %

46	Communications in Computer and Information Science, 2015. Publication	<1 %
47	Guoai Xu, Jiangtao Yuan, Guosheng Xu, Zhongkai Dang. "An Efficient Compartmented Secret Sharing Scheme Based on Linear Homogeneous Recurrence Relations", Security and Communication Networks, 2021 Publication	<1 %
48	Submitted to CSU, San Jose State University Student Paper	<1 %
49	DACIBERG LIMA GONÇALVES, JOHN GUASCHI. "THE BRAID GROUP $B_{n,m} (\mathbb{S}^{2})$ AND A GENERALISATION OF THE FADELL–NEUWIRTH SHORT EXACT SEQUENCE", Journal of Knot Theory and Its Ramifications, 2011 Publication	<1 %
50	iotone.ir Internet Source	<1 %
51	uwspace.uwaterloo.ca Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		