Signature Schemes for Privacy Protection in Blockchains

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

S DEVIDAS

SCHOOL OF COMPUTER AND INFORMATION SCIENCES
UNIVERSITY OF HYDERABAD
(P.O) CENTRAL UNIVERSITY
HYDERABAD, TELANGANA-500046, INDIA

April 2023

CERTIFICATE

This is to certify that the thesis entitled **"Signature Schemes for Privacy Protection in Blockchains"** submitted by **S Devidas** bearing **Reg. No. 16MCPC13** in partial fulfilment of the requirements for award of **Doctor of Philosophy in Computer Science** is a bonafide work carried out by him under our supervision and guidance.

This 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 award of any degree or diploma. The student has the following publications before submission of the thesis for adjudication and has produced evidence for the same.

1. "Identity verifiable ring signature scheme for privacy protection in blockchain". International Journal of Information Technology. (Accepted) (Scopus Indexed)
2. "A decentralized group signature scheme for privacy protection in a blockchain". International Journal of Applied Mathematics and Computer Science, 31(2):353–364, 2021. (SCIE and Scopus Indexed)
3. "Dynamic decentralized group signature scheme for privacy protection in blockchain". In International Conference on Innovative Computing and Communications, LNNS, volume 492, pages 745–760. Springer, 2023. (Scopus Indexed)
4. "Decentralized framework for record-keeping system in government using hyperledger fabric". In Proceedings of the Third International Conference on Computational Intelligence and Informatics, Advances in Intelligent Systems and Computing, volume 1090, pages 663–672. Springer, 2020. (Scopus Indexed)

Further, the student has passed the following courses towards the fulfillment of course work requirements for Ph.D.

Course Code	Course Name	Credits	Pass/Fail
CS 801	Data Structures and Algorithms	4	Pass
CS 802	Operating Systems and Programming	4	Pass
IT 811	Secure Computing	4	Pass
AI 820	Digital Image Processing	4	Pass

N. Rukma Rekha

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

Y.V. Subba Rao 27.04.2023

Dr. Subba Rao Y.V.
Supervisor
Associate Professor
School of CIS
Prof. C.R. Rao Road,
Central University
Hyderabad-46. (India)

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

DECLARATION

I, **S DEVIDAS**, hereby declare that this thesis entitled **" Signature Schemes for Privacy Protection in Blockchains "** submitted by me under the guidance and supervision of **Dr. N Rukma Rekha & Dr. Subba Rao Y.V** 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.

Signature of the Student

Name: **S DEVIDAS**

Reg.No.: **16MCPC13**

Dedicated to my parents, wife and family members, for everything what I am today and above all, devoted to The Almighty God!

Abstract

Name of the student: **S DEVIDAS** Roll No: **16MCPC13**
Degree for which submitted: **Ph.D.** Department/School: **SCIS**
Thesis title: **Signature Schemes for Privacy Protection in Blockchains**
Thesis supervisors: **N. Rukma Rekha & Subba Rao Y.V.**
Month and year of thesis submission: **April 2023**

A peer-to-peer electronic cash system introduced as bitcoin in 2008 focused on the need to overcome the un-trusted third-party's involvement in online financial transactions apart from many other aspects. Today, decentralized public ledger technology in a peer-to-peer network is becoming popular and is being called blockchain technology. Blockchains are majorly categorized as public blockchains, consortium blockchains and private blockchains. Blockchain technology has received considerable attention recently with the continuous development in financial and non-financial domains and its security features. This led to a flurry of advancements in various applications using blockchains. Blockchain offers various security features such as transparency, immutability, and traceability in business transactions . Its ability to work seamlessly across various fields is becoming very popular, and one such field is e-auction.

An e-auction is a method of selling products or services in which a vendor offers the items or services for sale, and potential buyers submit their bids. Each vendor expects to obtain competitive rates from the buyers who want to purchase these items or services at the lowest possible cost. To achieve the participants' privacy and fairness exchange, it is necessary to have a trustworthy third party to host the e-auction. However, trusted third party holds a lot of important information about the users. As a result, it constantly raises the possibility of threats ranging from single-point attacks to collusion attacks, and it is also challenging to identify a totally trusted organization that can take on such a function in practice.

Many e-auction protocols have recently been implemented on top of blockchain to benefit

from the blockchain's decentralization, transparency, immutability, and verifiability features and eliminate the protocol drawbacks caused by the centralized third party. Transactions play a vital role in any blockchain system. Transactions are first created by various users and are broadcast on the network, which is then validated by the network, and then all such validated transactions form a block to be added to the blockchain finally. The transaction data structure can encode the transfer of value from one party to some other party in the system, and every transaction is a public entry in the publicly available global ledger blockchain. Since every transaction in the blockchain network is publicly available, anyone in the network can inspect and analyze them. This public nature of the transaction brings the problem of identity privacy. Adversaries or curious parties can guess the real identity of the signers from the series of transactions by transaction graph analysis, and through the big data analysis. To address user identity privacy, transaction privacy, and other privacy issues in blockchains, various privacy-preserving schemes have been proposed and available in the literature. In general, in applications like e-auctions, users prefer to protect their identity privacy when using the service. In literature, digital signature schemes were popular in addressing signer's privacy issues . Few signature schemes themselves are capable of giving the signer anonymity. These kinds of signature schemes are called anonymous signatures; ring signatures and group signatures are two such schemes.

In public blockchain systems, every node has equal participation. The ring signature scheme is more suitable for public blockchain systems since it doesn't require a group manager to set up the group. The ring signature is more significant in terms of privacy because of its unconditional anonymity and unforgeability. Although users benefit from the everlasting anonymity of ring signatures, applications such as e-auctions fail to use ring signatures as there is a dire need to determine the real signer in such applications. Hence, this thesis proposes an identity-verifiable ring signature scheme for user identity privacy protection and to determine the real signer in e-auction-like applications deployed on public blockchains. In the proposed ($IVRSS$) the real signer of the message can voluntarily prove himself to all the other participants in the ring that he is the actual signer. All the remaining members can verify the correctness of the actual signer. The proposed scheme makes the blockchain-based e-auction protocols more transparent and provides user identity privacy.

In private blockchain systems, a controlling authority is required to add and remove groups and to dynamically add new participants and remove certain participants from the existing group. The group signature scheme is more suitable for private blockchain systems since

it requires a group manager to set up the group. Group signature schemes play a vital role in protecting the identity privacy of a group member who signs the messages using the group signature. However, in the existing group signature schemes, the centralized group manager controls all the participants and these managers can be malicious. The managers may take biased decision when there is a dispute among the group members or while revealing the identity of a group member. To overcome the trust issues related to centralized group managers and to improve user's privacy, a decentralized group signature scheme (DGSS) is proposed by decentralizing the role of group manager. The proposed DGSS is more suitable for blockchain-based sealed-bid auctions.

However, the limitation of DGSS is that it works only for sealed-bid auctions and static domains, where group members are fixed and will not permit new members to join and existing members to leave the network. In this thesis, DGSS is extended to propose a dynamic decentralized group signature scheme (DDGSS) which allows the group members to join and revoke at run-time. The performance of DGSS is also improved by reducing the number of multiplications to make it suitable for user identity privacy protection in light-weight blockchains or memory constraint devices. The proposed DDGSS is more suitable for blockchain-based open-bid auctions.

On the other hand, various issues are preventing industries from fully adopting the blockchain technology, one of the major issues is scalability. Scalability refers to a system's capacity to handle an increasing quantity of work while remaining stable. According to the literature, organisations need to improve their capacity to handle an increased amount of transactions or workload. In general, performance enhancement of Hyperledger Fabric fall into two categories: (i) architectural redesign and (ii) bottleneck reduction. In this thesis, we have adopted the first approach and redesigned the hyperledger fabric architecture to improve its performance and also, a decentralized scalable framework is proposed for record keeping in Government organizations.

Acknowledgements

गुरु ब्रम्हा गुरु विष्णुः गुरु देवो महेश्वरः ।
गुरुः साक्षात् परंब्रह्म तस्मै श्री गुरवे नमः ॥

—by **Veda Vyāsa** in *Skanda Purana*

Meaning: My Guru (teacher) is the representative of *Brahma*, *Vishnu*, and *Shiva*. He creates, sustains knowledge and destroys the weeds of ignorance. I offer my obeisance to my Gurus.

The above shloka precisely capture my thoughts about my teachers **Dr. N. Rukma Rekha** and **Dr. Subba Rao Y.V.** They put a lot of faith in me and accepted me as their student. They are the great mentors, researchers, developers, and administrators. Despite their very busy schedule, they always offered their help and advice related to research, teaching, or even non-technical aspects coming from their abundant knowledge and experience. They doesn't forget to pat the back on our achievements but also stand tall and support their students on each tumble (such as a paper rejection). Behind their technical avatar, they are the persons with very pure heart who are always ready to offer their help on any matter. I am indebted to them for the great amount of energy and time they invested in supervising me.

My gratitude is also due to my DRC member **Prof. Chakravarthy Bhagvati** for his extraordinary support to the whole process of this research. I was immensely benefited from his continuous assistance in finalizing the research problem, developing skills in interpretive approach as well as his sincere guidance throughout the various stages of this study. His assessment, comments and suggestions in strengthening the methodological approach were significant to improve the benchmark of this thesis.

I would like to thank one of the other DRC member **Prof. Salman Abdul Moiz** for his very useful suggestions and counselling in overcoming various bottlenecks during the study and also for his comments on the draft of my thesis. His cooperation and assistance in the overall presentation and all his precautionary measures made the methodology of this thesis comprehensive. I appreciate his continuous encouragement.

I am deeply indebted to all my teachers, who have greatly shaped my personality of what I am today and their ethically-embedded teachings have influenced my life to a great degree to put into practice all my life, which also reflects in my research as well. I would like to take this opportunity to thank all teachers and mentors of my life.

For several reasons, I express my sincere thanks to my wife **Mrs. Thirumala**. She is my motivator during tough situations. I extend deepest salutation to my whole family members for providing the needful facilities and emotional assistance. Their support imprints me the property to be flexible as water and rigid as rock to balance my twisted research life. I also thank my lab-mates **Mr. N. Shiva Krishna, Mr. G. Thirupathi, Mrs. Khaleda Afroz, Mr. B. Srinivas and Mr. Rolla Subramanyam** for building an enthusiastic environment and engaging in critical discussions related to research in both inside and outside the lab. I am very much thankful to my friends **Mr. G. R.Anil, Abdul Basith, Mr. Ajith Kumar Sahu, Mr. Govind, Mr. Anand, Mr. Abhimanyu and Mr. Pravindra**. They have always been appreciating the passion of my profession and unboundedly keeping in touch during my bad times.

I wish to express that I am indebt and owe so much to my mother **Mrs. Meni** and father **Mr. Govind** such that my depth of expressing acknowledgement will always be small when compare to their contributions of parenting. The unconditional sacrifices and determined motivation is what I have learnt from them at every stage of my life and this has enforced me to become a responsible citizen at first place. With an immense pride this doctoral dissertation is dedicated to my parents for providing me the strong roots to standalone and sustain in such long journey of life.

Finally, I bow The Almighty for being source of unknown power and strength in my life.

S DEVIDAS

Contents

List of Figures

List of Tables

Abbreviations

ZKP	**Z**ero **K**nowledge **P**roof
DGSS	**D**ecentralized **G**roup **S**ignature **S**cheme
MAC	**M**ssage **A**thentication **C**ode
DoS	**D**enial **o**f **S**ervice
HE	**H**omomorphic **E**ncryption
ABE	**A**ttribute **B**ased **E**ncryption
MPC	**M**uly **P**arty **C**omputation
SMPC	**S**ecure **M**uly **P**arty **C**omputation
NIZK	**N**on-interactive **Z**ero **K**owledge
TEE	**T**rusted **E**xecution **E**nvironment
HLF	**H**yper **L**edger **F**abric
PBFT	**P**ractical **B**yzantine **F**ault **T**olerance
MSP	**M**embership **S**ervice **P**rovider
CA	**C**ertificate **A**uthority
TLS	**T**ransport **L**ayer **S**ecurity
CC	**C**hain **C**code
DB	**D**atabase
DLP	**D**iscrete **L**ogarithm **P**roblem
RM	**R**egistration **M**anager
AM	**A**uction **M**anager
IM	**I**dentity **M**anager
DDGSS	**D**ynamic **D**ecentralized **G**roup **S**ignature **S**cheme
ECDSA	**E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm

RSA	**R**on Rivest Adi **S**hamir Leonard **A**dleman
ARS	**A**ccountable **R**ing **S**ignature
LRS	**L**inkable **R**ing **S**ignature
DVRS	**D**esignated **V**erifier **R**ing **S**ignature
DIVRS	**D**esignated **I**dentity **V**erifier **R**ing **S**ignature
IVRSS	**I**dentity **V**erifiable **R**ing **S**ignature **S**cheme
IoT	**I**internet **o**f **T**hings
SCM	**S**upply **C**hain **M**anagement
FPS	**F**ile **P**rocessing **S**ystem
ACID	**A**tomicity **C**onsistency **I**solation **D**urability

Symbols

m	message to be signed
σ	signature
v	initialization value
T_j	group manager
U_i	group member
x_i	private key
y_i	public key
k_{ij}	random integer from Z_q^*
x_{RM}	private key of registration manager
y_{RM}	public key of registration manager
x_{AM}	private key of auction manager
y_{AM}	public key of auction manager
x_{IM}	private key of identity manager
y_{IM}	public key of identity manager
GNO_i	identity of good

Chapter 1

Introduction

Prior to the development of databases and database management systems, File Processing Systems (FPS) were the most frequent method of storing digital data. FPS, as the name implies, store data in a variety of files that were accessible by applications. When compared to obsolete or constrained systems, the benefits of file processing systems are well accepted. But they are having lots of disadvantages like data redundancy, data access issues, data isolation, data integrity, atomicity, concurrency and security etc and the invention of database technology addressed all the above said disadvantages [107]. Network connection is critical for centralized databases and bottlenecks might happen when there is a lot of traffic. However, several people have limited access to the data since there is only one copy of it, there may be a single point of failure and it is maintained in a single location. Distributed databases address the drawbacks of centralized databases and provides various number of benefits, including data availability, reliability, performance, and modular development.

In the context of a distributed database [122], the atomic data transfers should satisfy the classical ACID (atomicity, consistency, isolation, and durability) properties. However, there are noticeable additional problems such as scalability and security. The necessity for secure atomic data transmission (transaction) is becoming increasingly important as the number of connected devices grows. Currently, lot of data is generated from the different networks. While solving the challenge of the scalability of these atomic transactions,

distributed databases ignore their security issue. Even though the security component is simple to comprehend, it is challenging to put that into action in an effective manner. There are a variety of security issues, such as transaction modification, non-forwarding of transactions, forwarding of fraudulent transactions, and so on [20]. Furthermore, adopting blockchain technology as a distributed database overcomes many of the security issues because of its key qualities, including decisions that are based on consensus, which ensures that all transactions are neither irreversible and immutable.

1.1 Context and Motivation

Blockchains are distributed ledgers that keep track of a growing collection of ordered records known as blocks. A distributed ledger is a sort of distributed database which is synchronized and cloned across its participants. Each block creates an ordered chain that contains timestamp, list of transactions, nonce, and a link to the preceding block. Each node utilizes a pair of public keys and private keys to broadcast transactions onto the blockchain. To begin, the node creates and signs a transaction, which it then broadcasts to the blockchain network. Each blockchain node verifies all transactions before broadcasting them to its peers, discarding any that are invalid. To record these valid transactions, the network's miner nodes create a new block and broadcast it to their peers, who validate it before appending it to the distributed ledger. This cycle continues forever. Each blockchain uses a different consensus method to resolve multiple states or forks in the network.

Based on their usage and various characteristics, blockchains are classified into three categories. First, public or permissionless blockchains where permission is not required for participants to join the network [29]. Participants may read and write transactions, maintain the copy of the ledger and take part in the consensus [164]. Second, private or permissioned blockchains, which are intended for use by a specific organization. Participants are invited to be part of the network and play a role in the decentralized maintenance of the blockchain [29]. Only authorized entities are eligible to join the network and maintain the blocks, which differentiates private blockchains from public blockchains [164]. Third is, a consortium blockchain [172], in which instead of allowing all users inside the blockchain

network to be part of the consensus process or enabling a single organization to have full control, it permits a predetermined set of organizations to monitor the consensus process.

Three fundamental elements—a peer-to-peer network, cryptographic techniques, and a consensus mechanism—form the foundation of the blockchain's unique architecture. These three characteristics give the blockchain the ability to handle not only digital currency but also a wide range of other applications. The following are some of the benefits of blockchain technology [112]: verifiability, immutability, transparency and permanent. Even though the blockchains are having lot of benefits they are still suffering from many challenges. The following are major challenges of blockchains [172]: security, privacy, energy consumption, scalability, interoperability and no regulation. Among these, security and privacy are major concerns with respect to the sensitive data that is stored on blockchains. Further, blockchain privacy is classified into identity privacy, transaction privacy and smart contract privacy. Scalability is another major factor that need to be focused on, as the usage of blockchains for various applications has drastically increased. The scalability is further classified as horizontal and vertical scalability.

In blockchains, privacy enable users to protect transaction details and personal data from unauthorized access, while ensuring the transparency and verifiability of the network. The transaction privacy enables to conceal the details of a transaction such as the amount transferred, identities of the the sender and receiver and the purpose of the transaction from any unauthorized users. Smart contract privacy enables to maintain the terms of a smart contract and its details to be confidential and private between the users. This includes the different rules, conditions of the contract and logic as well as the user identities. Identity privacy provides users to have control on their identity. The user identity privacy is important in blockchain to protect their personal data from being misused by unauthorized users. It helps in preventing fraud, identity theft, and any other malicious actions that can harm business and users. Moreover, identity privacy protection increases trust in blockchain-based applications, promoting their adoption for various other applications.

Another challenge in blockchain is its scalability, it enables the ability of a blockchain network to handle an increasing amount of transactions and users without compromising its performance. It is essential to assure the adoption of applications built on blockchain

and to make them usable in the real world with needs for high throughput and low latency. In blockchain, the scalability is classified into horizontal and vertical scalability [117]. The horizontal scalability refers to the ability of a network to increase its capacity and transaction throughput by adding more nodes to the network. The vertical scalability refers to the ability of a network to increase its transaction capacity and throughput by upgrading the hardware of the existing nodes or redesigning framework of the blockchain.

Among all the challenges our focus is on identity privacy protection of the users and vertical scalability. A blockchain transaction includes transaction ID, users address, trade value, timestamp, and the sender signature. Because of the open nature of the blockchain network, data mining techniques may be used to track transactions flow and extract individuals personal identities or any other sensitive information [58]. Anonymous signatures enable anonymity to the signer and these schemes can be used for signer identity privacy protection [172]. The group signature and ring signature are the two prominent and widely used anonymous signing techniques. Scalability problems have emerged as a result of the significant growth in the number of users of blockchain systems and have had a significant impact on the advancement of blockchain. In general, performance enhancement of blockchains fall into two categories [40]: (i) architectural redesign and (ii) bottleneck reduction.

1.2 Research Study Objectives

Blockchain technology is gaining acceptance in a wide range of financial and non-financial sectors. However, there are also privacy and scalability issues that might prevent blockchain's widespread use. Therefore, the objectives of this thesis are to propose different signature schemes for user identity privacy protection in blockchain and also to address scalability issues.

1.2.1 Research Objectives

Despite the fact that there has been a significant amount of study on the blockchain user identity privacy protection and improving its performance, there are research gaps that

must be solved. The research objectives of this study are specified below:

1. To propose an anonymous signature scheme for identity privacy protection in public blockchain.
2. To propose an anonymous signature scheme for identity privacy protection in private blockchains for static networks.
3. To propose an anonymous signature scheme for identity privacy protection in private blockchains for dynamic networks.
4. To increase transaction throughput in blockchains.

1.3 Research Problem

An e-auction is a method of selling products or services in which a vendor offers the items or services for sale and potential buyers submit their bids. Each vendor expects to obtain competitive rates from the buyers, who want to purchase these items or services at the lowest possible cost. Further, in e-auctions bidding can be placed in two different ways: (1) closed bid (2) open bid. A closed bid e-auction is a sort of online auction in which participants bids are kept secret from the other participants until the auction is over. Another sort of online auction called an open bid e-auction allows bidders to view the bid amount and, if they so choose, can submit a higher bid.To achieve the participants' privacy and the fairness exchange, it is necessary to have a trustworthy third party to host the e-auction. However, trusted third party holds a lot of important information about the users. As a result, it constantly raises the possibility of threats ranging from single-point attacks to collusion attacks, and it is also challenging to identify a totally trusted organisation that can take on such a function in practise. Many e-auction protocols have recently been implemented on top of blockchain to benefit from the decentralization, transparency, immutability, and verifiability features of the blockchain and to eliminate the e-auction protocol's drawbacks caused by the trusted third party. [95, 158, 38, 103, 90, 130, 72]. But as discussed above, blockchains are still facing few challenges. So, w.r.t. security and

privacy, as of now the blockchain is a conceptual trustworthy entity that we can rely on for availability and accuracy but not for privacy.

To address user identity privacy, transaction privacy and other privacy issues in blockchains, the literature contains a variety of privacy-preserving schemes that have been put forth. The currency mixing mechanism borrowed from the idea of Chaum [36] was proposed to protect users' addresses from being linked. Mixing multiple unrelated input addresses and output addresses of the users making hard for the outsiders to relate the output and input of the transaction. A centralized coin mixing service with audit function Mixcoin is designed by Bonneau et al. [26], which gives anonymous payment in Bitcoin and bitcoin-like cryptocurrencies. A CoinJoin scheme is proposed by Maxwell Gregory et al. [104], which is another method for anonymization of bitcoin transactions. The idea of joint payment was motivated by this scheme. If a user wants to make a payment, he finds another user who also wants to make a payment, and they can jointly make a payment together in one transaction. However, these coin mixing schemes have the involvement of a trusted third party for providing mixing services. A zerocoin scheme based on zero-knowledge proof (ZKP) is proposed by Miers et al. [110] to address the identity leakage problem of user. In this scheme, addresses of both parties can be hidden by the user through the Zerocoin, which makes the transaction unlinkable. However, Zerocoin can only exchange and mint fixed-value currency, and the data of Zerocoin's ZKP is relatively large, which needs additional computational resources and blockchain storage.

The data in the blockchain is public and available to everyone. If user's private information from transaction is removed from the database then the privacy issue of the users is fundamentally resolved. From this idea, many off-chain payment schemes [44], [128], [111], [68] are proposed. However, all the existing off-chain transaction schemes implement anonymous transactions between users through third parties, resulting in the need for trust.

Many different versions of the digital signature schemes were proposed in the literature for message authentication, data integrity and non-repudiation. Anonymous signatures enable anonymity to the signer and can be used for signer identity privacy protection [172]. Hence, in this thesis to addresses the identity privacy issues resulting in blockchain-based applications, anonymous signature schemes are proposed.

The group signature and ring signature are the two prominent and widely used anonymous signing techniques. The ring signatures are more suitable to achieve unconditional anonymity in the permissionless blockchain systems. Although users benefit from the everlasting anonymity of ring signatures, applications such as e-auctions fail to use ring signatures as there is a dire need to determine the real signer in such applications [139]. Hence, an identity verifiable ring signature scheme (IVRSS) is proposed for user identity privacy protection and to determine the real signer of the message in e-auction like applications which are deployed on permissionless blockchains.

The most suitable signature scheme for permissioned blockchain systems is group signature as it requires a group manager to set up the group. Group signature schemes play an important role in preserving the identity privacy of the signer. However, the centralized group manager in the present group signature schemes has control over every participant, and these managers can be malicious. When there is a disagreement among group members or when announcing results, an identity of group member is required and managers may make biased decisions. A decentralized group signature scheme (DGSS) [48] is proposed by decentralizing the function of the group manager in order to address the trust issues associated with centralized group managers and to preserve user identity privacy. The proposed DGSS is more suitable for permissioned blockchain-based sealed-bid e-auctions.

To the best of our knowledge, DGSS [48] is the first scheme to decentralize the designated group manager to make it suitable for user identity privacy protection in permissioned blockchains. However, the shortcoming of the scheme is that it works only for sealed-bid e-auctions and static environments. In a static environment, all the group members have to be determined in its initiation phase itself and there is no mechanism to add or revoke a group member [13]. So, we extend that scheme to propose a dynamic decentralized group signature scheme (DDGSS) [47] that works for open-bid e-auctions and dynamic environments as well. Thus, the proposed scheme allows to join new members into the group and also allows to revoke any group member whenever it is required. It is observed that the performance of the proposed scheme is more efficient than DGSS. This is achieved by reducing the number of multiplications in the Verification algorithm which makes it more suitable for memory restricted environments or lightweight blockchains.

In order to address the scalability problems of existing blockchains various techniques are proposed in the literature. Blockbench [50] was given by Tien Tuan et al., and it assesses latency, throughput, adaptability and scalability to internal failure. The authors utilised Blockbench to conduct a thorough evaluation of three private blockchains: Parity, Ethereum, and Hyperledger Fabric. The outcomes show that these frameworks have a long way to go before they completely replace the existing database frameworks used in traditional data management. Furthermore, the frameworks allocated to the structural decisions at various stages of the blockchain's product stack have problems in execution. A.Baliga et al. [15] used a technique for testing, in which they investigated throughput and latency of hyperledger fabric (HLF) by exposing it to various workload configurations. They tune various chaincode settings and transaction to analyze how they effect latency using a setup of smaller scale benchmarks made especially for HLF. To increase transaction throughput, Christian Gorenflo et al. [67] redesigned the permissioned blockchain HLF. Wang et al. [51] demonstrated the challenge of scaling permissioned blockchain applications to efficiently serve a large number of consumers. According to the literature survey, there is a need to improve blockchain's scalability to handle a greater number of transactions. In this thesis, we have adopted the first approach and redesigned the hyperledger fabric architecture to improve its performance.

1.4 Thesis Contributions

This section outlines the scope of our study as well as the contributions made toward achieving the study's goals.

1. To achieve first objective, an identity verifiable ring signature scheme (IVRSS) is proposed in this thesis. To the best of our knowledge, all the existing signature schemes are not allowing to identify the real signers of the message as per requirement. In the proposed IVRSS, actual signer of the message has the capability to prove himself to other members whenever it is required. The proposed scheme is more suitable for user identity privacy protection in blockchain-based e-auctions. Without adding any additional overhead, the proposed scheme keeps all the characteristics of the original

ring signature scheme. In blockchain-based e-auction protocols, the auctioneer is responsible for revealing of winning bid and his identity to all the participants to make the e-auction system transparent.

2. To achieve the second objective, a decentralized group signature scheme (DGSS) is proposed by decentralizing the role of the designated group manager to address the trust issues of centralized group manager. In literature most of the existing group signature schemes [2] contain designated group manager as centralized party to reveal the identity of the group member on the requirement. The proposed scheme is based on the assumption that a group manager may be malicious. A malicious manager carries the risk of revealing identities and collusion attack. In our scheme, the designated group manager is decentralized to address the identity privacy issue of blockchain-based applications. The proposed DGSS suits more for bidder identity privacy protection in permissioned blockchain-based sealed-bid auctions.

3. To achieve the third objective, a dynamic decentralized group signature scheme (DDGSS) is proposed. The proposed scheme allows to add new members into the network at any time, and also to revoke the members from the network. This can be utilized to protect the identity privacy of the signers in real-time distributed applications. In addition to that, the performance of the proposed scheme is better compared to DGSS. This is achieved by reducing the number of multiplication operations in the verification algorithm of the proposed scheme. The proposed DDGSS suits more for bidder identity privacy protection in permissioned blockchain-based open-bid auctions.

4. To achieve the fourth objective, a decentralized framework for record-keeping system in government offices using Hyperledger Fabric is proposed [46]. The existing blockchain-based record-keeping systems have lot of communication overhead and resource utilization overhead. The proposed framework validates documents and creates blocks only for those that are valid. Clients are notified if their documents are invalid. In our proposed system both ordering service and validating service are combined together. This will lower the system's resource usage overhead as well. The approved transaction is immediately sent for validation rather than being sent

back to the client. This reduces the system's communication cost, and the approved transaction is immediately recorded into the blockchain afterwards.

1.5 Outline of the Thesis

This thesis is organized into seven chapters as follows:

In **Chapter 1**, introduction, a brief description of the problem, context and motivation, contributions, and thesis outline is described.

In **Chapter 2**, a literate survey on cryptographic primitives used in blockchains and blockchain technologies is carried out. The work done w.r.t. the blockchain privacy area till date is also discussed in this chapter.

In **Chapter 3**, an identity-verifiable ring signature scheme is proposed for signer identity privacy protection in public blockchain-based e-auction protocols. The security analysis and correctness proof of our proposed scheme are also carried out in this chapter.

In **Chapter 4**, a decentralized group signature scheme for signer identity privacy protection in private blockchain-based e-auction protocol is proposed. Along with the blockchain-based e-auction protocol, correctness proof and security analysis of our proposed scheme are also discussed in this chapter.

In **Chapter 5**, a dynamic decentralized group signature scheme for signer identity privacy protection in private blockchains is proposed. The security analysis and correctness proof of our proposed scheme are also carried out in this chapter.

In **Chapter 6**, a sacalable decentralized framework for record-keeping systems in Government using a permissioned blockchain called Hyperledger Fabric is discussed in this chapter.

In **Chapter 7**, concluding remarks are given, and future directions are discussed.

1.6 List of Publications

As result of this study, three papers were published (1-Journal, 2-Conferences). One more paper is communicated and revised version is also submitted to Scopus indexed journal. The published and communicated papers are reported below:

1. S Devidas, Subba Rao YV, and N Rukma Rekha. A decentralized group signature scheme for privacy protection in a blockchain. International Journal of Applied Mathematics and Computer Science, 31(2):353–364, 2021. (SCIE and Scopus Indexed)

2. S Devidas, N Rukma Rekha, and YV Subba Rao. Dynamic decentralized group signature scheme for privacy protection in blockchain. In Proceedings of the International Conference on Innovative Computing and Communications, LNNS, volume 492, pages 745–760. Springer, 2023. (Scopus Indexed)

3. S Devidas, N Rukma Rekha, and YV Subba Rao. Decentralized framework for record-keeping system in government using hyperledger fabric. In Proceedings of the Third International Conference on Computational Intelligence and Informatics, Advances in Intelligent Systems and Computing, volume 1090, pages 663–672. Springer, 2020. (Scopus Indexed)

4. S Devidas, N Rukma Rekha, and YV Subba Rao. Identity verifiable ring signature scheme for privacy protection in blockchain. International Journal of Information Technology, pages 0–0, 2023. (Accepted for Publication) (Scopus Indexed)

Chapter 2

Literature Survey

This chapter, gives a background to all cryptographic primitives, security properties and blockchain terminologies that are utilized in this thesis. The detailed survey on privacy properties and blockchain privacy protection schemes is also carried out in this chapter. The definitions used in section 2.1 and 2.2 are taken from [162] and [148].

2.1 Security Properties

This section describes many security principles and ideas that are utilized to describe computer system's security and privacy. These terms are being utilized in this thesis to describe the proposed schemes for blockchain identity privacy protection and applications.

2.1.1 Confidentiality

Confidentiality guarantees that information will not be accessed by unauthorised people, processes, or devices. Data confidentiality simply refers to the prevention of unauthorised access to data, which is generally done via the use of cryptographic schemes and access control. Information can be sent across an open channel with data confidentiality protection, ensuring that an adversary cannot access that information.

In addition to data, users may also be tied to confidentiality. In such situations, confidentiality has the following characteristics: (a) *anonymity*: is the process of concealing user identity such that it cannot be detected among a group of users; (b) *undetectability*: refers to the ability to conceal a user's actions such that they cannot be recognized as the cause of an activity and (c) *unlinkability*: refers to the attacker's inability to determine if two or more activities, data or identities are linked.

2.1.2 Integrity

When information is complete and uncorrupted, then it is said to have integrity. When information is vulnerable to corruption, damage, destruction, or other disruption, its integrity is compromised. Digital signature, hash functions and Message Authentication Codes (MAC) are used to ensure data integrity. When a computation uses the standard procedures and is free from unauthorised tampering, it may also be said to have integrity because this guarantees the correctness of the computed data. Integrity protection guarantees immutability of the message with in the context of communication.

2.1.3 Availability

Computer systems or authorized users can access information without being interrupted or obstructed, and they can get it in the format they need. If the system is unavailable or responding slowly, availability cannot be achieved. The Denial-of-Service (DoS) attack is a well-known threat on availability that can be addressed by replication but cannot be stopped using cryptographic techniques.

2.1.4 Authentication

It is a security mechanism that enables the legitimacy of a message's origin or an individual's identity. The method of validating the identity or an entity (person, device or process) to verify its origin and integrity of it is called an authentication. In the absence of an effective authentication mechanism, it is difficult to believe that a person is who

they claim to be, or that a communication is from whom it claimed to be. For verification, authentication systems frequently use digital signatures. Authorization comes after successful authentication. The terms authorization and authentication are closely interrelated. Authentication is concerned with verifying identities, where as authorisation is concerned with access privileges provided to user or the act of providing such privileges. Access control are commonly used techniques to enforce authorizing permissions.

2.1.5 Non-repudiation

The inability to deny receipt of any message is known as non-repudiation. To prevent dispute between the sender and the receiver of the message, both the sender and receiver are given proof of delivery and proof of the sender's identity. In digital communication via digital signatures and cryptographic commitments this property is commonly used.

2.1.6 Transparency

Transparency property needs that all information and conversations associated to the personal data processing be understandable and easily accessible. It also makes the data accessible to everyone at any time, ensuring that all transactions are transparent and open.

2.1.7 Accountability

Accepting responsibility for one's own actions is known as accountability. The traceability of all user activities done on any system is also known as accountability. Accountability can also be enhanced through the use of authentication, unique user identification and record-keeping.

2.2 Cryptographic Primitives

The security properties discussed in the previous section are achieved using cryptographic techniques. In this section, basic cryptographic primitives which are used later in this thesis are introduced.

2.2.1 Symmetric Key Cryptography

Symmetric key cryptography, often known as symmetric encryption uses the same secret key for both encryption and decryption. After encryption, message will be changed into a format that cannot be accessed without the secret key.

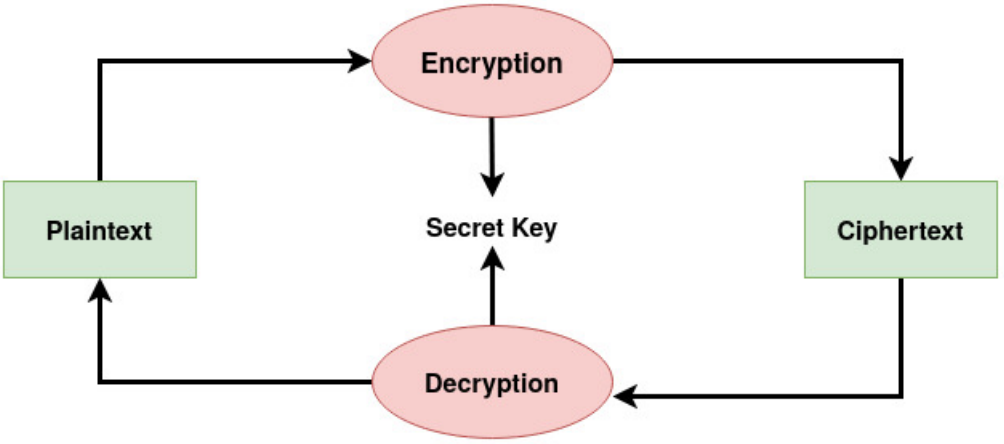


FIGURE 2.1: Symmetric key encryption

2.2.2 Asymmetric Key Cryptography

Asymmetric key cryptography, often known as asymmetric encryption uses different keys for both encryption and decryption. In this process, public and private keys are used for encryption and decryption respectively. Both the public and private keys must be related to the same user.

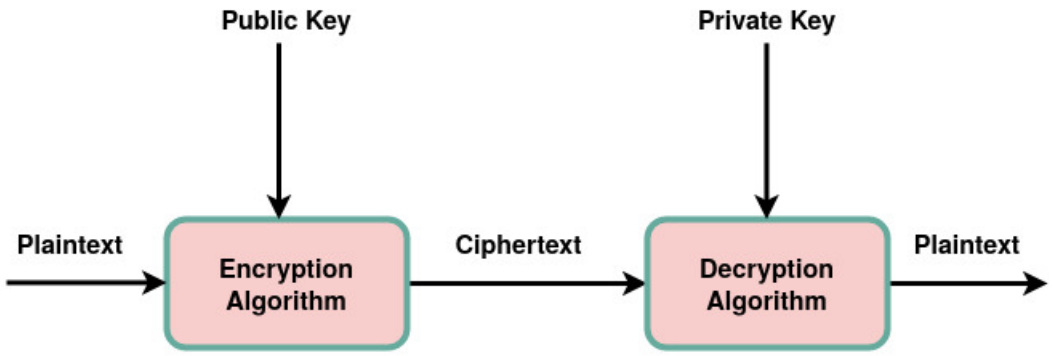


FIGURE 2.2: Asymmetric key encryption

2.2.3 Message Authentication Code (MAC)

The Message Authentication Code (MAC) is a symmetric key encryption algorithm used to provide message authentication. To build MAC protocol, the sending and receiving ends must exchange a symmetric key called k. A MAC is a checksum that is encrypted and computed on any message and delivered along with the same message.

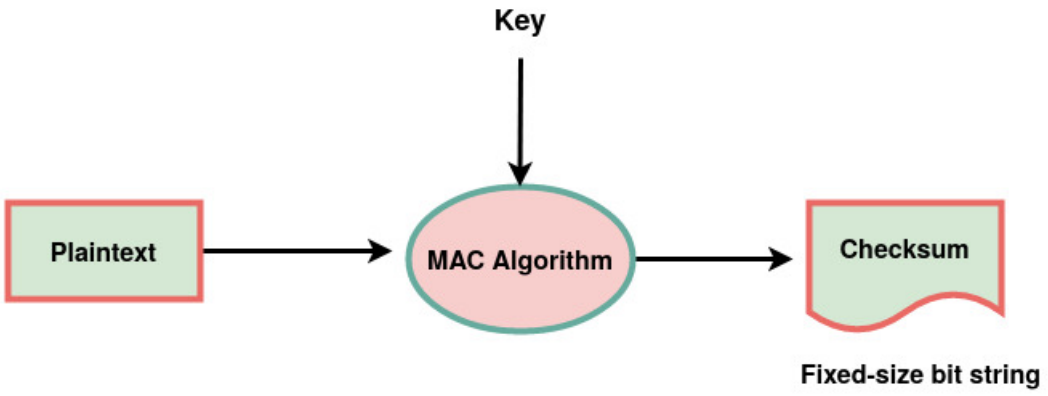


FIGURE 2.3: Message Authentication Codes

2.2.4 Digital Signatures

A digital signature is a method of associating an entity or a person to digital information. The digital information may be verified independently by the receiver or by any other person. A digital signature can be computed using message and a private key that is only known to the signer. In the actual world, the recipient of message needs to know that

it comes from the sender and that he won't be able to deny it. In business applications, where there is a high likelihood of data exchange dispute, this requirement is crucial.

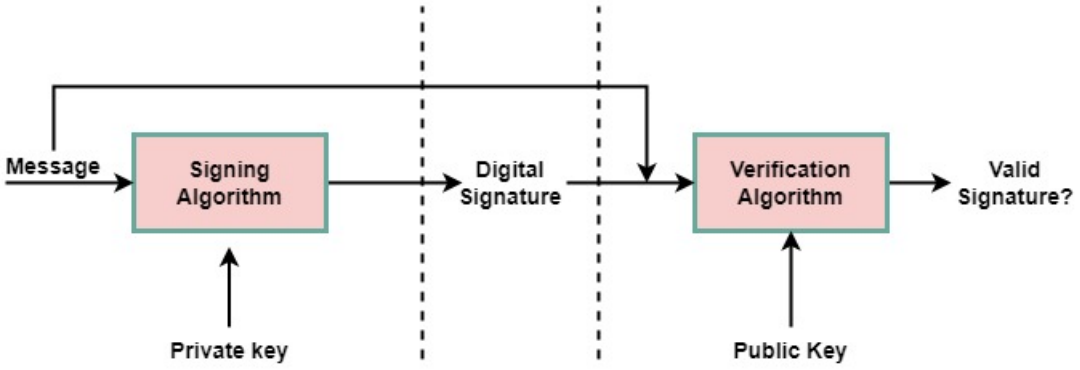


FIGURE 2.4: Digital signature

Blockchains rely heavily on digital signatures, which are generally used to authenticate transactions. As users submit transactions, they must prove to each system node that they have permission to utilize the funds while also preventing other users from doing the same.

2.2.5 Hash Functions

A cryptographic hash function changes input data of any length into output data of a fixed length. A hash function may be described mathematically as follows:

$$H : \{0,1\}^* \longrightarrow \{0,1\}^n$$

The collection of variable length binary elements, along with the empty string, is denoted by $\{0,1\}^*$, whereas the collection of binary elements of n length is denoted by $\{0,1\}^n$. As a result, hash function converts the collection of variable length binary elements into a set of binary elements with fixed lengths. Hash functions are used in designing digital signatures. The following are the desirable properties a cryptographic hash function should also satisfy.

- Collision Resistance

- Pre-image resistance

A hash is included in every block of blockchain. It can accept each block's hash as the block's unique id. This assists in the identification of a block and all of its contents, which are all as unique as a fingerprint. Changes within the block will cause the hash to change as the block is generated. Hence, hash value plays a prominent role in detecting changes to intersections. If the fingerprint of any blockchain changes, it is no longer the same block.

2.2.6 Merkle Hash Tree

A hash-based data structure known as a merkle tree is a generalized list of hashes. Each leaf node of merkle tree is hash of a data block, and every non-leaf node is hash of its children node. Merkle trees typically have a branching factor of two, indicating that each node can have up to two children. The merkle trees are used in distributed systems to verify data. As they use hashes rather than the whole files, they are efficient. The merkle trees are implemented as binary trees, as seen in the Figure 2.5. A merkle tree, on the other hand, can also be made as a n-ary tree, with n number of children per node.

Data verification is critical in a variety of peer-to-peer and distributed systems. Data verification guarantees that the data must be consistent throughout the network. However, verifying the complete file every time to validate data is computationally costly and time consuming. The merkle trees are used for this purpose to reduce the validating time. Basically, it is required to limit the amount of data being transferred across any network as much as possible. Instead of sending a complete file on the network, it is better to send a hash of the file to determine whether it matches or not.

2.2.7 Discrete Logarithm Problem

Discrete logarithm problem is one such problems, based on which many of the public-key cryptosystems are built. Let G be a multiplicative group of order m. Let g be a generator of G. Given any element $a \in G$ and $a = g^x$, finding $x < m$ is presumed to be difficult.

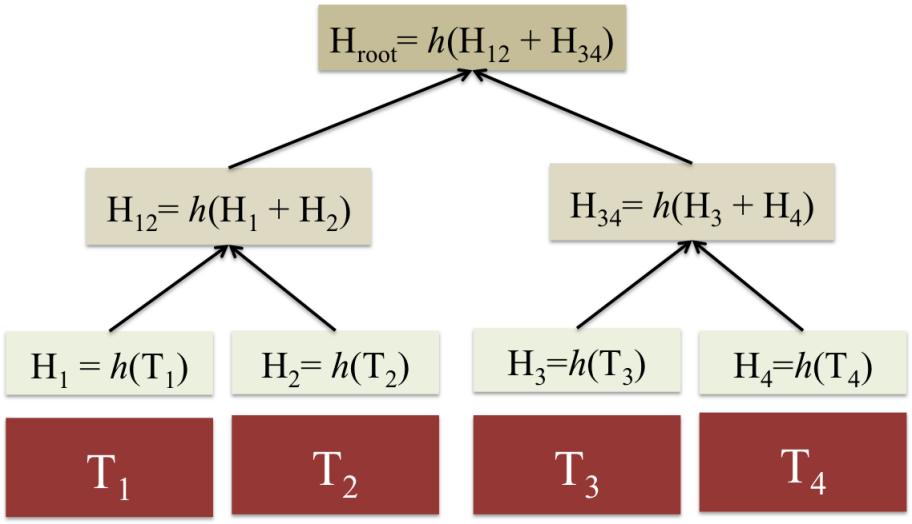


FIGURE 2.5: Merkle Tree [11]

2.3 Blockchain Basics

In this section, the details of architecture, functioning and components of blockchain including block structure, blockchain network and consensus used are explained.

2.3.1 Blockchain Technology

Blockchains are distributed ledgers that keep track of a growing collection of ordered records known as blocks. A distributed ledger functions as a type of replicated, distributed and synchronized database among its participants. Each block creates a ordered chain by including a timestamp, a list of transactions, a link to the previous block and a nonce. A transaction is defined as the transfer of a valuable asset from one owner to another. Each node utilizes a pair of public and private keys to broadcast transactions onto the blockchain. To begin, the node creates and signs a transaction, which is then broadcast to the blockchain network. Each blockchain node verifies all transactions before broadcasting them to its peers, discarding any that are invalid. To record these valid transactions, the network's miner nodes forms a new block and send it to their peers, who validate it before appending it to the ledger. This cycle continues forever. Each blockchain uses

a consensus method to resolve multiple states or forks in the network. A scenario of blockchain formation is depicted in fig. 2.6 as a series of blocks, each of which contains several transactions.

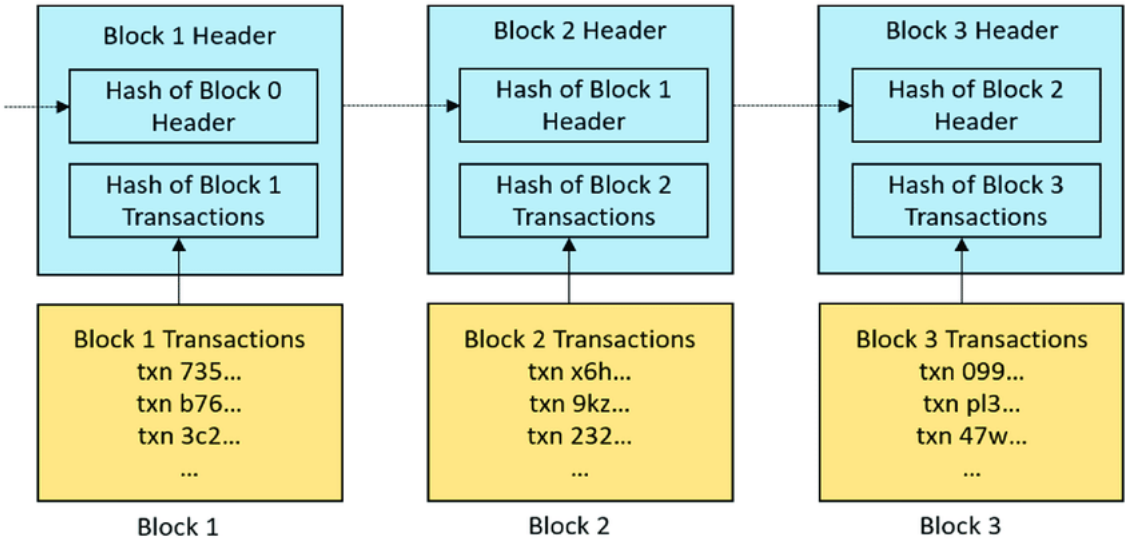


FIGURE 2.6: How blocks are chained to form a blockchain? [3]

2.3.2 Block Structure

A block is essentially a data structure which contains set of transactions that are to be added in the blockchain. Each block of blockchain contains list of transactions, block header and metadata. The fig. 2.7 represents the structure of a block.

2.3.2.1 Block Header

The block header of each block consists of: (a) previous block hash, the reference to a preceding block which links this block to the blockchain, (b) The mining parameters timestamp, difficulty, and nonce, (c) the merkle root that is used to effectively aggregate list of transactions in a block. The fig. 2.8 describes a block header structure.

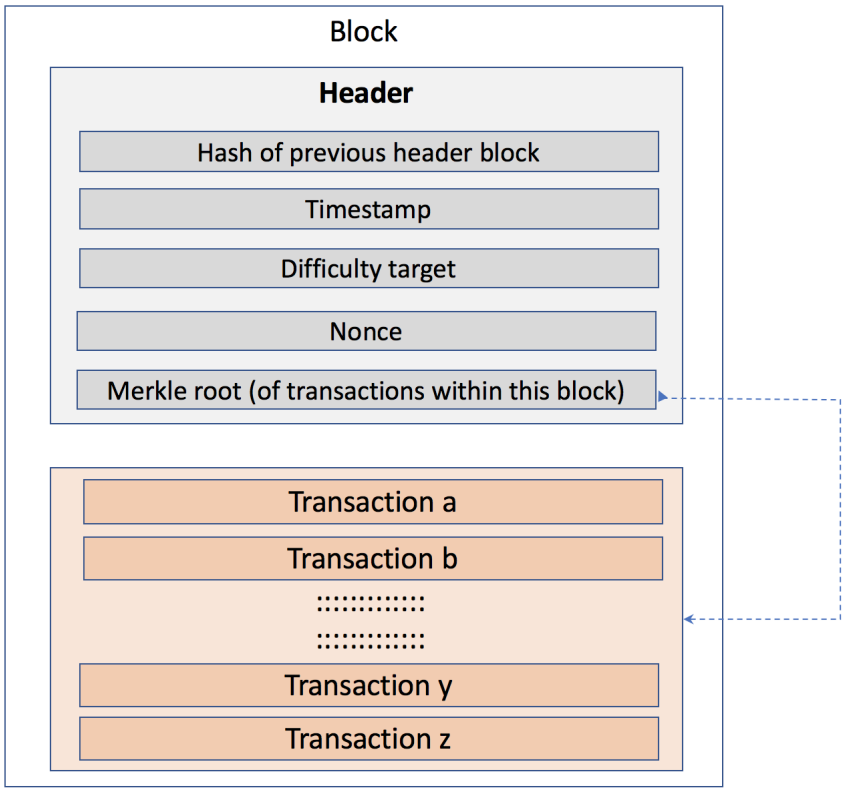


FIGURE 2.7: Structure of a block [116]

Size	Field	Description
4 Bytes	Version	A version number to track software/protocol upgrades
32 Bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 Bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 Bytes	Timestamp	The approximate creation time of this block
4 Bytes	Difficulty Target	The Proof-of-Work algorithm difficulty target for this block
4 Bytes	Nonce	A counter used for the Proof-of-Work algorithm

FIGURE 2.8: The structure of the block header [11]

2.3.2.2 Block Body

The actual data being added to the blockchain is included in the block body, commonly referred to as the "transaction data". Transactions are the actual data that is being recorded to the blockchain. These transactions include smart contract executions, cryptocurrency

transfers, and other types of data. All the transactions are aggregated in the form of a merke root and ot will be stored in the header of that block.

2.3.3 Peer-to-Peer Network

Blockchain is structured as a peer-to-peer network (P2P network). The P2P network refers to the fact that all nodes which are part of the network are equal, which means there are no special nodes, and the responsibility of delivering network services is shared among all the network nodes equally. There is no hierarchy within the network and there is no centralized server or service. The group of nodes using the blockchain P2P protocol is referred to as the blockchain network.

2.3.4 Nodes

A node is computer or device that participates in the blockchain network by maintaining a copy of the distributed ledger and verifying transactions. Nodes collaborate to preserve blockchain integrity by conversing each other to reach consensus. Further nodes can be classified as full nodes, mining nodes and lightweight node, each with different responsibilities and roles.

2.3.4.1 Full Node

The servers of a decentralized network are called full nodes. They validate new blocks, sync, store, copy, and disseminate transactions while preserving a transaction history. They keep monitoring to the rules, building a trustworthy database.

2.3.4.2 Mining Node

The mining nodes uses a proof-of-work consensus model, a validation technique that depends on arbitrary cryptographic problem to unlock tokens and add new blocks to a blockchain.

2.3.4.3 Lightweight Node

These nodes are intended to perform transactions and daily tasks quickly and simply. They don't download the entire blockchain; instead, they have only the needed information and rely on full nodes to run and verify transactions.

2.3.5 Consensus

Blockchain consensus is a mechanism through which network peers come to an agreement on current state of the chain. Consensus algorithms do this by establishing reliability and trust in the network. The verification of transactions will be done via consensus algorithm through which the security of the blockchain will be maintained. Some of the popular consensus mechanisms are proof of work (PoW), proof of stake (PoS), proof of capacity, proof of authority, delegated proof of stake etc.

2.3.6 Blockchain Benefits

The unique architecture of blockchain design is based on three key building blocks: (1) peer-to-peer network (2) cryptography and (3) consensus process. The blockchain gets lot of power from these three building blocks, allowing it to handle not only cryptocurrency but a lot of other applications. Some of the benefits of blockchain technology are listed below:

- **Decentralization**: The same copy of the ledger is replicated and disseminated by all the nodes, each node may independently validate the transactions without the need for any central authority.
- **Immutability**: The transactions can never be modified once they are recorded in the blockchain, making them irrefutable.
- **Transparency**: The distributed ledger keeps track of all the transactions. Because it is a public ledger, anybody can view and audit transactions. This provides asset provenance in various applications.

- **Chronological**: Blockchain utilizes cryptographic schemes such as hash functions which connects blocks in an ordered chain, which allows for a verification of the underlying transactions in a chronological fashion.
- **Consensus-driven**: Every blockchain has a consensus mechanism to assist the network in reaching quick decisions and transparently. The consensus mechanism ensures the system runs smoothly and helps the network nodes to quickly reach a consensus.

2.3.7 How does blockchain works

If client A want to transfer bitcoin to client B on the bitcoin network, client A will perform a bitcoin transaction. The transaction must be confirmed by the miners in order for the bitcoin network to commit it. To start the mining process, the transaction will be sent to all network nodes. To get approval, miners will group all the transactions into block, validate all the transactions of the block, and then broadcast the block and their verification to the network via a PoW consensus algorithm. The newly created block will be appended into the chain once verification of all the transactions is done by all other nodes. This entire process is depicted in Figure 2.9.

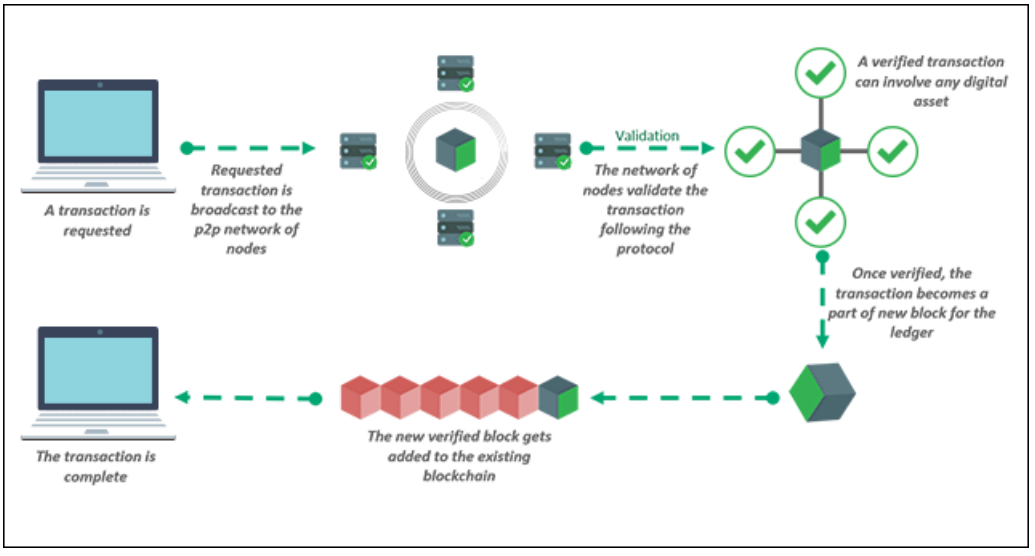


FIGURE 2.9: The process of appending new block [3]

The bitcoin transferred from client A to client B will become valid and complete iff the *block* having list of the transaction is confirmed by all network users and appended into theor distributed ledger. The Bitcoin blockchain uses a variety of security protocols, including merkle tree, hashchain, and digital signature along with the consensus algorithm, to prevent both the modification of every transaction record in a block after the block has been appended in the distributed ledger.

2.3.8 Types of Blockchains

Blockchain technology is capable of far more than just storing and managing digital currencies. In reality, there are several distributed ledger architectures, each with a unique level of centralization and access control to meet various business requirements. Based on different characteristics and their use, blockchains are classified into different categories.:

1. **Public or Permissionless Blockchain**: In public or permissionless blockchain, users can join the network without asking for permission. [29]. Participants read and write transactions, can take part in the consensus mechanism, and maintain copy of the ledger, making the ledger truly decentralized [164]. All participants may read, write, and validate new blocks, allowing them to keep a copy of whole blockchain [7]. In both formation and operation, public blockchains are secure. Though anyone can take part and broadcast transactions as blocks, every blocks must be validated by computationally expensive consensus protocol such as solving mathematical puzzles [7].

2. **Private or Permissioned Blockchain**: A private or permissioned blockchain is to be used only by a particular organization. In this network, users are invited to join and play a specialized role in the decentralized maintenance of the blockchain [29]. Only authorised entities are eligible to join the network and keep the blocks, which differentiates private blockchains from public blockchains [164]. Permissioned blockchains are more efficient than public blockchains since only known members join the network, and any tampering is identified by hashes and participant consensus in the same way that public blockchains are. Private blockchains, on the other hand, do not have anonymous nodes [164].

3. **Consortium Blockchain**: A consortium blockchain [172] security lets a pre-identified set of organizations to regulate the consensus process rather than letting any user inside the network to be part in it or giving a single entity to have complete authority.

	Public	**Private**	**Consortium**
Permission	Not Required	Required	Required
Access	Anyone	Single Organization	Multiple Organizations
Read	Anyone	Anyone	Anyone
Write	Anyone	Identified Nodes	Identified Nodes
Mining	Anyone	Identified Nodes	Identified Nodes
Example	Bitcoin, Ethereum	Hyperledger	Hyperledger, R3 Corda

TABLE 2.1: Classification of Blockchains

2.4 Privacy Properties of Blockchains

In this section, we present the privacy needs and dangers posed by the transactions, applications and network environment.

2.4.1 Privacy Requirements

The following requirements must be met by the blockchain in order to ensure privacy: (1) There should be no obvious or discoverable linkages between transactions. (2) Only the users are aware of the contents of transactions. Due to the public nature of blockchain transactions, everyone has unrestricted access to the network. The foundation for the privacy prerequisites is based on two things:

1. **Identity Privacy**: This includes the transactional connections between users as well as the transaction script's intractability with the true identities of their participants. Even if individuals participate in the blockchain using random addresses (or pseudonyms), they can only give minimal identity privacy. Some behavioural analysis tactics (e.g., know your customer (KYC) policy [65] or anti-money laundering legislation [142]) may provide some details on who is utilizing blockchain network and for what purpose by keeping track of the unencrypted network and traversing the open distributed ledger.

2. **Transaction Privacy**: The transaction details (e.g., transacting patterns or amount) of the public blockchain network is kept unknown to the public and may only be read by selected people. Many blockchain applications, such as big data based anonymous authentication and authorization, e-voting, e-auction and electronic health record administration require transaction privacy since users may prefer not to disclose critical information to any inquisitive blockchain entities and maintain their privacy.

2.4.2 Privacy Threats

As discussed earlier, a blockchain transaction includes a timestamp, the sender's signature, transaction's ID, the addresses of its users and trade values. Because of the open nature of the blockchain network, data mining techniques [131] may be used to extract individuals' physical identities or other information and track the flow of transactions.

2.5 Privacy Techniques used in Blockchains

This section explores several methods used for enhancing the privacy of blockchain systems that have been proposed in the literature:

2.5.1 Mixing Services

The Mixcoin protocol was first suggested by Chaum [36]. In this protocol, the network combines the transactions of a large number of users into a single transaction. Hence, attackers are unable to discover the mapping pair of each user in the input and output. As a result, its is almost impossible to find relation between the input and output. Consider the case where one person wishes to deliver a message 'M' to address A of another person. The person sending the message M will first encrypt it using the receiver's public key K_A, then includes A and finally encrypt the outcome with the public key K_I of the intermediary. The ciphertext, which is given to an intermediary, is denoted as L.H.S. of the following expression:

$$K_I(n_0, K_A(n_1, M), A) \longrightarrow K_A(n_1, M), A$$

The intermediary has transformed another ciphertext which is given in the above expression. The intermediary uses his private key to decrypt the original ciphertext in this process. The intermediary then sends the sub-ciphertext to A, who uses his own private key to decrypt it. It is worth noting that n_0 and n_1 are both random numbers, ensuring that no message is sent twice. In this technique, the relationship between source and destination of each message will be hidden when intermediary receives so many number of inputs and outputs. The sequence of arrival is concealed by dispersing uniformly sized pieces in random ways. A mix cascade may also be made by combining many intermediates, which reduces the risk of a single intermediary being the attacker.

During the past several years, a number of mixing services were created and utilized in the blockchain networks to obscure history of transactions and reduce the danger of deanonymization. Gregory Maxwell [106] created CoinSwap, a bitcoin mixing method based on third parties. The process of this protocol is that a mixer acts as an intermediate between a large number of senders and a large number of receivers. All transactions between sender and mixer, as well as the mixer and receiver, are escrow transactions that are hash-locked and can only be spent by redeeming transactions. The assets of the users cannot be stolen due to this lock mechanism. However, because the transactions are delivered in plaintext, the mixer can still track of all the transaction pairs and their metadata. In 2014, Bonneau et al. introduced Mixcoin [26], which uses a signature-based accountability system to reveal theft so that users can unambiguously verify whether the mixer has misbehaved. The article [157], which is based on the bitcoin mixing protocol Mixcoin, uses a blind signature mechanism and optimises the public log to hide the input and output addresses of any user in the mixing server. The third-party mixer, who must be trusted can be malicious. Coinjoin [105] is a mixing mechanism that removes the need for a third party. The Coinjoin has the characteristic that a single transaction can have multiple inputs and outputs. The link between inputs and outputs is mixed in the joint transaction such that the exact path of data flow is unknown to the other peers.

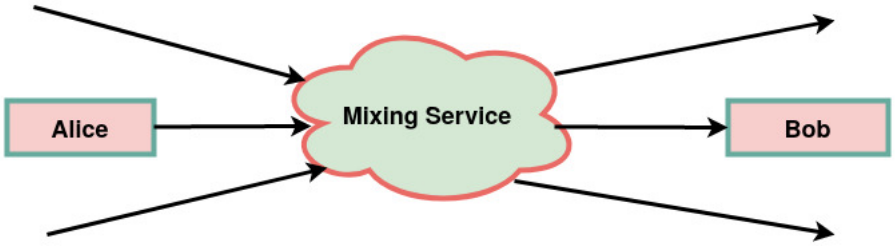


FIGURE 2.10: Mixing Service [129]

The mixing service is further classified into two types:

1. **Centralized Mixing:** Centralized mixers [180] accept user coin and send back different coins for a fee. The user must input his address in a form on the mixing service platform in order to start a transaction. Next user transmits his coin to a location the firm has provided.

2. **Decentralized Mixing:** These mixers [136] make an effort to improve upon the drawbacks of centralized mixing. People pool their coins in order to make a single large transaction, and then the coins are randomly returned to the pool participants. The pool's randomness increases with the number of users.

2.5.2 Homomorphic Encryption (HE)

The homomorphic encryption (HE) is extremely effective cryptographic technique [59] that is used to perform arithmetic operation on encrypted text directly and ensure that the operations on the ciphertext provide the same output as operations performed on the plaintext when decrypting the results. Consider the below scenario: A possesses the secret values $\{x_1, x_2, ..., x_n\}$, whereas B has the function $f(.)$. A and B wish to work together to solve $f(x_1, x_2, ..., x_n)$ without revealing secret values or method specifics. The HE system is defined as a collection of $E(.)/D(.)$ functions. A can transmit encrypted inputs $\{E(x_1), E(x_2), ..., E(x_n)\}$ to B, who then performs computations on the ciphertext, randomizes the result, and sends it back to A. Now, A will safely learn $f(x_1, x_2, ..., x_n)$ after decryption. In general, HE works like a black box, which returns the ciphertext of the identical operations on the matching plaintext when n ciphertexts and operations.

Because of this appealing property, HE is highly useful for concealing and making timely updates on the transactions, the amount and metadata. The Paillier cryptosystem [123] and the Pedersen commitment scheme [124] are good instances of HE systems which can be used to protect blockchain privacy.

- One of the popular implementations of HE scheme is the Pedersen commitment scheme. [124] . It allows the homomorphic operations on commitments directly and can hide the true message completely with a trapdoor.
- The Paillier cryptosystem [123] is an additive HE scheme, which works based on the problem of composite residuosity class. This means, using same public keys and with knowledge of ciphertexts m1 and m2 anyone can compute the ciphertext of (m1+m2).

Without affecting any functionality of the network, homomorphic encryption schemes may be utilized to store data on the blockchain [145]. As a result, the privacy issues associated with public blockchains are minimized because it is guarantees encryption of the data on blockchain. Homomorphic encryption ensures privacy while making encrypted data on the public blockchain easily accessible for audits and other purposes, such as tracking users costs. [147]. Ethereum smart contracts encrypt data stored on the blockchain using homomorphic encryption for privacy and greater control.

2.5.3 Attribute-Based Encryption (ABE)

In ABE, the ciphertext is encrypted using private key of the user and regulated using attributes. The encrypted data may be decoded using private key of the user if the attributes of user matches those of the ciphertext. A critical security feature of ABE is collusion-resistance. The idea of ABE [172] was first introduced with single authority in 2005. Since then, several improvements to the fundamental ABE have been recommended, such as the use of multiple authorities in a single ABE to jointly create users' private keys [34, 78, 93] and ABE schemes that support arbitrary predicates [64, 66]. Despite the fact that attribute-based encryption is quite successful, it hasn't been employed in many applications since it's difficult to implement well and the fundamental concepts aren't well

understood. A decentralized ABE [93] was proposed for using ABE in blockchains in 2011 but so far, ABE has not been utilized practically on any type of blockchains.

2.5.4 Secure Multi-Party Computation (SMPC)

SMPC [53] is a well-known cryptographic technique that is specifically designed to handle the cooperative computation problems for ensuring privacy when several parties do not trust each other. The SMPC is utilized in secure e-voting [113] and the well-known millionaire problem (MP) [75, 167], and their applications. SMPC can be represented as n participants, $P_1, P_2, ..., P_n$ participating in a collaboratively computing task where participant P_i knows only his own input x_i and participant P_i learns only his own output y_i. SMPC can be mathematically represented as:

$$f(x_1, x_2, ..., x_n) = (y_1, y_2, ..., y_n)$$

The below are two security properties for SMPC:

1. **Privacy**: Participant P_j cannot acquire any other input x_i $(j \neq i)$.
2. **Consistency**: All honest users can finally acquire same output $y_1 = y_2 = ... = y_n$.

SMPC provides user privacy, computation accuracy, and decentralization features, allowing data privacy and safe usage. It performs a distinct function in smart contracts, random number generation, key management and other technologies. By incorporating SMPC technology, blockchain can increase its data confidentiality and adapt to more application scenarios. With the use of blockchain technology, SMPC may accomplish redundant computing and can gain verified qualities. They are complementary to one another and work together for achieving the goal of privacy preservation. Naturally, in practical application there is a bottleneck owing to the difficulties and poor efficiency of SMPC technology, and its performance must be improved in the future [159].

2.5.5 Non-interactive zeroknowledge proof (NIZK)

The ZKP is a cryptographic scheme used for proving a given statement without revealing any other information. The NIZK is a type of ZKP in which both prover and verifier do not interact with each other, making it appropriate for use in blockchain fore verifying messages anonymously in a distributed manner. Blum et al. [23] developed the notion of NIZK proof. The NIZK proof is highly suited for developing privacy-preserving protocols due to its property of independently proving the correctness of a statement without revealing additional data. The NIZK proof system is defined as follows: Consider a PPT (probabilistic polynomial time) algorithms pair (P, V) where P is prover and V is a verifier. If (P, V) satisfies the following conditions for language $L \subseteq NP$ (is a class of languages)and a security parameter k, it is known as the NIZK proof system for L. Completeness, soundness and zero knowledge are thee major characteristics of NIZK.

To avoid transaction graph analysis, Zerocoin [110] uses the cryptographic technique NIZK proof, which has the properties of completeness, soundness, and zero knowledge. The primary concepts underlying this proposal are similar to decentralized mixing, in which a coin is first minted and then returned with completely new coin with no historical information. The validity of the created currency is verified using NIZK proof, and an equal-priced coin will be returned. Despite the fact that Zerocoin guarantees excellent anonymity, there are still limitations that must be addressed [56]. Zerocoin's privacy-enhancing features need a substantial amount of computation power, users with restricted resources may find it difficult to adopt. In this, transactions are larger than traditional transactions, which increases the cost of processing and storing them on the blockchain. To address these limitations, Zerocash [140] is proposed by offering identity and transaction privacy at the same time, resulting in a better privacy protection for blockchain.

2.5.6 Trusted Execution Environment (TEE)

Global Platform (GP) developed the concept of a trusted execution environment (TEE) in 2010. It can provide a confidential and secure space for critical information and processing in an untrustworthy environment while also separating the environment in which

software executes [176]. TEE can give solutions with high-level performance for difficult cryptographic problems like SMPC that blockchain cannot. For example, IntelSGX [45] is a multi-party computing solution that employs hardware to construct a black box with trust. In TEE, sensitive information can be protected from malicious parties using isolation property. It can actively protect from external security issues and ensure the integrity and security of data more effectively than traditional security systems. Currently, the expense of the hardware platform and the difficulty of developing trustworthy applications are limiting the advancement of this technology [70].

2.5.7 Hidden Address

The hidden address approach is commonly employed in digital money to resolve the issue of input-output address correlation.[42]. Because it is possible to track an address on the blockchain, numerous addresses must be used to create confusion. When the sender makes a transaction, it first creates a temporary intermediate address with the help of the recipient's public key using the elliptic curve encryption technique, and then uses that address to spend the coin. Lastly, the receiver calculates the transaction by using its own public key in order to spend it. It is impossible to determine actual users intermediary address because of the random uncertainty. The addresses of both parties of a transaction are hidden from other users or attackers on the blockchain, protecting user privacy and the security of the coin. However, there are several drawbacks of using a hidden address. The privacy of transactions flow can be broken by analyzing the relationship between sender and receiver using transaction graph [131]. As a result, the hidden address technique must be further improved.

2.5.8 Differential Privacy

Cynthia Dwork developed the concept called differential privacy which is a mathematical proof that gurantees individual privacy [54]. It guarantees that the attacker will not learn any new information about an individual. The databases D1 and D2 differ in only one record, with almost same output probability. Because the attacker cannot distinguish whether it is originating from $D1$ or $D2$, the privacy is protected, and he can not determine

anything new using auxiliary information. If for all $D1$ and $D2$ differing with almost one element and all $S \subseteq Range(M)$ the randomized algorithm M gives $\in$-differential privacy.

$$Pr[M(D1) \subseteq S] \leq exp(\in).Pr[M(D2) \subseteq S],$$

where $S$is the the probability space and privacy budget is over the coin flips of the mechanism M. Differential privacy has been used on a variety of applications and data sets. It has been used to protect privacy during location pattern mining. Shen et al. [73] proposed an algorithm for location pattern mining data sets that enables differentiated privacy in the paper. They divide the required amount of differential privacy into multiple levels to create realistic differential privacy.

In blockchains, differential privacy is used to access private databases using queries which aggregates data, as well as to get user data with some statistical variances from different sources while ensuring the users' desired privacy settings. The first scenario is more relevant to permissioned blockchains enable third parties to access their anonymized information. The second scenario is relevant to blockchains which are used for sensor data collection, where the entire ledger can be analyzed statistically, but has statistically shifted data in a single transaction. Differential privacy is used to prevent an adversary inferring sensitive personal data in federated learning and record crowd-sourcing operations. Gai et al. [61] proposed a data-sharing strategy using blockchain that allows owners of data to control privacy - preserving processes and avoid data mining-based attacks on blocks.

2.6 Digital Signature Schemes in Blockchain

In this section, digital signature scheme and its variations, anonymous signatures schemes and literature survey on digital signatures schemes used in blockchains are carried out.

2.6.1 Digital Signature

Modern technology has made personal information more readily available to the public, and as a result, privacy is gaining importance as a security feature. Anonymity and unlinkability are the two key concepts that define privacy [12]. Anonymity refers to the concealment of a user's identity or personally identifying information in authentication communications. An unauthorised party cannot determine whether two authentication messages were produced by the same user or not. In general, users prefer to protect their privacy when using a service, but the service provider may need to compromise that policy to gather enough user data. Many different versions of the digital signature schemes were proposed in the literature.

2.6.1.1 RSA Signature Scheme

The RSA signature scheme is used for creating and verifying digital signatures with the help of the RSA algorithm. A pair of private and a public keys are used to sign messages and to confirm their authenticity. The sender uses a mathematical function to sign the message using his private key, which generates a digital signature. The recipient can then verify authenticity of the message by using public key provided by the sender by applying a matching function on the signature. Similar to the RSA encryption technique, security of the RSA signature scheme relies on the difficulty of factoring large integers.

2.6.1.2 ECDSA

The ECDSA is a public-key cryptographic algorithm used to crate secure digital signatures. It is frequently used in applications like authentication, digital certificates and secure communication and is based on elliptic curve cryptography. A public and private key are used by ECDSA to verify and sign the messages. ECDSA's security relies on the difficulty of elliptic curve discrete logarithm problem. ECDSA have various advantages over other signature schemes, including faster computation, smaller key sizes, and improved security.

2.6.2 Anonymous Signatures

Some signature schemes themselves are capable of giving the signer anonymity. These kind of signature schemes are referred to as anonymous signatures. The group signature and ring signature are two commonly used anonymous signatures. Currently these two signature schemes are being used to protect user identity privacy in blockchains.

2.6.2.1 Group Signatures

A cryptographic technique called group signature was first proposed in 1991 by Chaum et al.[35]. Any group member can use their secret key for signing a message on behalf of the entire group anonymously. Any group member with the help of group public key can verify the signature and confirms that one of the group members produced the signature. The signature verification process reveals nothing about the real identity except the group membership of the signer. A group manager controls the process of adding members to the group, resolving disputes, and disclosing the original signer.

A tuple of four probabilistic polynomial-time (PPT) algorithms makes up a group signature scheme $GS = (KeyGen, Sign, Verify, Open)$:

1. **KeyGen**: $KeyGen(.)$ algorithm takes input 1^{λ} and 1^{n}, where parameter λ and $n \in N$ is the group size, $k \in N$ is the security and gives a tuple $(gpk, gmsk, gsk)$ where $gmsk$ is the group manager's secret key, gpk is the group public key and gsk is an n-vector of keys with a secret signing key $gsk[i]$ for player $i \in [n]$.

2. **Sign**: $Sign(.)$ algorithm takes a message m and a secret signing key $gsk[i]$ as an input and gives a signature on m with $gsk[i](i \in [n])$.

3. **Verify**: $Verify(.)$ algorithm takes a message m, a candidate signature σ for m and group public key gpk as an input the and gives either 1 or 0 as an output.

4. **Open**: $Open(.)$ algorithm takes a message m, a signature σ of m and the group manager secret key $gmsk$ as input and gives an identity i or the symbol $\perp$ to indicate failure as an output.

In blockchains systems as nodes leave and join the network dynamically, we require an authority to join and revoke groups, as well as mechanisms to revoke membership of specific participants and to add new members to the group as nodes leave and join the network dynamically. The group signature is suitable for private and consortium blockchain systems since a group manager is required to set up the group. To the best of our knowledge,PlatON [127] is the only one that utilized group signature scheme in their platform to provide users anonymity.

2.6.2.2 Ring Signatures

The ring signature is a type of group signature, more significant in terms of privacy than a normal group signature because of its unconditional anonymity and unforgeability. Rivest et al. [134] proposed a digital signature called the ring signature which enables to create an anonymous and valid signature from a set of possible users without revealing their real identity. Consider a user A_s choosing a group of participants along with him and forming a ring $\{A_0, A_1, ..., A_n\}$. All the participants of the scheme should have their public keys from signature schemes like $RSA, ECDSA$ etc. The user A_s signs any message with his secret key (S_{As}) and public keys $(P_{A0}, ..., P_{As}, ..., P_{An})$ of all other members. Then, the verifier knows that the message was signed by one of the member from the ring, but doesn't have an idea on who signed it. As a result, the identity of signer remains completely anonymous using this signature.

To achieve properties like anonymity and unlinkability, various ring-based privacy preservation techniques for blockchain have been developed [119, 149]. Nicolas van Saberhagen first described the utilization of ring signatures for concealing the origin of transactions in their CryptoNote protocol, which was first released in 2012 and modified in 2013 [120]. The proposed approach employs traceable ring signatures, allowing a member to sign one legitimate transaction with a single secret key. A "key image" (hash of signer's private key) will be appended along with the transaction in the CryptoNote. It ensures that each transaction input may only be spent once in order to protect against double-spending attacks. Nodes will reject any of the new transactions which has utilized the same "key

image" which is already utilized. Furthermore, based on the transaction amount, the analysis attack can be possible on the CryptoNote protocol [168]. RingCT (Ring Confidential Transactions), proposed by Noether et al. [119], is a modified CryptoNote version. The modified CryptoNote uses ring signature with larger size to enhance security and privacy. It uses multi signature scheme based transactions and a new proof of work algorithm for faster transaction processing. By using confidential transactions of Greg Maxwell, the suggested approach also conceals the amount being sent. RingCT depends on the sender's capacity to trace the keys which are containing same amount to form necessary set of keys for creating ring signature to be successful in hiding source of a transaction. The larger the group, the better is the scheme; smaller groups could be traced using transaction graph analysis.

2.6.3 Literature Survey on Signature Schemes in Blockchain

In this section, we'll carefully go through a few cutting-edge digital signature technologies used in blockchain. In 2016, an Interactive Incontestable Signature (IIS) was proposed by Zhu et al. [179] that prevents the falsification of owner information and ensure the authenticity of trader information. The major goal of this signature schemes in blockchain is to provide a tamper-proof and secure way of transaction verification and ensures blockchain integrity. The IIS could guarantee that the transaction is validated by the dealer and is irrefutable. By using this signature, the merchant could guarantee the owner that the transaction will be incorporated in the blockchain in an undeniable way. The merchant's non-repudiation and the owner's unforgeability were both secured by the signature scheme. The problem of quick nonrepudiation confirmation during blockchain deployment was resolved by this scheme.

Sato et al. [141] stated that if the underlying cryptographic algorithms are compromised then the security of blockchain data would be compromised. If SHA256 is compromised then that may lead to double spending attacks. It results in a denial of payment when RIPEMD160 compromises. From an ECDSA breach, false alarm claims and currency theft would come. Combining the hash function with the digital signature causes compromise and led to refused transactions, double payments currency theft, and modification to

transaction data. They suggested that, instead of using trusted third party mechanism, it is better to utilize a long signature scheme to address the security issues in the distributed ledger.

Without depending on trusted third parties, Aitzhan et al. [4] found a way to address the problem of transaction security in decentralized smart grid energy trading. The decentralized energy trading system for validation uses blockchain technology, multiple signatures, and information flows that were encrypted anonymously. Peers were able to conduct trades safely while secretly negotiating energy pricing. Transactions were secured by the system using P2P community data replication mechanisms as they were duplicated across all active nodes. Additionally, the system was able to defend against any double spend attacks and resolve the byzantine general problem that occurred in online payment systems due to the usage of workload proofs.

Yuan et al. [169] gave an aggregate signature system that can be used in blockchain big data transactions. The scheme was used for the transactions with several inputs and outputs whose value need to be concealed. Regardless of how many inputs and outputs were included in the transaction, the size of the signature always remained constant, thereby increasing performance of the signature.

Shen et al. proposed a method for hiding the transaction value in Monero. Similar to Monero [119], bitcoin was assigned using a workload proof "dig" procedure without the establishment of a trusted party. The Crypto Note system, on which the original Monero protocol was built, uses one-time keys and ring signatures to hide the nature and origin of transactions. Gregory Maxwell, a developer of the Bitcoin core, had proposed methods employing a commitment mechanism to conceal transaction values.

In addition to introducing the ECDSA's uses in blockchain, Benjamin [80] demonstrated how ECDSA was applied to bitcoin blockchain. The emphasis on the elliptic curve theory and its use in cryptography is huge area and it is proven that the security of an elliptic curve-based cryptosystems is based on the difficulty of elliptic curve based DLP. The complexity of this problem also made sure that the blockchain transactions made on the Bitcoin network were secure and authentic.

ShenTu et al. [144] given a scheme to enhance bitcoin blockchain privacy and prevent the centralized currency mixing from combining bits with numerous inputs and outputs in order to expose their link with one another. This scheme was based on the elliptic curve based blind signature system. Compared to RSA Coin-Mixing, their blind signature system based on blind-mixing was 10.5 times quicker, and it can resist against super attackers.

Mercer et al. [109] proposed a unique ring signature scheme (URS) for the existing blockchain networks. With an emphasis on integrating privacy on the blockchain, to fast create ethereum smart contracts, he offered the first prototype that was compliant with the blockchain library. The solution's security and privacy features have been shown, and researchers have compared it with other frequently used blockchain privacy solutions to demonstrate how successful it is. On ethereum blockchain, the URS is an expensive one to implement.

Current ECC blind signatures are incompatible with the ECDSA and hence cannot be used directly in bitcoin transactions, claim Andreev et al. [8]. He proposed a scheme that enables the creation of a blind signature that works for the current bitcoin blockchains. Here, signatories might offer services for the storage of private keys and transaction authentication without being aware of the amount being transferred. The program might secretly lock some amount with the help of participants when used with multi-signature transactions. Secret parameters must never be used more than once in a single signature, as in standard ECDSA.

Dikshit et al. [49] said that all transactions of bitcoin will be recorded and kept in a blockchain. Bitcoin has to be kept in a secure wallet as these transactions were open to everyone. Due to the irreversibility of bitcoin transactions, the bitcoin wallets that are generally accessed with a key, becomes impossible to access if the key is lost. Previous researchers suggested several methods to address this issue, but these methods are having drawback of processing and managing the keys. Their proposed a scheme addresses this problem and allows all participants to acquire a single share and fulfill the concept requirements.

Cruz et al. [43] researched several schemes for an efficient and secure e-voting system, but these were challenging to utilize when casting real ballots. This regrettable situation is caused by a number of technological factors, such as anonymous communication channel which is challenging to deploy on the Internet. In the proposed scheme, other well-known protocols (like digital signature) was combined with bitcoin protocol to provide safe, transparent and anonymous e-voting system and several key aspects of the e-voting system were discussed, along with verifiability, soundness, anonymity and fairness. It was shown that using the bitcoin protocol gave other benefits like immutability and irriversability in addition to communication anonymity.

In order to overcome the drawbacks of using public-key certificates, Lin et al. [99] proposed a novel ID-based linear homomorphic signature scheme, which is used in identity-based cryptosystems. The suggested approach was shown to be secure against existential forgery on adaptively chosen messages and ID attacks under the random oracle model. ID-based linearly homomorphic signature systems may be employed in e-commerce applications.

Guo et al. [71] presented an attribute-based signature method for e-Health to confirm the validity of Electronic Health Records (EHRs) stored on blockchain. A patient should endorse a statement in accordance with an attribute in the proposed scheme, with the patient disclosing nothing further but the proof that he had attested to it. Furthermore, this scheme is used by many authorities to generate and disseminate public/private keys rather than a single, trusted authority. It was carried out to solve the escrow issue and complies with blockchain's distributed data storage paradigm. This protocol prevents collusion attacks by distributing the secret pseudo random function seeds among the authorities. The complete privacy and unforgeability of the signer and attribute was formally shown on the premise of the computational bilinear Diffie-Hellman.

In order to protect the blockchain network over the already employed conventional channels, Li et al. [94] suggested a new signature scheme based on lattices. They combined the algorithms ExtBasis and RandBasis in the key generation phase to create the sub-private keys needed to validate the data. This randomises the ExtBasis algorithm's output and enhance the security of user information. The comparative analysis reveals that the strategy was more effective than comparable studies and secure in a random oracle against

an attack using a message that was adaptively selected. This scheme therefore was more appropriate for the Post-Quantum Blockchain Network transaction implementation. Ren et al. [132] proposed a compact NIZK protocol and used it to strengthen a ring signature to address the need for lower bandwidth cost in distributed ledger. Their scheme required less storage space for computations related to signatures and pairings throughout the verification phase.

In Table 2.2, we compare the signature schemes processing method and their usage in various applications domains.

Scheme	Methods	Usage	Application Field
Aitzhan et al. [4]	Aggregate signature	To solve the general byzantine problem and protect any electronic payment system from double payment attacks.	Providing transaction security in decentralized smart grid.
Andreev [8]	Blind signature	The signer has anonymity and can hold private keys. Provides transactions authentication without being aware of the amount being transferred.	Bitcoin
Cruz and Kaji [43]	Blind signature	Protects privacy of the voters.	Blockchain-based e-voting system.
Dikshit and Singh [49]	ECDSA and threshold signature	Possibility of greatly enhancing Bitcoin security.	Bitcoin
Guo et al. [71]	Group signature	To prevent collusion attack by distributing the secret pseudorandom function seeds.	Blockchain-based EHRs
Benjamin [80]	ECDSA	The discrete logarithm issue of the elliptic curve's computational difficulties is the source of the security of the elliptic curve-based cryptosystem.	Bitcoin
Li et al. [94]	Aggregate signature	Secure against the adaptively chosen message attack in the random oracle.	Utilised to strengthen the security of the blockchain network over existing traditional routes.
Lin et al. [99]	Proxy signature	Protection from message Identity theft and existential theft on adaptively chosen message.	e-business and cloud computing.
Mercer [109]	Ring signature	Integrity, anonymity and unforgeability.	Smart contracts
Shen et al. [119]	Ring signature	To ensure privacy, hides the total number of transactions between the sender and the recipient.	Monero
Ren et al. [132]	Ring signature	Anonymity and untraceability.	In blockchain with low bandwidth.
Sato et al. [141]	Proxy signature	This method can prevent the hard fork and key pair change when the signature scheme is compromised.	When the digital signature and hash function of the underlying encryption scheme are compromised.
ShenTu et al. [144]	Blind signature	Defends against super attackers.	For transactions mixing in blockchain.
Yuan et al. [169]	Aggregate signature	The possible forgeries of the attacker cannot be realised, and the security efficacy of the aggregate signature is nearly same.	To conceal transaction details.

TABLE 2.2: Comparison of Various Digital Signature Schemes used in Blockchains

2.7 Summary

In this chapter, a detailed background of all security properties, cryptographic primitives and blockchain terminologies that are utilized throughout the thesis are given. In this chapter, we first analyzed the blockchain privacy threats and requirements. We then analyzed different blockchain privacy protection schemes in detail. We then discussed different signature schemes used in blockchains to protect privacy of the signer. After the comparison of the schemes and review, we found that although group signature schemes are available for identity privacy protection, they are not compatible for decentralized applications because of centralized group manager. There is a dire need to decentralize the group manager. Also, despite various ring signature schemes available in literature, the same cannot be implemented for certain blockchain applications like e-auction that may need user identity at the end. These open challenges are addressed by our proposed schemes in the next chapters.

Chapter 3

Identity Verifiable Ring Signature Scheme for Privacy Protection in Permissionless Blockchains

In the previous chapter, detailed survey on blockchain primitives, privacy properties and privacy protection schemes are carried out. In blockchain-based e-auction applications, the users don't want to reveal their monetary values and identity information to other users in the network. Any curious node or malicious node in the network can analyze all the transactions of any user and know their real identity as all the transactions within network are public to everyone. In blockchain applications, there is no privacy for the identity of user. To address the user identity privacy issue in e-auction like applications which are deployed on public blockchains, an identity verifiable ring signature scheme (IVRSS) is proposed in this chapter.

The remaining chapter is organized as follows. The first section describes about problem statement. In second section, the preliminaries required for this chapter are discussed. The third section describes about the formal model and security definitions of the proposed scheme. In fourth section, an identity verifiable ring signature scheme is proposed. Correctness and security analysis of the proposed identity verifiable ring signature scheme is done in fifth section, followed by conclusion in sixth section.

3.1 Problem Statement

The decentralized public ledger technology in peer-to-peer network [96, 84] is becoming popular now-a-days. The technology has received considerable attention recently with the continuous development in financial and non-financial domains [57] and also because of its security features. This led to a flurry of advancements in various applications using blockchains. Blockchain offers various security features such as transparency, immutability, traceability in business transactions [5] as discussed in the previous chapter. Although all blockchain systems possess these security features, few of business applications like e-auction, crowd funding, healthcare etc. emphasize on users privacy.

Transactions play a vital role in any blockchain system [174]. Transactions are first created by various users and are broadcast on the network, which are then validated by the network nodes. Then all such validated transactions form a block and are added into the blockchain. Transaction data structure can encode the transfer of asset from one party to other party in the system and every transaction is a public entry [10] in publicly available global ledger blockchain. In order to transact on blockchain, user requires a public, private key pair where public key is used for account identification and private key to sign the transaction. Since every transaction in the blockchain network is publicly available, anyone in the network can inspect and analyse them.

Let us consider an application called e-auction where identity privacy is a critical issue. The most well-known variant is English auction, in which the buyer who offers the highest price will win the auction. A buyer who wants to purchase goods or services from sellers need to submit bids. The buyers want at the lowest price, where as each seller hopes to get competitive prices from the buyers. To facilitate this mechanism a trusted third party is required to host the auction such that the privacy of the participants and the fairness in exchange is achieved. But the trusted third party holds a lot of important information about the users. This may lead to potential threats [77] from single-point attacks to collusion attacks all the time and it is also difficult to find a fully trusted third party to play such a role in reality.

Recently, many auction protocols [38] were deployed on top of blockchain to take advantage of the decentralization, transparency, immutability and verifiability properties of the blockchain, and to get rid of the above shortcomings that were brought by the third party. But as every transaction in the blockchain network is publicly available, transaction analysis [131] can reveal the original identity of the user and their monetary values. Blockchain can be very much regarded as a trusted party for correctness and availability but not for privacy. Considering public nature of the transactions in the network and the privacy challenge of identity of user in blockchains, a crypto primitive is required to address it, and one such primitive available in literature is signature schemes.

Digital signature is a cryptographic technique used to verify the authenticity of the message and identity of the sender. A valid signature ensures integrity of the message as well as non-repudiation of the message. Rivest et al. [134] proposed ring signature that is setup-free and unconditionally anonymous. The ring signature scheme will be having only ring members and there will be no managers to supervise the members. The signer of this scheme chooses public keys of other members of the ring at random, combines their public keys, his private key and uses random numbers, and utilizes additional technologies to complete the ring signature. The signature verifier cannot determine who signed the message but can only confirm that it is signed by one of the ring member. Therefore, ring signature is very suitable for user identity privacy protection in public blockchains as there is no centralized party to control the network.

Rivest et al. [134] introduced ring signatures in 2001, which rely on the trapdoor permutation' existence to ensure user anonymity. Following that, Bresson et al. [28] demonstrated that ring signature of Rivest may be used with lower security premises and enhanced the signatures to a threshold ring signature to address the security issue. At the same time, Abe et al. [1] proposed a 1-out-of-n signature technique with low storage and computation costs, while Zhang and Kim [170] presented ring signatures from pairings and ID-based blind signatures. Moni Noar [115], on the other hand, published a deniable ring authentication in 2002, which combines Riverst et al. [134]'s ring signature with Dwork et al.'s [55] deniable authentication. A zero-knowledge authentication proof mechanism is provided by the deniable authentication. Xu et al. [165] presented a bilinear pairs ring signature system in 2004. Bender et al. [19] proposed substantially better security principles for

ring signatures in 2006, arguing that earlier ring signatures are vulnerable to chosen public key attacks. Boneh et al. [24] then suggested a secure and efficient ring signature system in the random oracle scenario. Shacham et al. [143] provided more convincing security definitions for unforgeability and anonymity than Bender's solution. While one stream of work addressed ring signature security issues, another focused on ring signature efficiency [52, 69]. A sub-linear-sized ring signature scheme was given by Chandran et al. [32] without random oracles in 2007 and demonstrated several drawbacks in terms of signature size. A traceable constant-sized ring signature scheme was given by Ke Gu et al. [69]. Furthermore, various certificate-less ring signature systems [69, 39, 171, 33] addressed the ring signature key-escrow problem.

Although in ring signatures permanent anonymity seems to benefit users, absolute anonymity causes serious challenges in real-world applications where knowledge about identity of the user is very essential. As a result, ring signatures' full anonymity became a critical issue in the cryptography domain. Despite the benefits of absolute anonymity and flexibility provided by ring signatures, ring signers are subject to user attacks since they can misuse their powers. This entire discussion indicates that absolute anonymity in ring signatures is questionable.

To address the above problem, Xu and Yung [166] proposed an accountable ring signature (ARS) technique and it is the first one to identify users. While ARS enable players to pick a ring, they are required to add a tag along with ir, enabling authorities to trace them. To fill the gap between group and ring signature schemes, the ARS is used. When both ring and group signatures schemes fail then in such situations ARS is more useful.

Bootle et al. [27] constructed a short accountable ring signature using random oracles based on the decisional Diffie-Hellman assumption. At the moment of signing, users have the option of choosing a ring and a tracer. Liu et al. [101], on the other hand, proposed the first LRS (linkable ring signatures) in a mechanism for ad-hoc group signature schemes. Wei et al .[161] provided a tracing approach for the signatures entitled tracing-by-linking, which was similar to Liu et al.'s work on LRS [101]. This approach determines the public key of a double signer. The traceable ring signature is simple to use and produces one of three outcomes for the provided two signature-message pairs. It lets the user know if

the specified inputs are linked or independent. If the signer is identified, then it outputs public key of the signer.

Ring signatures have gained a lot of attention in various multi-user cryptographic applications where a main requirement is user anonymity. E-voting, e-cash, and e-auction are examples of such applications. Applications like e-auctions can also be deployed on permissionless blockchains, where unconditional privacy to the user identity is required and at the same time the highest bidder has to be identified.

To address similar problem in blockchains, Saraswat et al. [139] proposed a designated identity verifier ring signature scheme in 2015. In their scheme, the actual signer of the message can prove himself to the designated identity verifier without revealing his identity, at his own choice. Even after the signer has proved to the designated identity verifier that he is the actual signer, the designated identity verifier cannot prove to anyone else that he is indeed the signer of the message. But in blockchain-based e-auction protocols, there is a necessity that the auctioneer has to reveal the winning bid and his identity to all the participants. To address the above challenge, we propose an identity verifiable ring signature scheme in which the actual signer of the message can voluntarily prove himself to all the other participants in the ring that he is indeed the actual signer.

3.2 Preliminaries

In this section, we explain the basic idea of a ring signature and the components that served as the foundation for our proposed scheme; namely (i) combining function (ii) ring signature. Table 3.1 defines the notations we use throughout this chapter.

3.2.1 Notion

Let *Alice* wants to create a ring signature for a message m using a ring of n members, where the signer *Alice* is A_s, $1 \leq s \leq n$, $S = \{A_1, A_2, ..., A_n\}$. Here, each $A_i \in S$, $(1 \leq i \leq n)$ is known as a member of the ring. The key pair for *Alice* (A_s) is (P_s, S_s), where P_s is a public key and S_s is a secret key. In this scheme, the ring member is identified

Notion	Explanation
key-gen	key generation algorithm
ring-sign	Signature generation algorithm
ring-verify	Signature verification algorithm
ring-identify	Signer identification algorithm
A_s	Signer of the message
n	ring size
σ	ring signature
P_i	Public key of i^{th} member
S_i	Secret key of i^{th} member
m	message to be signed
k	Symmetric key computed by signer
v	Initialization value
$C_{k,v}$	Combining function
x_i	Random integers chosen from $\{0,1\}^b$
$g_n(.)$	Trapdoor functions
f_i	Trapdoor one-way permutation
S	Set of public keys
CMA	Chosen message attack

TABLE 3.1: Notions used in this chapter

with his public key. Consequently, S denotes the public keys set of all ring members. The two algorithms listed below make up a ring signature and the security requirements for ring signature techniques are anonymity and unforgeability [133]:

1. **ring-sign(m, S)**: The real signer A_s can generate a ring signature σ using S and his private key S_s, for a given message m and a set of ring members $S = \{A_1, A_2, ..., A_n\}$.

2. **ring-verify(m, σ)**: A ring signature σ comprising $S = \{A_1, A_2, ..., A_n\}$ and a message m are given. Then any verifier can check whether the given $(m,\ \sigma)$ is a valid ring signature produced by one of the ring members or not.

3.2.2 Combining Function

A combining function $C_{k,v}(y_1, y_2, ..., y_n)$ is a function, which takes any key k, initialization value v and $y_1 = g_1(x_1),\ y_2 = g_2(x_2),\ ....,\ y_n = g_n(x_n) \in \{0,1\}^b$ list of arbitrary values as an input, where $g_1,\ g_2, ...,\ g_n$ are trapdoor functions and x_i's $(1 \leq i \leq n, i \neq s)$ are

random integers chosen for all other ring users independently and uniformly from $\{0,1\}^b$. The combining function outputs $z \in \{0,1\}^b$ s.t. for any given $k, v, s, 1 \leq s \leq n$ and any other inputs $y_i, i \neq s$ with fixed values, being one-to-one mapping $C_{k,v}$ maps input y_s to z as an output. This mapping can be solved efficiently, but it is difficult to solve the verification equation for $x_1, x_2, ..., x_n$ without knowledge of secret keys or the inverting of any of the trapdoor functions $g_1,\ g_2, ...,\ g_n$. Rivest et al. [134] proposed the below combining function:

$$\begin{aligned} z &= C_{k,v}(y_1, y_2, ..., y_n) \\ &= C_{k,v}(g_1(x_1), g_2(x_2), ..., g_n(x_n)) \\ &= E_k(g_n(x_n) \oplus E_k(..... \oplus E_k(g_1(x_1) \oplus v))) \end{aligned} \tag{3.1}$$

Equivalently, Ren et al. [133] gave the following combining function:

$$\begin{aligned} y_s = E_k(g_{s-1}(x_{s-1}) \oplus ..... \oplus E_k(g_1(x_1) \oplus)) \oplus E_k^{-1}(g_{s+1}(x_{s+1}) \oplus ..... \\ \oplus E_k^{-1}(g_n(x_n) \oplus E_k^{-1}(z))) \end{aligned} \tag{3.2}$$

3.2.3 Ring Signature

Rivest et al. [134] introduced a ring signature based on the RSA. In this ring signature scheme, one of the ring member A_i possesses an RSA-based public key $P_i = (n_i, e_i)$, obtained through the one-way trapdoor function f_i over Z_{n_i}

$$f_i(x_i) = x_i^{e_i}\ mod\ n_i$$

The assumption mentioned about the inverse permutation f_i^{-1} can be solved efficiently only by A_i. Each ring member of the RSA-based method has a unique modulus, and this makes it hard to combine the signatures. To address this issue, a common domain $\{0,1\}^b$ is created by combining all of the trapdoor permutations, where $2b$ is a power of two, that is larger than remaining moduli n_i's. The g_i over $\{0,1\}^b$ is defined as follows, where g_i an extended trapdoor permutation.

Let $m_i = q_i n_i + r_i$ for any $b - bit$ input m_i, where q_i and r_i are non-negative integers, and $0 \leq r_i \leq n_i$. Then

$$g_i(m_i) = \begin{cases} q_i n_i + f_i(r_i), & if (q_i + 1) n_i \leq 2^b \\ m_i, & \text{else} \end{cases} \tag{3.3}$$

Its presumed that there exists a disclosed ideal symmetric encryption method E such that the function E_k is permuted over $b - bit$ strings for each key k of length l. The existence of a publicly disclosed collision-resistant hash function h, that maps arbitrary length inputs into strings of l length and serves as the key for E, is also assumed. The two algorithms that constitute the ring signature scheme based on RSA is proposed in [134] will be covered in this section:

3.2.3.1 Signing Algorithm

Assume *Alice* wants to sign a message m on behalf of the ring of n members, where A_s *Alice* for any s, $1 \leq s \leq n$. Now, *Alice* will compute ring signature σ of the message m using the public keys set $P_1, P_2, ..., P_n$ of all other ring members and her private key S_s as shown in Algorithm 1.

The Algorithm 1 requires public keys set S and m as an input and outputs a ring signature σ of the message m as a result. The symmetric key k is initially computed by the signer A_s as $k \leftarrow h(m)$. Then, the signer chooses $v \in \{0,1\}^b$ as an initialization value at random. The signer A_s chooses x_i, $1 \leq i \leq n$, $i \neq s$ for remaining ring members independently and uniformly from $\{0,1\}^b$ and computes $y_i = g(x_i)$. Now, A_s uses eq. 3.2 to solve the ring equation for y_s. After solving y_s, the signer A_s applies his trapdoor knowledge to invert g_s on y_s resulting x_s as $x_s = g_s^{-1}(y_s)$. Finally, the ring signature of the message m is $(S, v, x_1, x_2, ..., x_n)$.

Algorithm 1 Rivest et al.'s [134] Signing Algorithm

Require: m *and* S
Ensure: σ
1: $k \Leftarrow h(m)$
2: $v \Leftarrow \{0,1\}^b$
3: $i = 1$
4: **while** $i \leq n$ **do**
5: **if** $i = s$ **then**
6: *Continue*
7: **end if**
8: $x_i \Leftarrow \{0,1\}^b$
9: $y_i \Leftarrow g_i(x_i)$
10: $i = i + 1$
11: **end while**
12: Solve y_s
13: $x_s \Leftarrow g_s^{-1}(y_s)$
14: $\sigma \Leftarrow (S, v, x_1, x_2, ..., x_n)$
15: *return* σ

3.2.3.2 Verification Algorithm

The verification of the message m on ring signature σ can be done as shown in Algorithm 2. Algorithm 2 takes message m, public keys set S and ring signature σ and returns 1 if the ring signature is valid, otherwise 0. The verifier computes $y_i = g_i(x_i)$, $1 \leq i \leq n$ for all the ring members. Then computes hash on the message m to obtain the symmetric key k. The verifier verifies whether the y_i's satisfies $C_{k,v}(y_1, y_2, y_3, ..., y_n) = v$ or not. If satisfied it returns 1 otherwise 0.

Algorithm 2 Rivest et al.'s [134] Verification Algorithm

Require: m, σ *and* S
Ensure: 1 *or* 0
1: $i = 1$
2: **while** $i \leq n$ **do**
3: $y_i \Leftarrow g_i(x_i)$
4: $i = i + 1$
5: **end while**
6: $k \Leftarrow h(m)$
7: **if** $C_{(k,v)} = v$ **then**
8: *return* 1
9: **else**
10: *return* 0
11: **end if**

3.3 Formal Model and Security Definitions of Proposed Identity Verifiable Ring Signature Scheme

This section introduces the mathematical definition of our proposed scheme as well as its security features. The correctness and security definitions discussed in this section are taken from Bender et al.'s [18] paper.

3.3.1 Formal Model of IVRSS

An IVRSS is a tuple of four PPT (Probabilistic Polynomial-Time) algorithms:

$$IVRSS = (key - gen, ring - sign, ring - verify, ring - identify)$$

that are described as follows:

1. **key-gen**: The key generation algorithm accepts λ as an input and returns a public keys set S, secret key S_s, a symmetric key u_k and a random $Nonce$ value as an output.

$$(S, S_s, u_k, Nonce) \leftarrow key - gen(1^\lambda)$$

2. **ring-sign**: The signature generation algorithm $ring - sign(.)$ accepts message m, a randomly chosen $Nonce$, public keys set S of other ring members as an input and returns signature σ as an output.

$$\sigma \leftarrow ring - sign(m, S, Nonce)$$

3. **ring-verify**: The signature verification algorithm $ring - verify(.)$ takes message m, ring signature on the message σ, public keys set S and cipher text C as an input. If the signature is valid it returns 1; otherwise 0.

$$0 \; or \; 1 \leftarrow ring - verify(m, \sigma, \; S \; and \; C)$$

4. **ring-identify**: The signer identification algorithm $ring - identify(.)$ takes message m, ring signature on the message σ, public keys set S, cipher text C, random $Nonce$ and symmetric key u_k associated with $Nonce$ as an input and outputs $true$ (1) if the signer is identified otherwise $false$ (0).

$$0 \; or \; 1 \leftarrow ring - identify(m, \sigma, S, C, Nonce \; and \; uk)$$

3.3.2 Definition of Correctness

An IVRSS is considered to be correct iff the below mentioned two cases hold for any two members of ring $S[i], S[j] \leftarrow S$ and each valid signable message m:

$$\begin{aligned} Pr[(S, S_s, Nonce, u_k) \Leftarrow key - gen(1^\lambda), \sigma \Leftarrow ring - sign(m, S, Nonce) : \\ ring - verify(m, \sigma, \; S \; and \; C) = 1] = 1 \end{aligned} \tag{3.4}$$

$$\begin{aligned} Pr[(S, S_s, Nonce, u_k) \Leftarrow key - gen(1^\lambda), \sigma \Leftarrow ring - sign(m, S, Nonce) : \\ ring - identify(m, \sigma, S, C, Nonce \; and \; uk) = 1] = 1 \end{aligned} \tag{3.5}$$

The eq. 3.4 refers to the correctness of the signature verification and the eq. 3.5 refers to the correctness of the real signer's identity verification.

3.3.3 Unforgeability

An IVRSS is said to be unforgeable if the advantage

$$Adv^{UF-CMA}_{IVRSS,A} = Pr\left[Game^{UF-CMA}_{IVRSS,A}(\lambda) = 1\right]$$

for any PPT adversary A is negligible in the $Game^{UF-CMA}_{IVRSS,A}$ defined in Game Algorithm 3. where the attacker A could not have utilized m to query the signing oracle $ring - sign(S, S_s, Nonce, .)$.

Algorithm 3 $Game^{UF-CMA}_{IVRSS,A}(\lambda)$

Require: λ
Ensure: $ring - verify(S, m, \sigma)$
1: $(S, S_s, Nonce) \Leftarrow key - gen(1^\lambda)$
2: $(m, \sigma) \Leftarrow A^{ring-sign(S,S_s,Nonce,.)}(S)$
3: $return\ ring - verify(S, m, \sigma)$

3.3.4 Signer Anonymity

A $IVRSS$ satisfies anonymity if the advantage

$$Adv^{ANON}_{IVRSS,A} = \left| Pr\left[Game^{ANON-1}_{IVRSS,A}(\lambda) = 1\right] - Pr\left[Game^{ANON-0}_{IVRSS,A}(\lambda) = 1\right] \right|$$

for any adversary $A = (A1, A2)$ is negligible in the $Game^{ANON-b}_{IVRSS,A}$, $b = 0, 1$ defined in Game Algorithm 4.

Algorithm 4 $Game^{ANON-b}_{IVRSS,A}(\lambda)$

Require: λ
Ensure: b
1: $(S, S_s, Nonce) \Leftarrow key - gen(1^\lambda)$
2: $(m, p_0, p_1, st) \leftarrow A_1^{ring-sign(S,S_s,Nonce,.)}(S)$
3: $\sigma \Leftarrow ring - sign(S, S_s, Nonce, m)$
4: $b^1 \Leftarrow A_2^{ring-sign(S,S_s,Nonce,m)}(\sigma, st)$
5: $return\ b$

3.4 Proposed Identity Verifiable Ring Signature Scheme (IVRSS)

In this section, an Identity Verifiable Ring Signature Scheme (IVRSS) is proposed. The proposed scheme comprises of four PPT algorithms: Key generation algorithm, Signature generation algorithm, Verification algorithm and Identification algorithm. In this scheme, the combining function $C_{(k,v)}$ and extended trapdoor permutation $g_i(m_i)$ are adopted from Rivest et al.'s [134] ring signature scheme.

3.4.1 Key Generation Algorithm

The key generation algorithm accepts λ security parameter and returns a public keys set S, secret key S_s, a random $Nonce$ value and a symmetric key u_k for signing nonce as an output for any signer.

Algorithm 5 Key Generation Algorithm

Require: λ
Ensure: $(S, S_s, u_k, Nonce)$
1: $(S, S_s, u_k, Nonce) \Leftarrow key - gen(1^\lambda)$
2: $return\ (S, S_s, u_k, Nonce)$

3.4.2 Signature Generation Algorithm

Given a message m, a signer A_s with secret key S_s, public keys set S of all other group members and a randomly chosen, symmetric key u_k, $Nonce$ can generate a ring signature by Algorithm 6. The Algorithm 6 takes a message m, public keys set S and a random value $Nonce$ as an input and returns σ (ring signature) of the message m as an output.

The signer encrypts the random $Nonce$ value using a symmetric key u_k and generates cipher text C. Then, the symmetric key k is computed by the signer A_s as $k \leftarrow h(m||C)$. Now, the signer chooses $v \in \{0,1\}^b$ as an initialization value at random. The signer A_s chooses x_i, $1 \leq i \leq n$, $i \neq s$ for all other ring members independently and uniformly from $\{0,1\}^b$ and calculates $y_i = g(x_i)$. Now, A_s uses eq.3.2 to solve the ring equation for y_s. After solving y_s, the signer A_s applies his trapdoor knowledge to invert g_s on y_s to get x_s as $x_s = g_s^{-1}(y_s)$. The obtained ring signature of message m is $(S, v, C, x_1, x_2, ..., x_n)$.

3.4.3 Signature Verification Algorithm

The verification of the message m on ring signature σ can be done as shown in Algorithm 7. The Algorithm 7 accepts ring signature σ, message m, public keys set S and cipher text C and returns $true$ (1) if the ring signature is valid, otherwise returns $false$ (0). The verifier computes $y_i = g_i(x_i)$, $1 \leq i \leq n$ for all the ring members. Then to get the

Algorithm 6 Signature Generation Algorithm

Require: m, S, u_k *and* $Nonce$
Ensure: σ
1: $C \Leftarrow E_{u_k}(Nonce)$
2: $k \Leftarrow h(m||C)$
3: $v \Leftarrow \{0,1\}^b$
4: $i = 1$
5: **while** $i \leq n$ **do**
6: **if** $i = s$ **then**
7: *Continue*
8: **end if**
9: $x_i \Leftarrow \{0,1\}^b$
10: $y_i \Leftarrow g_i(x_i)$
11: $i = i + 1$
12: **end while**
13: Solve y_s (using eq. 3.2)
14: $x_s \Leftarrow g_s^{-1}(y_s)$
15: $\sigma \Leftarrow (S, v, C, x_1, x_2, ..., x_n)$
16: *return* σ

symmetric key k, he computes $h(m||C)$. Then, the verifier checks whether the y_i's satisfy $C_{k,v}(y_1, y_2, ..., y_n) = v$ or not. If it is satisfied then returns *true* (1) otherwise returns $false$ (0).

Algorithm 7 Signature Verification Algorithm

Require: m, σ, S, *and* C
Ensure: 1 *or* 0
1: $i = 1$
2: **while** $i \leq n$ **do**
3: $y_i \Leftarrow g_i(x_i)$
4: $i = i + 1$
5: **end while**
6: $k \Leftarrow h(m||C)$
7: **if** $C_{(k,v)}(.) == v$ **then**
8: *return* 1
9: **else**
10: *return* 0
11: **end if**

3.4.4 Signer Identification Algorithm

The signer identification of the message m on ring signature σ can be done as shown in Algorithm 8. As and when required the real signer of the message m can voluntarily

reveal his secret $Nonce$ to prove himself as indeed the signer. Algorithm 8 accepts message m, ring signature σ, public keys set S, cipher text C, $Nonce$ and returns uk if the ring signature is valid, otherwise returns 0.

Algorithm 8 Signer Identification Algorithm

Require: m, σ, S, C, $Nonce$ *and* u_k
Ensure: 0 *or* 1
1: $i = 1$
2: **while** $i \leq n$ **do**
3: $y_i \Leftarrow g_i(x_i)$
4: $i = i + 1$
5: **end while**
6: $k \Leftarrow h(m||C)$
7: **if** $C_{(k,v)}(.) == v$ **then**
8: **if** $D_{uk}(C) == Nonce$ **then**
9: $return\ uk$
10: **else**
11: $return\ 0$
12: **end if**
13: **else**
14: $return\ 0$
15: **end if**

3.5 Correctness and Security Analysis

We now analyse the security of proposed IVRSS for unforgeability, anonymity and correctness.

Theorem 3.1. *The proposed IVRSS is unforgeable against chosen message attack.*

Proof. An Identity Verifiable Ring Signature Scheme $IVRSS(key-gen, ring-gen, ring-verify, ring-identify)$ is unforgeable if for any polynomial $n(.)$ and for every PPT adversary A, the probability that A wins the game is negligible against chosen message attack:

1. Key pairs (P_i, S_i), $(1 \leq i \leq n)$ are generated using $key - gen(1^\lambda)$, and the set of public keys P is given to A.

2. Adversary A is given access to a signing oracle $Oring - sign(.,.,.,.)$, where $Oring - sign(S_s, S, m, Nonce)$ outputs $ring - sign(m, S)$ and it is required that $S \subseteq P$ and $Ps \in S$.

3. Adversary A outputs $(P^*, m^*, \sigma^*, Nonce^*)$ and succeeds if $S^* \subseteq P$, $ring{-}verify(m^*, \sigma^*, C^*) = 1$ and A never queried $(*, m^*, S^*, Nonce^*)$ to its signing oracle.

Theorem 3.2. *The unconditional signer anonymity exists in our proposed IVRSS.*

Proof. Consider the following game with An Identity Verifiable Ring Signature Scheme $IVRSS(key - gen, ring - gen, ring - verify, ring - identify)$, a polynomial $n(.)$, and a PPT adversary A:

1. Key pairs (P_i, S_i), $(1 \leq i \leq n)$ are generated using $key - gen(1^\lambda)$, and the set of public keys P is given to A.

2. A is given access to a signing oracle $Oring{-}sign(.,.,.,.)$, where $Oring{-}sign(S_s, S, m, Nonce)$ outputs $ring - sign(m, S)$ and we require that $S \subseteq P$ and $Ps \in S$.

3. A outputs a message m, distinct indices i_0, i_1, and a ring $S \in P$ for which $P_{i0}; P_{i1} \in R$. A random bit b is chosen, and A is given the signature $\sigma \leftarrow ring{-}sign(m, S, Nonce)$.

4. The adversary outputs a bit b_0, and succeeds if $b_0 = b$.

Theorem 3.3. *The correctness of proposed IVRSS is implied successful iff the signature is verified and the real identity of the actual signer is uncovered.*

Proof.

$$[key - gen(1^k), \sigma \Leftarrow ring - sign(m, S, u_k, Nonce) :$$
$$ring - verify(m, \sigma,\ S\ and\ C) = 1] = 1$$

$$[key - gen(1^k), \sigma \Leftarrow ring - sign(m, S, Nonce) :$$
$$ring - identify(m, \sigma, S, C, Nonce\ and\ uk) = 1] = 1$$

The correctness of our proposed IVRSS relies on the following two conditions:

1. For the given message m, signature σ, public keys set S and a cipher text over nonce value C, the $ring - verify()$ algorithm should return 1 when it is valid.

2. For the given message m, signature σ, public keys set S, randomly chosen $Nonce$, a cipher text over nonce value C and symmetric key u_k, the $ring - identify()$ should return 1 when it is valid.

3.6 Comparison

In general, in applications like e-auctions, users prefer to protect their identity privacy when using the service for a fair exchange. Many different versions of the digital signature schemes were proposed in the literature. Some signature schemes [126] themselves are capable of giving the signer anonymity. Ring signature schemes are currently used to protect user identity privacy in blockchains. This section compares our proposed IVRSS with the existing ring signature schemes, which can be used in blockchain-based e-auction protocol to protect user identity privacy and comparison is given in Table.3.2

Scheme	**Who can verify**	**Application**	**Anonymity**	**Unforgeability**	**Transparency**
ARS [166]	Tracer	E-auction	Yes	Yes	No
LRS [101]	Linkability	E-voting	Yes	Yes	No
TRS [60]	Tracer	E-voting	Yes	Yes	No
DIVRS [139]	Identity verifier	E-auction	Yes	Yes	No
IVRSS	Any ring member	E-auction	Yes	Yes	**Yes**

TABLE 3.2: Comparison of tracing methods in ring signature

In the accountable ring signature [166], the signer may sign on behalf of any group (or ring), and he may select a master independent of the group. The master doesn't have to participate in the key generation of the parties in the ring to reveal the signer's identity to the group. Linkable ring signature [101] is a suitable scheme for e-cash and e-voting applications among the schemes given in Table 3.2 for ring signatures as they prevent re-submissions of votes. Note that only signatures of the same event can be linked only when signatures are generated. Therefore, it is impossible to link the same signer who makes signatures for various actions. Additionally, the TRS [60] possesses a quality that

maintains a balance between traceability and anonymity. So, TRS have an advantage over other methods as they can be used to trace a person's identity with some anonymity. However, ring size is the delimiting factor in all these schemes. The signer S of a message has the additional authority, under the designated identity verifier ring signature approach, to at any moment and without exposing his identify to anyone else, prove his identity to the designated identity verifier V.

Saraswat et al. [139] proposed a designated identity verifier ring signature scheme. In their scheme, the actual signer of the message can prove himself to the designated identity verifier without revealing his identity, at his own choice. Even after the signer has proved to the designated identity verifier that he is the actual signer, the designated identity verifier cannot prove to anyone else that he is the indeed signer of the message. To address the above challenge, we proposed a verifiable ring signature scheme (IVRSS) in which the actual signer of the message can voluntarily prove himself to all the other participants in the ring that he is the actual actual signer.

The use of ring signatures in blockchain-based e-auctions helps protect user identity privacy due to the unconditional anonymity they offer to the signers. However, the highest bidder or winner must be declared when the auctions are over. To the best of our knowledge, all the current ring signatures, and even those that have been enhanced for the blockchain domain, do not offer any mechanisms for identifying the actual message signers. Our proposed $IVRSS$ keeps its applicability for all prior applications of a ring signature with an additional mechanism of proving the signer's identity to all other ring members voluntarily of his own choice. Without adding any extra overhead, our proposed $IVRSS$ keeps all the characteristics of the original ring signatures.

3.7 Conclusion

Existing ring signatures have an everlasting anonymity and seems to benefit users, but they are unable to identify the real signers of the message if any such requirement arises. In this chapter, an identity verifiable ring signature scheme is proposed, that not only

has all of the properties of a ring signature, but also the property that the signer can voluntarily prove himself to other members that he is the indeed signer of the message.

Chapter 4

A Decentralized Group Signature Scheme for Privacy Protection in Permissioned Blockchains

In the previous chapter an identity verifiable ring signature scheme for privacy protection in public blockchain based e-auction protocols is proposed. In some situations, e-auctions are conducted for a group of known participants, such as registered contractors, approved suppliers, and members of an organization. In this, auctions might need to fulfil certain restrictions, such as having a certain degree of expertise or fulfilling particular technological specifications. This condition requires a private blockchain where as IVRSS is for public blockchains. Hence, to address the identity privacy issue of the users in private blockchain based e-auction protocol, a decentralized group signature scheme (DGSS) is proposed in this chapter. The proposed DGSS is more suitable for private blockchain-based sealed-bid e-auction applications.

The rest of the chapter is organized as follows: The first section defines the problem statement. Correctness and security definitions are discussed in second section. The third section describes proposed decentralized group signature scheme for privacy protection in blockchains. The numerical example on the proposed scheme is discussed in fourth section. In section five, proof of correctness and security analysis of the proposed scheme

is discussed. The sixth section describes comparison of proposed scheme with other similar schemes. Finally, the chapter is concluded in seventh section.

4.1 Problem Statement

A private blockchain is designed for use within a specific community or organization [172]. It has strict access controls to ensure that only authorized users can participate. This private network is made up of several nodes, each with specific responsibilities and roles. The participants are authorized members of the network who can transact and access the distributed ledger [76]. In the network, nodes can maintain a copy of the ledger and run the blockchain. There are various types of nodes in a permissioned or private blockchain, including client nodes, validator nodes and consensus nodes [9]. The validator nodes are responsible for transaction validation and appending them into the blockchain. Consensus nodes are responsible to participate in the consensus algorithm for reaching consensus. In a private blockchain, the responsibilities and roles of nodes are typically given and controlled by the governing organization or the network administrator.

The group signature scheme is more suitable for private blockchain systems since it requires a group manager to set up the group. Idea of digital signatures was extended for groups by Chaum and Heyst [35] where a first group signature scheme was proposed in 1991. In private blockchains systems, we also require an authority to control the network. PlatON [127] utilized group signature scheme in their platform to provide users anonymity. Any member of the group can anonymously sign a message using group signature without revealing his real identity and the identity of the signer can be revealed by a designated group manager of group.

Ateniese et al. [14] discussed different security weaknesses in earlier group signature schemes and presented a proven-secure group signature. Some open challenges and new research directions in group signature scheme were discussed. In 2009, Lee et. al. [92] proposed a new group signature scheme that can achieve authenticity, integrity and non-repudiation with confidentiality by using authenticated encryption. Using this new group signature scheme, they designed a sealed-bid auction protocol that confidentiality of the

bids is maintained till the bids are opened. To restrict the tracing capabilities to message related openings, Sakai et al. [137] suggested a message-dependent opening for group signatures in 2012. In 2013, Sun et. al. [152] have proposed another group signature scheme by adding one more random number to Lee et al.'s group signature to improve the security weaknesses. A token indicates the user's revocation status in group signature systems with verifier-local revocation [91]. Using lattice hardness difficulties, Ling et al. [100] proposed a quantum-safe accountable tracing group signature system. In 2018, Tsai et. al. [155] claimed that their group signature scheme is based on the discrete logarithm problem that addresses security and efficiency concerns.

In literature all the researchers have focused on various security issues of group signature schemes and their designated group manager has always remained as a trusted third party only. In this chapter, we decentralize the role of designated group manager of a static group signature scheme to address the trust related issues of centralized group manager. Also, we utilized our proposed decentralized group signature scheme (DGSS) to enhance the bidder's privacy in private blockchain-based closed-bid e-auction protocol.

4.2 Correctness and Security Definitions

In this chapter, we adopt the definition of the group signature schemes and security definitions from the work of Bellare, Micciancio, and Warinschi [16].

Definition 3.1 (decentralized group signature scheme) A decentralized group signature scheme $DGSS = (Init, Sign, Verify, Identify)$ is a collection of four polynomial-time algorithms defined as following:

Init: The initiation algorithm $Init$ takes the secret key of group managers x_j, random integer k_{ij} in Z_q^* chosen by group managers, public key y_i $(1 \leq i \leq m)$ of the group members as an input and returns (r_{ij}, s_{ij}) $(1 \leq j \leq n)$.

Sign: The signing algorithm $Sign$ takes message $M_{original}$, an attachment value M_{check}, two random integers N_1, N_2 in Z_q^* as input and returns group signature $\{A, B, C, D, M_{check}\}$.

Verify: The verification algorithm $Verify$ takes group signature $\{A, B, C, D, M_{check}\}$, private key of the receiver x_l, collision-resistant hash function $h(.)$ as input and returns 1 if the signature is valid otherwise 0.

Identify: The identifying algorithm $Identify$ takes public keys of the group members $y_i (1 \leq i \leq n)$, random integers $k_{ij} (1 \leq j \leq m)$ of each group member, parameter B of DGSS as input and returns the public key y_i of the actual signer.

Definition 3.2 (Correctness) The signature produced by the honest group member should always be accepted i.e., the Verify(.) algorithm should return 1. The Identify(.) algorithm should always identify the actual signer of the message for the given valid message and DGSS.

Definition 3.3(Unforgeability) It is computationally difficult for any unauthorized member to produce a valid signature on behalf of the group. Only authorized member of the group can produce a valid signature on behalf of the group.

Definition 3.4(Anonymity) It is computationally difficult for anyone to determine the actual signer of the message for a given valid group signature.

Definition 3.5 (Unlinkability) It is computationally difficult for anyone to determine if two valid group signatures are produced by the same user or not.

Definition 3.6 (Traceability) It is computationally difficult for anyone except the group managers to track the identity of the actual singer. If there is any dispute among the group members or as per requirement, all the group managers together can identify the actual signer.

4.3 Proposed Decentralized Group Signature Scheme (DGSS)

To address the user identity privacy challenges in blockchain, a decentralized group signature scheme ($DGSS$) is proposed that is based on the difficulty of discrete logarithm problem. In literature most of the existing group signature schemes [2] contain designated group manager as centralized party to reveal the identity of the group member based on

requirement. The proposed scheme is based on the assumption that a group manager may be malicious. A malicious manager carries the risk of revealing identities and collusion attack. On the other hand, origin of blockchain has begun to bring the trust among the untrusted party, where the individual party can behave maliciously but as a group they cannot. The existing group signature schemes available in the literature are not suitable to address the privacy leakage of e-auction protocol because of its centralized group manager. In this section we decentralize the designated group manager of the discrete logarithm-based group signature scheme [92] to address the privacy issue of blockchain-based e-auction protocol. After decentralizing the group manager's role, all the group managers together can only open signer's identity when it is required.

The proposed *DGSS* comprises of four polynomial-time algorithms: the Initiation algorithm, the Signing algorithm, Verification algorithm, and the Identification algorithm. The decentralized group signature scheme (*DGSS*) is described as follows:

4.3.1 Initiation Algorithm

Let p and q be two large prime numbers such that $q|p-1$, and g be a generator with order q in $GF(p)$. Each group member U_i $(1 \leq i \leq \text{m})$ selects the private key x_i and computes the public key y_i=g^{x_i} mod p. Receiver l chooses his private key x_l randomly and computes public key $y_l = g^{x_l}$ mod p. Each group manager T_j $(1 \leq j \leq \text{n})$ selects his private key x'_j and computes the public key y'_j=$g^{x'_j}$ mod p. For each group member U_i, each group manager T_j randomly chooses an integer k_{ij} in Z_q^* and computes

$$r_{ij} = (y_i.k_{ij} - x'_j)\ mod\ q \tag{4.1}$$

$$s_{ij} = y_i^{k_{ij}}\ mod\ p \tag{4.2}$$

Now, each group manager T_j sends (r_{ij}, s_{ij}) pair to the group member U_i. After receiving(r_{ij}, s_{ij}) pairs from all the group managers, the group member U_i computes his certificate as follows:

$$R_i = \sum_{j=1}^{n} r_{ij} \tag{4.3}$$

$$S_i = \prod_{j=1}^{n} s_{ij} \tag{4.4}$$

After computing (R_i,S_i), the group member U_i can verify the validity of the certificate (R_i,S_i) by checking the following equation:

$$S_i^{y_i} \ mod \ p = (g^{R_i}.\prod_{j=1}^{n} y'_j)^{x_i} \ mod \ p \tag{4.5}$$

Proof of validity is given in sec.4.5.1

4.3.2 Signing Algorithm

A short message M_{check} is added as a verification test. Group member U_i generates a decentralized group signature for message $M_{original}$ by computing the following steps:

1. Compute $M = M_{check} \| M_{original}$, where $\|$ is a concatenation.
2. Group member U_i selects two random numbers N_1,N_2 in Z_q^*.
3. U_i computes four parameters A, B, C, D as follows:

$$A = x_i.N_1.N_2 \ mod \ q \tag{4.6}$$

$$B = S_i^{N_1.N_2.y_i} \ mod \ p \tag{4.7}$$

$$C = M.y_l^{-N_1.A.h(B)} \ mod \ p \tag{4.8}$$

$$D = N_1 - R_i.h(C) \ mod \ q \tag{4.9}$$

4. Decentralized Group signature for the message M is {A, B, C, D, M_{check}}.

4.3.3 Verification Algorithm

Receiver can now reconstruct and check the validity of message M using the following steps:

1. Reconstruction of the message M is computed as follows:

$$M = C.\left[g^{D.A}.\prod_{j=1}^{n} y_j'^{-h(C).A}.B^{h(C)}\right]^{x_l.h(B)} \quad mod\ p \tag{4.10}$$

2. And message M is valid if and only if

$$M_{check} \stackrel{?}{=} head(M, s) \tag{4.11}$$

Where, $h(.)$ is a collision-resistant hash function; M_{check} is a binary string with s bits; and $head(M, s)$ is a function which returns the first s bits of binary string M. The signature is valid if and only if the above equation holds. The proof of validity is given in section 4.5.

4.3.4 Identification Algorithm

The *DGSS* must be opened to reveal the identity of the actual signer when there is a dispute among the group members or if there is any such requirement. As the group manager T_j has access to (y_i, k_j) of each group member U_i, the group manager T_j acquires the (y_i, k_{ij}) of U_i and looks for the signature that satisfies following equation:

$$B = g^{A.\sum_{j=1}^{n} k_{ij}.y_i} \quad mod\ p \tag{4.12}$$

for i= 1, 2, 3, ..., n, where n is the size of group. Thereby, the group manager can determine the signer.

4.4 DGSS Example

In this section, numerical example of the proposed *DGSS* is discussed in detail.

4.4.1 Initiation Algorithm

- Let p=227, q=113 such that $q|p-1$
- GF(227)={0,1,2,.....,226} and 4 is generator with order q.
- Group member U_1 chooses private key x_1=3 and computes public key y_1=64 (i.e.,$y_i = g^{x_i} \bmod p$)
- Each group manager T_j $(1 \leq j \leq 3)$ computes their corresponding (r_{1j}, s_{1j}) pairs as follows:

Group Manager T_1:

- T_1 chooses his private key x'_1=4 and computes public key y'_1=29
- T_1 randomly chooses an integer k_{11}=3 from z^*_{113} for group member U_1 and computes (r_{11}, s_{11}) pair as follows:

$r_{11} = (64 \times 3 - 4) \bmod 113 = 75$ (from (4.1))
$s_{11} = 64^3 \bmod 227 = 186$ (from (4.2))

- T_1 sends (r_{11}, s_{11})=(75, 186) to U_1

Group Manager T_2:

- T_2 chooses his private key x'_2=5 and computes public key y'_2=116
- T_2 randomly chooses an integer k_{12}=7 from z^*_{113} for group member U_1 and computes (r_{12}, s_{12}) pair as follows:

$r_{12} = (64 \times 7 - 5) \bmod 113 = 104$ (from (4.1))
$s_{12} = 64^7 \bmod 227 = 213$ (from (4.2))

- T_2 sends (r_{12}, s_{12})=(104, 213) to U_1

Group Manager T_3:

- T_3 chooses his private key x'_3=6 and computes public key y'_3=10
- T_3 randomly chooses an integer k_{13}=8 from z^*_{113} for group member U_1 and computes (r_{13}, s_{13}) pair as follows:

  $r_{13} = (64 \times 8 - 6) \bmod 113 = 54$ (from (4.1))
  $s_{13} = 64^8 \bmod 227 = 12$ (from (4.2))

- T_3 sends (r_{13}, s_{13})=(54, 12) to U_1
- Now, group member U_1 computes his (R_1, S_1) pair as follows:

  R_1=(75+104+54) mod 113=7 (from (4.3)) $S_1 = (186 \times 213 \times 12) \bmod 227 = 78$ (from (4.4))

- U_1 verifies the correctness of of his (R_1, S_1) pair using the equation no. (4.5):

  $78^{64}\ mod\ 227 = \left[4^7(29 \times 116 \times 10)\right]^3\ mod\ 227$

  82=82

- Since, the equation (4.5) holds, the (R_1, S_1) pair is valid.

4.4.2 Signing Algorithm

- Group manager U_1 generates a group signature for the message $M_{Original} = 27$ by concatenating $M_{Check} = 5$ using the following steps:

  1. M=27||5 ($M = M_{Original}||M_{Check}$)
     $M = 221$ (Concatenation over binary string)
  2. U_1 selects two random integers $N_1 = 4$, $N_2 = 5$ in Z^*_{113}
  3. U_1 computes four parameters A, B, C, D as follows:

$$A = (3 \times 4 \times 5)\ mod\ 113 \qquad (from(4.6))$$

A=60

$$B = 78^{4\times5\times64}\ mod\ 227 \qquad (from(4.7))$$

B=7

$$C = 221 \times 16^{-4\times60\times h(7)}\ mod\ 227 \qquad (from(4.8))$$

C=203

$$D = 4 - 7 \times h(203)\ mod\ 113 \qquad (from(4.9))$$

D=61

Note: Let h(7)=3 and h(203)=8.

4. Group signature for the message 221 is $\{60, 7, 203, 61, 5\}$

4.4.3 Verification Algorithm

- Receiver can now reconstruct the message using the group signature $\{60, 7, 203, 61, 5\}$ and check the validity of the message using his private key x_{11}=2 and public y_{11}=16.

1. Reconstruction of the message is computed using the eq.(4.10):

$$203 \times \left[4^{61\times60} \times (29 \times 116 \times 10)^{-8\times60} \times 7^8\right]^{2\times4}\ mod\ 227$$

$$=203 \times \left[4^{3660} \times (33640)^{-480} \times 7^8\right]^8\ mod\ 227$$

$$=203 \times \left[4^{3660} \times (33640)^{198} \times 7^8\right]^8\ mod\ 227$$

$$=203 \times \left[4^{44} \times (33640)^{198} \times 7^8\right]^8\ mod\ 227$$

$$=221$$

2. The message 221 is valid iff the eq. (4.11) holds.

   head(221,3)=head(11011101,3)

   =101

   =5

- Since, the equation (4.11) holds, the message is valid.

4.4.4 Identification Algorithm

- All the group managers uses the public keys of the group members and their random integer k_{ij} to identify the actual signer of the message.
- The public key of group member U_1 is y_1=64 and random integers of all the group managers are k_{11}=3, k_{12}=7 and k_{13}=8.
- If the equation (4.12) holds then the user with public key x_1=3 is the actual signer. So, by eq. 4.12

  $B = 4^{60\times((3\times64)+(7\times64)+(8\times64))}\ mod\ 227$

  $=4^{60\times(192+448+512)}\ mod\ 227$

  $=4^{60\times1152}\ mod\ 227$

  $=4^{69120}\ mod\ 227$

  $=4^{190}\ mod\ 227$

  $=7$

- The equation (4.12) holds. Hence the user with public key $y_1 = 64$ is the actual signer of the message.

4.5 Proof of Correctness and Security Analysis

The security analysis and proof of correctness for proposed DGSS is discussed in this section. The proposed *DGSS* is holding all the security properties of Lee et.al. [92] group signature scheme namely correctness, unforgeability, anonymity, unlinkability and traceability even after decentralizing the group manager.

4.5.1 Correctness for certificate

After computing (R_i, S_i) pair the group member U_i can verify the validity of the certificate as follows by taking R.H.S of eq. 4.5:

$$(g^{R_i}.\prod_{j=1}^{n} y'_j)^{x_i} \ mod\ p$$

$$= (g^{\sum_{j=1}^{n} r_{ij}}.\prod_{j=1}^{n} y'_j)^{x_i} \ mod\ p \qquad \text{(from (3.4))}$$

$$= (g^{\sum_{j=1}^{n} (y_i.k_{ij}-x'_j)}.\prod_{j=1}^{n} y'_j)^{x_i} \ mod\ p \qquad \text{(from (4.1))}$$

$$= (g^{y_i.\sum_{j=1}^{n} k_{ij}-\sum_{j=1}^{n} x'_j}.\prod_{j=1}^{n} g^{x'_j})^{x_i} \ mod\ p$$

$$= (g^{y_i.\sum_{j=1}^{n} k_{ij}-\sum_{j=1}^{n} x'_j}.g^{\sum_{j=1}^{n} x'_j})^{x_i} \ mod\ p$$

$$= g^{x_i.y_i.\sum_{j=1}^{n} k_{ij}} \ mod\ p$$

$$= (g^{x_i.\sum_{j=1}^{n} k_{ij}})^{y_i} \ mod\ p$$

$$= (\prod_{j=1}^{n} g^{x_i.k_{ij}})^{y_i} \ mod\ p$$

$$= (\prod_{j=1}^{n} y_i^{k_{ij}})^{y_i} \ mod\ p$$

$$= (\prod_{j=1}^{n} s_{ij})^{y_i} \ mod\ p \qquad \text{(from (4.2))}$$

$$= S^{y_i}$$

It is discussed in sec 4.3.3 that group signature on message M={A, B, C, D, M_{check}} from the R.H.S of equation 4.10, it can be easy to determine the message M can be reconstructed as follows:

$$\text{C.}\left[g^{D.A}.\prod_{j=1}^{n} y_j'^{-h(C).A}.B^{h(C)}\right]^{x_l.h(B)} \ mod\ p$$

$$= C.\left[g^{(N_1 - R_i.h(C)).A}.g^{-\sum_{j=1}^{n} x_j'.h(C).A}.S_i^{N_1.N_2.y_i.h(C)}\right]^{x_l.h(B)} \ mod\ p \qquad (from (4.9))$$

$$= C.\left[g^{N_1.A - R_i.h(C).A}.g^{-\sum_{j=1}^{n} x_j'.h(C).A}.(g^{R_i}.\prod_{j=1}^{n} y_j')^{x_i.N_1.N_2.h(C)}\right]^{x_l.h(B)} \ mod\ p \quad (from (4.5))$$

$$= C.\left[g^{N_1.A - R_i.h(C).A}.g^{-\sum_{j=1}^{n} x_j'.h(C).A}.g^{R_i.x_i.N_1.N_2.h(C)}.g^{\sum_{j=1}^{n} x_j'.x_i.N_1.N_2.h(C)}\right]^{x_l.h(B)} \ mod\ p$$

$$= C.\left[g^{N_1.A - R_i.h(C).A}.g^{-\sum_{j=1}^{n} x_j'.h(C).A}.g^{R_i.A.h(C)}.g^{\sum_{j=1}^{n} x_j'.A.h(C)}\right]^{x_l.h(B)} \ mod\ p \qquad (from (4.6))$$

$$= C.\left[g^{N_1.A - R_i.h(C).A. - \sum_{j=1}^{n} x_j'.h(C).A. + R_i.A.h(C) + \sum_{j=1}^{n} x_j'.A.h(C)}\right]^{x_l.h(B)} \ mod\ p$$

$= M.y_l^{-N_1.A.h(B)}.g^{N_1.A.x_l.h(B)} \ mod \ p$ (from (4.8))

$= M.g^{-N_1.A.h(B).x_l}.g^{N_1.A.x_l.h(B)} \ mod \ p$

$= M.g^{-N_1.A.h(B).x_l + N_1.A.x_l.h(B)} \ mod \ p$

$= M$

4.5.2 Security Analysis

The security of proposed *DGSS* is based on the difficulty of the discrete logarithm problem. The *DGSS* satisfies all the security properties as follows:

4.5.2.1 Unforgeability

Attacker can generate a valid group signature if and only if he have a valid (R_i, S_i) and x_i. Even with the assumption that attacker has a valid (R_i, S_i), in order to generate a valid group signature, he first need to compute value of B by Eq.4.7, which is not feasible as N_1, N_2 are not known and then values of parameters A, C, D by eq.4.6, 4.8 and 4.9 are also not known. In addition, as per the proposed scheme the attacker does not have the secret key x_i, thereby, he can never be able to forge the group signature.

4.5.2.2 Anonymity

Given a valid group signature {A, B, C, D, M_{check}} it is difficult for anyone except the group managers to identify the actual signer. All the private information inside group signature is protected by random parameters. In group signature {A, B, C, D, M_{check}}, only A and B have identity information. So, whether the scheme has anonymity by A and B, or not is discussed below.

Attack 1: Given a valid group signature {A, B, C, D, M_{check}} and the equation $A = x_i.N_1.N_2 \ mod \ q$ one can compute

$$g^A = g^{x_i.N_1.N_2} \ mod\ p$$

$$= y_i^{N_1.N_2} \ mod\ p$$

If the attacker has N_1, N_2 then he/she can compute y_i and can find the actual signer's identity. But the random integers N_1, N_2 are unknown and thus its not feasible to find out the actual signer. So, the proposed *DGSS* has anonymity by parameter A.

Attack 2: Given a valid group signature {A, B, C, D, M_{check}} and the equation $B = S_i^{N_1.N_2.y_i} \ mod\ p$ one can compute

$$S_i^{N_1.N_2.y_i} = (g^{R_i}.\prod_{j=1}^{n} y'_j)^{x_i.N_1.N_2} \ mod\ p \qquad \text{(from (4.5))}$$

$$= (g^{\sum_{j=1}^{n} r_{ij}}.\prod_{j=1}^{n} g^{x'_j})^{x_i.N_1.N_2} \ mod\ p \qquad \text{(from (3.4))}$$

$$= (g^{\sum_{j=1}^{n}(y_i.kij - x'_j)}.g^{\sum_{j=1}^{n} x'_j})^{x_i.N_1.N_2} \ mod\ p \qquad \text{(from (4.1))}$$

$$= g^{\sum_{j=1}^{n} kij.y_i.x_i.N_1.N_2} \ mod\ p$$

$$= y_i^{\sum_{j=1}^{n} k_{ij}.N_1.N_2.y_i} \ \text{mod p}$$

If the attacker has $\sum_{j=1}^{n} k_{ij}$, N_1, N_2 then he/she can compute y_i and can find the actual signer's identity. But $\sum_{j=1}^{n} k_{ij}$, N_1, N_2 are unknown and hence no one can find out the actual signer. So, the proposed *DGSS* has anonymity by B. Because of anonymity of A and B, proposed *DGSS* has anonymity by C and D respectively by Eq. 4.8 and Eq. 4.9. Hence, entire group signature {A, B, C, D, M_{check}} has anonymity.

4.5.2.3 Unlinkability

Lemma: To determine whether the two group signatures {A, B, C, D, M_{check}} and $\{A', B', C', D', M'_{check}\}$ are generated by the same user, the following equation should hold,

$$\frac{B}{B'} = \left(\frac{g^A}{g^{A'}}\right)^{\sum_{j=1}^{n} k_{ij}.y_i} \quad mod\ p \tag{4.13}$$

Corollary: It is computationally infeasible to determine that two group signatures were generated by the same user.

Proof:

$$\frac{B}{B'} = \frac{S_i^{N_1.N_2.y_i}}{S_i^{N'_1.N'_2.y_i}}\ mod\ p \qquad \text{(from (4.7))}$$

$$= \frac{\left(\prod_{j=1}^{n} s_{ij}\right)^{N_1.N_2.y_i}}{\left(\prod_{j=1}^{n} s_{ij}\right)^{N'_1.N'_2.y_i}}\ mod\ p \qquad \text{(from (3.5))}$$

$$= \frac{\left(\prod_{j=1}^{n} {y_i}^{k_{ij}}\right)^{N_1.N_2.y_i}}{\left(\prod_{j=1}^{n} {y_i}^{k_{ij}}\right)^{N'_1.N'_2.y_i}}\ mod\ p \qquad \text{(from (4.2))}$$

$$= \frac{\left({y_i}^{\sum_{j=1}^{n} k_{ij}}\right)^{N_1.N_2.y_i}}{\left({y_i}^{\sum_{j=1}^{n} k_{ij}}\right)^{N'_1.N'_2.y_i}}\ mod\ p$$

$$= \frac{\left(g^{x_i \sum_{j=1}^{n} k_{ij}}\right)^{N_1.N_2.y_i}}{\left(g^{x_i \sum_{j=1}^{n} k_{ij}}\right)^{N'_1.N'_2.y_i}}\ mod\ p$$

$$= \frac{\left(g^{x_i \cdot N_1 \cdot N_2}\right)^{\sum_{j=1}^{n} k_{ij} \cdot y_i}}{\left(g^{x_i \cdot N_1' \cdot N_2'}\right)^{\sum_{j=1}^{n} k_{ij} \cdot y_i}} \; mod\ p$$

$$= \left(\frac{g^A}{g^{A'}}\right)^{\sum_{j=1}^{n} k_{ij} \cdot y_i} \; mod\ p \qquad \text{(from (4.6))}$$

Corollary stands true because attacker don't have knowledge of $\sum_{j=1}^{n} k_{ij}.y_i$, and solving eq. 4.13 boils down to solve DLP hard problem along with unknown random parameter k_{ij}.

4.5.2.4 Traceability

The group managers T_j have access to $(y_i, \sum_{j=1}^{n} k_{ij})$ of each group member U_i. So, they can acquire (y_i, k_j) of U_i satisfying the equation $B = g^{A.\sum_{j=1}^{n} k_{ij}.y_i} \; mod\ p$ for i=1, 2, ..., m

Where m is the number of group members. So, the set of group managers together can determine the actual signer, thereby making the proposed DGSS traceable if required.

4.6 Blockchain-based e-auction protocol using DGSS

In this section a detailed application of permissioned blockchain-based sealed-bid e-auction protocol using our proposed *DGSS* is discussed. Also different types of roles used in proposed permissioned blockchain-based e-auction protocol are discussed.

4.6.1 Roles

There are mainly four roles in the proposed protocol: Bidder, Registration Manager, Auction Manager and Identity manager.

Bidder: The user/bidder U_i with unique identity ID_i chooses a private key x_i and computes public key $y_i = g^{x_i}$ mod p. The user/bidder with valid key pair can bid for the goods.

Registration Manager: The registration manager(RM) with private key x_{RM} and public key $y_{RM} = g^{x_{RM}}$ mod p is responsible for registering each bidder and computes respective key pair for the bidders.

Auction Manager: The auction manager(AM) with private key x_{AM} and public key $y_{AM} = g^{x_{AM}}$ mod p is responsible for maintaining the goods information. The AM is also responsible for determining the winning bid and also opens it to other bidders to check the validity of the winning bid.

Identity Manager: The AM can only determine the winning bid without knowing the real identity of the winner. So, AM sends the winning bid to the RM and the RM can find the real identity of the winner, but RM takes more time for determining the winner's real identity. For reducing the winner identity determination time, AM sends winning bid and its information to IM. The IM with private key x_{IM} and public key $y_{IM} = g^{x_{IM}}$ mod p processes it and sends its corresponding information to RM. Finally, RM can determine the winner's identity in a short time.

4.6.2 Proposed e-auction protocol

Proposed blockchain-based e-auction protocol comprises of three phases: bidder registration phase, bidding phase and winner identification phase. Each phase of the proposed protocol is described as follows:

4.6.2.1 Bidder Registration Phase

Bidder U_i ($1 \leq i \leq$ m) secretly sends (ID_i, y_i) to all the registration manager RM_j's ($1 \leq j \leq$ n) for the registration. After receiving the registration request, each RM_j chooses a random integer k_{ij} such that $\gcd(k_{ij}, q) = 1$ where p and q are large prime numbers, and another random integer RN_{ij} for each U_i and computes certificate for the bidder as follows:

$$r_{ij} = y_i.k_{ij} - x_{RM} \; mod \; q \tag{4.14}$$

$$s_{ij} = y_i^{k_{ij}} \ mod \ p \tag{4.15}$$

Now, each RM_j sends their corresponding (r_{ij}, s_{ij}) pair to the bidder U_i. The bidder collects all the (r_{ij}, s_{ij}) pairs from all the group managers and computes his summarized pair as follows:

$$R_i = \sum_{j=1}^{n} r_{ij} \tag{4.16}$$

$$S_i = \prod_{j=1}^{n} s_{ij} \tag{4.17}$$

After computing (R_i, S_i) and $\sum_{j=1}^{n} RN_{ij}$, the bidder B_i can verify the validity of the certificate by the following equation:

$$S_i^{y_i} = (g^{R_i}.y_{RM})^{x_i} \ mod \ p \tag{4.18}$$

The certificate is valid for the bidder U_i if the above eq. 4.18 holds. The $\sum_{j=1}^{n} RN_{ij}$ is a linking value, and the RM_j can use it to reveal the real identity of the winning bidder. In the meantime, the RM_j stores the bidder's information as off-chain storage as the Table 4.1 shown below:

TABLE 4.1: The bidder's information at RM_j's off-chain storage

Identity	**Public key**	**Integer**	**Linking value**
ID_1	y_1	$\sum_{j=1}^{n} k_{1j}$	$\sum_{j=1}^{n} RN_{1j}$
ID_2	y_2	$\sum_{j=1}^{n} k_{2j}$	$\sum_{j=1}^{n} RN_{2j}$
ID_3	y_3	$\sum_{j=1}^{n} k_{3j}$	$\sum_{j=1}^{n} RN_{3j}$
.	.	.	.
.	.	.	.
.	.	.	.
ID_m	y_m	$\sum_{j=1}^{n} k_{mj}$	$\sum_{j=1}^{n} RN_{mj}$

4.6.2.2 Bidding Phase

If a bidder U_i wants to participate in the auction, then he needs to compute the following steps:

1. Bidder U_i sends his random number $\sum_{j=1}^{n} RN_{ij}$ and identity of the goods kept for auction GNO_i to IM.
2. IM selects a random integer d_i and computes $NO_i = GNO_i||d_i$
3. IM signs NO_i and RN_i using x_{IM} as $S = sign_{x_{AM}}[NO_i, RN_i]$. IM sends signature S and NO_i to the bidder.
4. IM maintains off-chain storage database for linking values as shown in Table 4.2.
5. The bidder can verify whether the $\sum_{j=1}^{n} RN_{ij}$ of the decryption is equal to the bidders $\sum_{j=1}^{n} RN_{ij}$ and this step protects anyone from modifying the NO_i.
6. The bidder computes $M = (GNO_i||T_i, NO_i, P_i)$, here P_i is the price of his bid, and T_i the timestamp.

TABLE 4.2: The linking value of bidders at IM's off-chain storage

Linking value	NO_i
$\sum_{j=1}^{n} RN_{1j}$	NO_1
$\sum_{j=1}^{n} RN_{2j}$	NO_2
$\sum_{j=1}^{n} RN_{3j}$	NO_3
.	.
.	.
.	.
$\sum_{j=1}^{n} RN_{mj}$	NO_m

7. Now, the bidder U_i chooses two random numbers N_1, N_2 in Z_q^* and computes signature of the bid as follows:

$$A = x_i.N_1.N_2 \ mod \ q \tag{4.19}$$

$$B = S_i^{N_1.N_2.y_i} \ mod \ p \tag{4.20}$$

$$C = M.y_{AM}^{-N_1.A.h(B)} \ mod \ p \tag{4.21}$$

$$D = N_1 - R_i.h(C) \ mod \ q \tag{4.22}$$

8. Finally, the bidder U_i sends his bid signature $\{A, B, C, D, GNO_i\}$ to the AM.

9. After receiving all the bids, the AM maintains auction information as Table 4.3 in his off-chain storage.

TABLE 4.3: The auction information table at AM's off-chain storage

User	**Signature**
U_1	$\{A_1, B_1, C_1, D_1, GNO_1\}$
U_2	$\{A_2, B_2, C_2, D_2, GNO_2\}$
U_3	$\{A_3, B_3, C_3, D_3, GNO_3\}$
.	.
.	.
.	.
U_m	$\{A_m, B_m, C_m, D_m, GNO_m\}$

4.6.2.3 Winner Identification Phase

After the end of the bidding process, AM, IM and RM will cooperate to find and publish the identity of the winner U_w as follows:

1. AM opens all the bids using the following equation:

$$M_i = C_i. \left[g^{D_i.A_i}. \prod_{j=1}^{n} y_{RM}^{-h(C_i).A_i}.B_i^{h(C_i)} \right]^{x_{AM}.h(B_i)} \ mod \ p. \tag{4.23}$$

After opening all the bids the AM finds the highest bid M_j by executing his smart contract and checks the validity of the bid with equation: $GNO_i = head(M_j, S)$.

2. AM selects a random number N_3 and computes: $Q_j = X_{RM}.N_3 mod q$ and $C'_j = M_j.(C_j.M'_j)^{N_3}$ mod p. Then, AM publishes $\{A_j, B_j, C'_j, D_j, GNO_i\}$ and Q_j such that anyone can verify the validity of the winning bid. Every winning bid satisfies the following equation:

$$M_j = C'_j. \left[g^{D_j.A_j}. \prod_{j=1}^{n} y_{RM}^{-h(C_j).A_j}.B_i j^{h(C_j)} \right]^{Q_j.h(B_j)} \quad mod\ p. \tag{4.24}$$

3. AM sends the winning bid $\{A_j, B_j, C_j, D_j, GNO_j\}$ and NO_j to IM. Then, IM finds the corresponding linking value $\sum_{j=1}^{n} RN_j$ of NO_j by looking into Table 4.2.

4. IM then sends the winning bid $\{A_j, B_j, C_j, D_j, GNO_j\}$ and linking value $\sum_{j=1}^{n} RN_{ij}$ to RM. RM then finds the corresponding ID_j, y_j and $\sum_{j=1}^{n} k_{ij}$ of $\sum_{j=1}^{n} RN_{ij}$ by looking into Table 4.1. Then RM checks whether $U_j = g^{A_j. \sum_{j=1}^{n} k_{ij}.y_j}$ mod p holds or not. If it holds, U_j that has identified ID_j is the winner.

Now, the RM_j sends the transaction details of winning bidder to the ordering service. The ordering service collects all such transactions to creates a new block. The new block will be sent to all the RM_j's. The RM_j's verifies the block and append it into their blockchain.

4.7 Comparison

In this section, comparison of proposed *DGSS* with other group signature schemes is done. In the literature almost all the group signature schemes have addressed different issues related to the group signature schemes but their group manager was left as a centralized party only. As per our knowledge, our *DGSS* addressed the issues of centralized group manager by decentralizing the managers role. The proposed *DGSS* holds all other security features even after decentralizing the group manager. Table 4.4 compares the

proposed *DGSS* with the corresponding existing group signature schemes against some of the security properties such as anonymity and unforgeability.

Scheme	**Year**	**Anonymity**	**Unforgeability**	**GM Decentralized**
Chaum [35]	1991	Yes	Yes	No
G Ateniese [14]	1999	Yes	Yes	No
CC Lee [92]	2009	Yes	Yes	No
Y Sakai [137]	2012	Yes	Yes	No
A Langlois [91]	2014	Yes	Yes	No
S Ling [100]	2015	Yes	Yes	No
B Libert [98]	2016	Yes	Yes	No
Q Alamélou [6]	2017	Yes	Yes	No
S Krenn[87]	2019	Yes	Yes	No
Q Luo[102]	2020	Yes	Yes	No
DGSS[48]	2021	Yes	Yes	**Yes**

TABLE 4.4: Comparison of group signature schemes

4.8 Conclusion

A novel DGSS is proposed to address the identity privacy challenges in blockchain based applications. Lee et al. [92] group signature scheme was extended to DGSS by decentralization of group manager to eliminate the basic requirement of having trust on the group manager and also to improve identity privacy of group members. The security properties like unforgeability, anonymity, unlinkability and traceability for the proposed DGSS are also discussed. The proposed DGSS is more suitable for permissioned blockchain-based sealed-bid e-auctions. Usage of anonymous signatures for public blockchains and its mathematical security is already explored in the previous chapter. Proof of correctness for proposed scheme ensures that the original message can still be reconstructed correctly, even after it is distributed among several group managers. Also a framework of blockchain-based sealed e-auction protocol with DGSS is proposed.

Chapter 5

Dynamic Decentralized Group Signature Scheme for Privacy Protection in Blockchain

In the previous chapter, a decentralized group signature scheme is proposed to protect user identity privacy in permissioned blockchain based applications. However, the proposed DGSS works only for sealed-bid e-auctions and static domains, where the network members are fixed and prevent new members from joining or current ones from leaving. The decentralized group signature scheme (DGSS) is expanded in this chapter to propose a dynamic decentralised group signature scheme (DDGSS) for open-bid electronic auctions that enables participants to join and leave at any moment. In order to guarantee user identity privacy on lightweight blockchains or memory-constrained devices, the efficiency of the DGSS is further enhanced by lowering the amount of multiplication operations.

This chapter is organized as follows: In the first section, problem statement is defined. The second section describes correctness and security definitions of group signature scheme. In third section, proposed dynamic decentralized group signature scheme is discussed. The fourth section describes proof of correctness and security analysis of the proposed scheme. Performance evaluation of the proposed dynamic decentralized group signature scheme is done in the fifth section. The chapter is concluded in sixth section.

5.1 Problem Statement

Various privacy-preserving techniques have been developed and are available in the literature to handle user identity privacy, transaction privacy, and other privacy challenges in blockchains. The currency mixing mechanism borrowed from the idea of Chaum [36] and the mixing services, was proposed to protect users' addresses from being linked. Mixing several unrelated user input and output addresses makes it difficult for outsiders to connect the transaction's input and output. A centralised coin mixing platform with a built-in audit feature Mixcoin is created by Bonneau et al. [26], which allows anonymous payment in Bitcoin and bitcoin-like coins. A CoinJoin scheme is proposed by Maxwell Gregory et al. [104], which is another method for anonymization of bitcoin transactions. The idea of joint payment was motivated by this scheme. If a person wants to make a payment, he can discover another user who wants to make a payment as well, and the two of them can make a combined payment in one transaction. However, in order to provide mixing services, many coin mixing schemes enlist the help of a trusted third party. A zerocoin scheme based on zero-knowledge proof (ZKP) is proposed by Miers et al. [110] to address the identity leakage problem of user. In this scheme, users can mask the addresses of both parties to a transaction using Zerocoin in this manner, rendering the transaction unlinkable. However, Zerocoin, on the other hand, can only exchange and issue fixed-value currencies, and the data in Zerocoin's ZKP is quite big, necessitating additional processing resources and blockchain storage.

The data in the blockchain is public and available to everyone. If user's private information from transaction is removed from the database then the privacy issue of the users is fundamentally resolved. From this idea, many off-chain payment schemes [44], [128], [111], [68] are proposed. However, all the existing off-chain transaction schemes implement anonymous transactions between users through third parties, resulting in the need for trust, which in return conflict with no-trust model of blockchains.

Group signature is an important technique that can play a role in providing identity privacy for signers in the blockchains. The group is made up of a few members and a group manager, and one of them can sign the message on behalf of the group anonymously. The group signature validator can only validate the signature created for the group; he or

she cannot determine the actual identity of the signer. This way, identity privacy can be achieved with help of group signatures. Only the group managers can add new members into the group and trace the actual signer of the message when the dispute arises or as per the requirement. The existing group signatures are classified into two types: (1) static group signature schemes (2) dynamic group signature schemes. Static group signature compute all the required parameters at the beginning and doesn't allow any new member into the group and revocation of existing members is also not possible. On the other hand, the dynamic group signature schemes add new members into the group at any time and members can also be revoked from the group anytime.

The security properties of group signature schemes like unforgeability, anonymity, unlinkability and traceability help to securely transact on the blockchain. To overcome the issues related to the centralized group manager in the existing group signature schemes, we proposed a static decentralized group signature scheme(DGSS) in 2021 [48] that is suitable for use in the private blockchain environment. But, due to the static nature of the scheme, it does not apply to all blockchain-based applications.

Key contribution of this chapter is that the static *DGSS* [48] is extended to propose a dynamic decentralized group signature scheme that allows to add new members into the network at any time, and also to revoke the members from the network. Also, our proposed scheme works for open-bid e-auctions which are deployed on private blockchains, where as *DGSS* works for closed-bid e-auctions which are deployed on private blockchains. This scheme can be utilized to protect the identity privacy of the signers in real-time distributed applications. In addition to that, the performance of the proposed scheme is better compared to our earlier scheme in terms of time complexity especially in scenarios where participants join and leave the network dynamically. This is achieved by reducing the number of multiplication operations in the verification algorithm of the proposed scheme.

5.2 Correctness and Security Definitions

In this paper, we adopt security definitions and the definition of the group signature scheme from the work of Bellare et al. [16].

Definition 5.1 (dynamic decentralized group signature scheme)

A dynamic decentralized group signature scheme $DDGSS$=(Init, Join, Sign, Verify, Identify, Revoke) is a collection of six polynomial-time algorithms defined as following:

5.2.1 Init

For all group members and managers, the initiation algorithm $Init$ will produce public and private key pairs (x_i, y_i).

5.2.2 Join

The group managers may run the interactive joining protocol to enrol new user as a member. The $Join$ algorithm will take random integer chosen by each group manager, the secret key of group manager and public key of the group members as an input and returns a certificate for the member.

5.2.3 Sign

The signing algorithm $Sign$ accepts a message M and two random integers N_1, N_2 as input and outputs a group signature as $\{A, B, C, D, M\}$.

5.2.4 Verify

The verification algorithm $Verify$ takes a combination of group signature, message M along with the collision-resistant hash function $h()$ as input and returns $true$ if the signature is valid, otherwise returns $false$.

5.2.5 Identify

The identification algorithm $Identify$ takes a combination of group signature $DDGS$, message M along with the collision-resistant hash function $h()$ as input, it is processed by all the group managers and returns the public key y_i of the actual signer.

5.2.6 Revoke

To revoke any group member, the group managers run the *Revoke* algorithm.

Definition 5.2 (Correctness)

Always accept the signature of an honest group member i.e., the Verify(.) algorithm should return 1. The Identify(.) algorithm should always identify the real signer of the message for any given valid message and group signature.

Definition 5.3 (Unforgeability)

It is computationally difficult for any unauthorized member to produce a valid signature on behalf of the group. Only authorized members of the group can generate a valid signature on behalf of the entire group.

Definition 5.4 (Anonymity)

No one can tell who signed the message for a given valid group signature since it's computationally difficult.

Definition 5.5 (Unlinkability)

It is computationally difficult to determine if the two legitimate group signatures were generated by the same person.

Definition 5.6 (Traceability)

It is computationally difficult for anyone except the group managers to trace the identity of the actual signer. If there is any dispute among the group members or as per requirement, all the group managers together can identify the actual signer.

5.3 Proposed Dynamic Decentralized Group Signature Scheme

The proposed dynamic decentralized group signature scheme (DDGSS) consists of six polynomial-time algorithms: $Init, Join, Sign, Verify, IdentifyandRevoke$. The DDGSS is described as follows:

5.3.1 Init Algorithm

Let p and q be two large prime numbers such that q | p - 1, and g be a generator with order q in GF(p). Each group member U_i ($1 \leq i \leq$ m) selects the private key x_i and computes the public key $y_i{=}g^{x_i}\ mod\ p$. Each group manager T_j ($1 \leq j \leq$ n) selects his private key x'_j and computes the public key $y'_j{=}g^{x'_j}\ mod\ p$.

5.3.2 Join Algorithm

In the proposed DDGSS, when a group member computes his public/private key pair, he will send his public key and identity information to the all the Group Managers (GMs) for registration. Every user U_i sends his public information (y_i, ID_i) to the Group Managers for registration using a secure channel. All the GMs will add the public information of U_i as an item in Public Key State List (PKSL). The PKSL stores the information about existing and revoked members of the group. In the table, $T_{i-start}$ indicates the joining time and T_{i-end} indicates the revocation time of the $i^{th} - member$. The PKSL structure for the member U_i is shown below:

S.No	**Public Key**	**Identity Information**	**Time Start**	**Time End**
i	y_i	ID_i	$T_{i-start}$	T_{i-end}

After the registration is done, the GMs will start generating the membership pair for each U_i. For each U_i, each T_j can randomly choose an integer k_{ij} in Z_q^* and computes

$$r_{ij} = (y_i.k_{ij} - x'_j) \; mod \; q \tag{5.1}$$

$$s_{ij} = y_i^{k_{ij}} \; mod \; p \tag{5.2}$$

Now, each group manager T_j sends (r_{ij}, s_{ij}) pair to the group member U_i. After receiving(r_{ij}, s_{ij}) pairs from all the group managers, the group member U_i computes the certificate (R_i,S_i) as follows:

$$R_i = \sum_{j=1}^{n} r_{ij} \tag{5.3}$$

$$S_i = \prod_{j=1}^{n} s_{ij} \tag{5.4}$$

Next, after computing (R_i,S_i), U_i can check the correctness of the certificate by verifying the following equation:

$$S_i^{y_i} \; mod \;\; p \; = (g^{R_i}.\prod_{j=1}^{n} y'_j)^{x_i} \; mod \; p \tag{5.5}$$

5.3.3 Sign Algorithm

In our proposed DDGSS, any group member U_i can sign the message M using the following steps. Here, h() is a collision-resistant hash function and || denotes a concatenation.

1. Choose any two random integers N_1, N_2 in Z_q^*
2. Compute A, B, C and D parameters as follows:

$$A = x_i.N_1.N_2 \; mod \; q \tag{5.6}$$

$$B = h^{-1}(M||A||D).g^{-N1.A.h(M||A||D)} \ mod \ p \tag{5.7}$$

$$C = g^{N_1 - R_i.h(B)} \ mod \ q \tag{5.8}$$

$$D = S_i^{N_1.N_2.y_i} \ mod \ p \tag{5.9}$$

3. *DDGS* for the message M is {A, B, C, D, M}.

5.3.4 Verify Algorithm

The verification of the generated signature can be done using the following equation. The group signature is valid iff the following equation holds.

$$[B.h(M||A||D)]^{-1} = \left[C^A \left(\prod_{j=1}^{n} y_j'^{-A} D \right)^{h(B)} \right]^{h(M||A||D)} \tag{5.10}$$

5.3.5 Identify Algorithm

The real signer of the proposed group signature has to be revealed as per the requirement. Each group manager has access to (y_i, k_{ij}). Hence, all the group managers together can acquire to the $(y_i, \sum_{j=1}^{n} k_{ij})$ of group member U_i and it is required to satisfy the following equation:

$$D == g^{A.y_i. \sum_{j=1}^{n} k_{ij}} \ mod \ p \tag{5.11}$$

Here, $i = 1, 2, 3, ...m$, where m is the number of group members.

5.3.6 Revoke Algorithm

In order to revoke any group member U_i from the group, the Group Managers will modify the Time-End (T_{i-end}) of the member in $PKSL$. The revoked member could not run the $sign()$ algorithm again. However, the earlier signatures of the U_i are still valid.

5.4 Proof of Correctness and Security Analysis

The security analysis and proof of correctness of the proposed *DDGSS* is discussed in this section. The proposed *DDGSS* is holding all the security features of Devidas et al.'s [48] decentralized group signature scheme even after extending it to the dynamic setting and is also reducing the number of multiplication operations in the verification phase.

5.4.1 Correctness of the proposed scheme

5.4.1.1 Correctness of Certificate

After computing (R_i, S_i) pair the U_i can verify the certificate's validity as follows:

$$S^{y_i} \ mod \ p = \left(g^{R_i} . \prod_{j=1}^{n} y'_j \right)^{x_i} \ mod \ p \qquad \text{(from (5.5))}$$

$$= \left(g^{\sum_{j=1}^{n} r_{ij}} . \prod_{j=1}^{n} y'_j \right)^{x_i} \ mod \ p \qquad \text{(from (5.3))}$$

$$= \left(g^{\sum_{j=1}^{n} (y_i . k_{ij} - x'_j)} . \prod_{j=1}^{n} y'_j \right)^{x_i} \ mod \ p \qquad \text{(from (5.1))}$$

$$= \left(g^{y_i . \sum_{j=1}^{n} k_{ij} - \sum_{j=1}^{n} x'_j} . \prod_{j=1}^{n} g^{x'_j} \right)^{x_i} \ mod \ p$$

$$= \left(g^{y_i . \sum_{j=1}^{n} k_{ij} - \sum_{j=1}^{n} x'_j} . g^{\sum_{j=1}^{n} x'_j} \right)^{x_i} \ mod \ p$$

$$= \left(g^{x_i . y_i . \sum_{j=1}^{n} k_{ij}} \right) \ mod \ p$$

$$= \left(g^{x_i . \sum_{j=1}^{n} k_{ij}} \right)^{y_i} \ mod \ p$$

$$= \left(\prod_{j=1}^{n} g^{x_i . k_{ij}} \right)^{y_i} \ mod \ p$$

$$= \left(\prod_{j=1}^{n} y_i^{k_{ij}}\right)^{y_i} \ mod\ p$$

$$= \left(\prod_{j=1}^{n} s_{ij}\right)^{y_i} \ mod\ p \qquad \text{(from (5.2))}$$

$$= S^{y_i}$$

5.4.1.2 Correctness of Signature Verification

The correctness of the generated signature can be verified as follows:

$$[B.h(M||A||D)]^{-1} = \left[C^A \left(\prod_{j=1}^{n} y_j'^{-A} D\right)^{h(B)}\right]^{h(M||A||D)}$$

$$RHS = \left[C^A \left(\prod_{j=1}^{n} y_j'^{-A} D\right)^{h(B)}\right]^{h(M||A||D)}$$

$$= \left[\left(g^{N_1 - R_i.h(B)}\right)^A \left(\prod_{j=1}^{n} y_j'^{-A}.S_i^{N_1.N_2.y_i}\right)^{h(B)}\right]^{h(M||A||D)} \qquad \text{(from (5.8 and 5.9))}$$

$$= \left[\left(g^{N_1 - R_i.h(B)}\right)^A \left(\prod_{j=1}^{n} y_j'^{-A}.\left(g^{R_i}\prod_{j=1}^{n} y_j'\right)^{x_i.N_1.N_2}\right)^{h(B)}\right]^{h(M||A||D)} \qquad \text{(from (5.5))}$$

$$= \left[\left(g^{N_1 - R_i.h(B)}\right)^A \left(\prod_{j=1}^{n} y_j'^{-A}.\left(g^{R_i}\prod_{j=1}^{n} y_j'\right)^{A}\right)^{h(B)}\right]^{h(M||A||D)} \qquad \text{(from (5.6))}$$

$$= \left[\left(g^{N_1 - R_i.h(B)}\right).g^{R_i.h(B).A}\right]^{h(M||A||D)}$$

$$= g^{N_1.A.h(M||A||D)}$$

$$= \left[g^{-N_1.A.h(M||A||D)}\right]^{-1}$$

$$= \left[h^{-1}(M||A||D).g^{-N_1.A.h(M||A||D)}.h(M||A||D)\right]^{-1}$$

$$= [B.h(M||A||D)]^{-1}$$

$$= LHS$$

From the above simplification, the eq. 5.10 holds. Hence, the given verification algorithm is valid.

5.4.2 Security Analysis

The security of proposed $DDGSS$ is based on the hardness assumption of the discrete logarithm problem (DLP). The proposed $DDGSS$ satisfies all the security properties discussed in 5.2 as follows:

5.4.2.1 Unforgeability:

Any attacker can generate a valid group signature if and only if he knows a valid certificate (R_i, S_i) and private key x_i. In case, if attacker has a valid certificate (R_i, S_i), he has to compute A, B, C, D by eq. 5.6, 5.7, 5.8 and 5.9. Without the secret key x_i and N_1, N_2, it is not feasible to forge the group signature. N_1, N_2 are random values and since $y_i = g^{x_i}$, solving x_i reduces to solving DLP.

5.4.2.2 Anonymity:

For a valid group signature $\{A, B, C, D, M\}$, identifying the true signer is difficult for anyone except the group managers. All confidential information is protected by a set of randomly generated numbers N_1, N_2. In the group signature $\{A, B, C, D, M\}$, only A and D parameters contain the actual identity information of signer. Hence, the scheme should be examined whether it has anonymity by A *and* D or not.

*Attack*1 : For a valid group signature $\{A, B, C, D, M\}$ and the equation $A = x_i.N_1.N_2 \ mod \ q$, one can compute that

$$g^A = g^{x_i.N_1.N_2} \ mod \ p$$

$$= y_i^{N_1.N_2} \ mod \ p$$

In the above equation, N_1, N_2 are random integers and if the attacker have N_1, N_2 then only he can compute y_i to identify the actual signer. Since, the random integers N_1, N_2 are unknown, no one can find the real signer. i.e., the proposed group signature scheme has anonymity by A.

*Attack*2 : For a valid group signature $\{A, B, C, D, M\}$ and the equation $D = S_i^{N_1.N_2.y_i} \ mod\ p$, one can compute that

$$S_i^{N_1.N_2.y_i} = (g^{R_i}. \prod_{j=1}^{n} y'_j)^{x_i.N_1.N_2} \ mod\ p \qquad \text{(from (5.5))}$$

$$= (g^{\sum_{j=1}^{n} r_{ij}}. \prod_{j=1}^{n} g^{x'_j})^{x_i.N_1.N_2} \ mod\ p \qquad \text{(from (5.3))}$$

$$= (g^{\sum_{j=1}^{n} (y_i.kij - x'_j)}.g^{\sum_{j=1}^{n} x'_j})^{x_i.N_1.N_2} \ mod\ p \qquad \text{(from (5.1))}$$

$$= g^{\sum_{j=1}^{n} kij.y_i.x_i.N_1.N_2} \ mod\ p$$

$$= y_i^{\sum_{j=1}^{n} k_{ij}.N_1.N_2.y_i} \ \text{mod p}$$

If the attacker has $\sum_{j=1}^{n} k_{ij}$, N_1, N_2 then he can compute y_i and can find the actual signer's identity. But $\sum_{j=1}^{n} k_{ij}$, N_1, N_2 are all unknown random values and hence no one can find out the actual signer. So, the proposed $DDGSS$ has anonymity by D. Because of anonymity of A and D, proposed $DDGSS$ has anonymity by B and C respectively by Eq. 7 and Eq. 8. Hence, entire group signature $\{A, B, C, D, M\}$ has anonymity.

5.4.2.3 Unlinkability:

5.4.2.4 Lemma:

To determine whether the two group signatures {A, B, C, D, M} and $\{A', B', C', D', M'\}$ are created by the same user or not, the following equation should hold,

$$\frac{D}{D'} = \left(\frac{g^A}{g^{A'}}\right)^{\sum_{j=1}^{n} k_{ij}.y_i} \quad mod\ p \tag{5.12}$$

5.4.2.5 Corollary:

It is computationally infeasible to determine that two group signatures were generated by the same user.

5.4.2.6 Proof:

$$\frac{D}{D'} = \frac{S_i^{N_1.N_2.y_i}}{S_i^{N'_1.N'_2.y_i}}\ mod\ p \qquad \text{(from (5.9))}$$

$$= \frac{\left(\prod_{j=1}^{n} s_{ij}\right)^{N_1.N_2.y_i}}{\left(\prod_{j=1}^{n} s_{ij}\right)^{N'_1.N'_2.y_i}}\ mod\ p \qquad \text{(from (5.4))}$$

$$= \frac{\left(\prod_{j=1}^{n} {y_i}^{k_{ij}}\right)^{N_1.N_2.y_i}}{\left(\prod_{j=1}^{n} {y_i}^{k_{ij}}\right)^{N'_1.N'_2.y_i}}\ mod\ p \qquad \text{(from (5.2))}$$

$$= \frac{\left({y_i}^{\sum_{j=1}^{n} k_{ij}}\right)^{N_1.N_2.y_i}}{\left({y_i}^{\sum_{j=1}^{n} k_{ij}}\right)^{N'_1.N'_2.y_i}}\ mod\ p$$

$$= \frac{\left(g^{x_i \sum_{j=1}^{n} k_{ij}}\right)^{N_1.N_2.y_i}}{\left(g^{x_i \sum_{j=1}^{n} k_{ij}}\right)^{N'_1.N'_2.y_i}}\ mod\ p$$

$$= \frac{\left(g^{x_i.N_1.N_2}\right)^{\sum_{j=1}^{n} k_{ij}.y_i}}{\left(g^{x_i.N'_1.N'_2}\right)^{\sum_{j=1}^{n} k_{ij}.y_i}}\ mod\ p$$

$$= \left(\frac{g^A}{g^{A'}}\right)^{\sum_{j=1}^{n} k_{ij}.y_i} \ mod\ p \qquad \text{(from eq.6)}$$

Corollary stands true because an attacker doesn't have knowledge of random numbers k_{ij}'s of other group managers to compute $\sum_{j=1}^{n} k_{ij}.y_i$ and solving Eq. 5.12 is equivalent to DLP hard problem along with unknown random parameter k_{ij}.

5.4.2.7 Traceability:

All the group managers T_j $(1 \leq j \leq \text{n})$ can access $(y_i, \sum_{j=1}^{n} k_{ij})$ of all the group members U_i $(1 \leq j \leq \text{m})$. So, they can access (y_i, k_{ij}) of U_i satisfying the equation $B = g^{A.\sum_{j=1}^{n} k_{ij}.y_i}\ mod\ p$. Here, $i = 1, 2, ..., m$, where m indicates the total number of group members. So, the set of group managers together can identify the actual signer, thereby making the proposed *DDGSS* traceable if required.

5.5 Performance Evaluation

The efficiency of the proposed scheme is compared with *DGSS* [48] which is proposed in the previous chapter. Computation of the time complexity is basically employed for the evaluation of performance of the proposed scheme. The notations used to evaluate the performance are:

- T_h - time required for executing a one-way hash function $h()$.
- T_{exp} - time required for executing a modular exponentiation operation.
- T_{Nmul} - time required for multiplication with modular N.

Signature Scheme	**Static or Dynamic**	**Signature Generation**	**Signature Verification**	**Total**
DGSS [48]	Static	$2T_{exp}$+$8T_{Nmul}$+$2T_h$	$4T_{exp}$+$6T_{Nmul}$+$2T_h$	$6T_{exp}$+$14T_{Nmul}$+$4T_h$
Proposed DDGSS	Dynamic	$3T_{exp}$+$8T_{Nmul}$+$2T_h$	$3T_{exp}$+$4T_{Nmul}$+$2T_h$	$6T_{exp}$+$12T_{Nmul}$+$4T_h$

Both DGSS and the proposed scheme are based on discrete logarithm problem only. In DGSS, the signer needs the cost of $2T_{exp}$+$8T_{Nmul}$+$2T_h$ and signature verification needs the cost of $4T_{exp}$+$6T_{Nmul}$+$2T_h$. In the proposed scheme, the signer needs cost of $3T_{exp}$+$8T_{Nmul}$+$2T_h$ and signature verification needs cost of $3T_{exp}$+$8T_{Nmul}$+$2T_h$. Hence, the DGSS needs the total cost of $6T_{exp}$+$14T_{Nmul}$+$4T_h$ and the DDGSS needs the total cost of $6T_{exp}$+$12T_{Nmul}$+$4T_h$. Compared with DGSS, the proposed scheme is better in terms of performance, which makes our scheme suitable for memory constrained domains and light-weight blockchains.

5.6 Comparison

In this section, comparison of proposed *DDGSS* with other dynamic group signature schemes is done. In the literature almost all the dynamic group signature schemes have addressed different security issues but their group manager was left as a centralized party only. As per our knowledge, our *DDGSS* addressed the issues of centralized group manager by decentralizing the managers role. The proposed *DDGSS* holds all other security features even after decentralizing the group manager. Table 5.1 compares the proposed *DDGSS* with the corresponding existing dynamic group signature schemes against some of the security properties such as anonymity and unforgeability.

Scheme	**Year**	**Anonymity**	**Unforgeability**	**GM Decentralized**
M Bellare [17]	2005	Yes	Yes	No
X Zho [178]	2007	Yes	Yes	No
JY Hwang [74]	2015	Yes	Yes	No
B Libert [98]	2016	Yes	Yes	No
MNS Perera [125]	2018	Yes	Yes	No
Y Sun [151]	2019	Yes	Yes	No
J Camenisch [30]	2020	Yes	Yes	No
Y Sun [150]	2021	Yes	Yes	No
H Kim [81]	2021	Yes	Yes	No
DDGSS [48]	2022	Yes	Yes	**Yes**

TABLE 5.1: Comparison of dynamic group signature schemes

5.7 Conclusion

A dynamic decentralized group signature scheme is proposed to address the user identity privacy issues in blockchain-based applications. In the previous chapter, a decentralized group signature scheme is proposed by decentralizing the group manager but it is suitable for static environments only. Hence, it does not allow new members to join at run-time and the group members cannot be revoked. In this chapter, the DGSS is extended as a dynamic decentralized group signature scheme for open-bid e-auctions. The proposed scheme can join new members in the group at any time and also revoke the group members. The performance of our scheme is more efficient compared to DGSS. This is achieved by reducing multiplication operations of verification algorithm. The proposed scheme is suitable for both blockchain based applications of dynamic setting and as well for the memory constrained devices and light-weight blockchains. The security properties like unforgeability, anonymity, unlinkability and traceability for the proposed scheme are also discussed. The suggested scheme is more suited to permissioned blockchain-based open-bid e-auctions. The proof of correctness for the proposed scheme ensures that the original message can still be reconstructed correctly, even after it has been distributed among several group managers.

Chapter 6

A Scalable Decentralized Framework for Record Keeping Systems using Hyper Ledger Fabric

In chapter 3, an identity verifiable ring signature scheme for privacy protection in blockchains is proposed. This scheme is applicable for e-auction protocols which are deployed on public blockchains. In the chapter4, a decentralized group signature scheme is proposed to protect user identity privacy in private blockchain-based sealed-bid e-auctions. A dynamic decentralized group signature scheme is proposed to address identity privacy issue in private blockchain-based open-bid e-auctions in chapter5. In this chapter, a decentralized framework for record-keeping system in Government using hyperledger fabric with enhanced performance are discussed. The schemes proposed in the chapter4 and chapter5, can be deployed on this proposed decentralized framework.

This chapter's flow is as follows: The problem statement is discussed in the first section. The second section describes preliminaries and detailed overview of the Hyperledger Fabric (HLF). The proposed decentralized framework for record-keeping system in Government

using hyperledger fabric is described in third section. The chapter is concluded in the final section.

6.1 Problem Statement

For any government, well-managed records are the foundation to preserve verifiability, immutability, transparency, resilience and collaboration. Records that are transparently managed can be used to evaluate how well the government is performing. Records can protect the interests and rights of the people, and hold officials accountable for their actions in the future. Government records must be kept safe while guaranteeing the participants to a transaction that have full privacy and confidentiality, and that data can only be accessed by those who need to know. But saying it is easier than doing it. Centralized databases are vulnerable to costly security breaches and they have a single point of failure. Blockchain technology can solve all the above said problems and can be used as record-keeping system in Government.

Hyperledger Fabric is one of the best blockchains for providing a modular and secure industrial blockchain platform. The permissioned blockchain, hyperledger fabric is more popular in commercial sector for maintaining records. However, various issues are preventing industries from fully adopting this blockchain. HLF is designed to be scalable, but like other blockchain networks it has few scalability issues. These issues can impact the performance of the blockchain network as the number of nodes and transactions increases. The execution of a smart contract may need a lot of resources, and as the number of transactions on the network grows, this can increase the execution time, which may affect the smart contract's scalability. One of the above said issues is scalability [83]. Scalability refers to a system's capacity to handle an increasing quantity of work while remaining stable [177]. According to the literature, organisations need to improve their capacity to handle an increased amount of transactions or workload [153].

The notion of scalability can be defined as [117]: (1) Horizontal: it is accomplished by expanding the current network with more machines or (2) Vertical: It is accomplished by improving performance of the pool of resources that are already available. In this thesis,

our focus is on the vertical scalability. In general, performance enhancement of hyperledger Fabric fall into two categories [40]: (i) architectural redesign and (ii) bottleneck reduction. In this chapter, we have adopted the first approach and redesigned the hyperledger fabric architecture to improve its performance and a decentralised framework is proposed for record keeping in government organisations to address scalability issue.

6.2 Preliminaries

In this section, all the preliminaries used in proposing a decentralized framework for record-keeping system in Government using HLF are discussed.

6.2.1 Hyperledger Fabric

The order-execute architecture [9] is followed by the public blockchains like bitcoin, ethereum etc. and permissioned blockchains like Tendermint, Chain or Quorum etc. This implies that a consensus mechanism to order the transactions will be used by the blockchain network first.The order-execute approach have some drawbacks, such as sequential execution across all peers, deterministic code only, and difficulties in achieving execution confidentiality. To address the problems of order-execute blockchain platforms, the Linux Foundation released HLF [9], an open source permissioned blockchain platform, in 2015. Hyperledger Fabric is a blockchain platform developed by a consortium. The HLF supports general-purpose programming languages like Java and Go to implement chaincode and it is modular in design. To do the above one, the execute-order-validate [9] architecture is used. A transaction submitted by client will be simulated by peers in the execution phase and read-write set will be generated as an endorsement outcome. The client then sends the approved transactions to an order service.In ordering phase, the ordering service collects all the approved transactions that are organized into blocks. Peers validate the blocks in the validation phase to update the ledger state. It verifies transaction requirements and transaction integrity. The hyperledger fabric is having different types of components as follows and all the below definitions are taken from [9]:

6.2.1.1 Nodes

A blockchain is made up of a number of nodes that communicate with one another to execute transactions. Since the hyperledger fabric is a permissioned network, each node is assigned a unique identity by the membership service provider (MSP). The hyperledger fabric is having three types of nodes; orderers, peers, and clients. Peers are nodes that execute transactions and keep track of them on the ledger. By default, all peers will be the committers since they receive ordered block from ordering service and maintain it in their ledger. The peers may have an additional duty of endorser. The transactions are ordered by orderers. Finally, the end-users will function as clients, sending transaction requests to peers.

6.2.1.2 Membership Service Provider (MSP)

The MSP is responsible for maintains the identities of all nodes (clients, peers, and orderers) in the network and granting credentials to the nodes for authorization and authentication. Because HLF is permissioned, all communication between nodes takes place through messages that are validated, usually using digital signatures. All nodes, including peers, must recognize the authentications and the same identities as legitimate, according to the MSP configuration.

6.2.1.3 Chaincode

A chaincode (smart contract) is a program in which application logic will be implemented and it has to be executed during the execution phase. The chaincode is the heart of a hyperledger fabric blockchain, and it will be invoked by the endorsing nodes. System chaincodes are special chaincodes that are used to manage the blockchain system and maintain parameters.

6.2.1.4 Endorsing Policy

An endorsement policy will be evaluated in the validation phase. Any untrusted party cannot change the endorsement policies. A fabric endorsement policy is often created using the chaincode and acts as a static library for transaction validation. Only designated administrators may use system management capabilities to change endorsement policies. A common endorsement policy allows the chaincode to specify the endorsers for a transaction in the form of a set of required peers; it employs a monotone logical expression on sets like three out of five or $(A \wedge B) \vee C$. Custom endorsement policies can use any logic they choose.

6.2.1.5 Ordering Service

In HLF, an ordering service is responsible for execution of transaction's ordering, creating new blocks, and disseminating the new blocks to all blockchain peers. Solo [9], Kafka [88], and Raft [121] are the three types of ordering services now available. A single node will be there in the solo ordering service. Because a single fault in Solo cannot be tolerated, it is suggested that it be used strictly for experiments. Both Raft and Kafka ordering services use a "leader and follower" node setup to allow multiple nodes to tolerate crash faults. The leader node orders transactions, while follower nodes replicate them. Raft ordering is often more simple and efficient than Kafka ordering.

6.2.1.6 Channels

HLF Channels are logical entities that represent a grouping of two or more Blockchain network members/participants for the purpose of executing private and secret transactions. The following are the essential components of a channel:

- Members of the organisation.
- One or more anchor peers per member organization.
- Channel policies.

- Shared ledgers which are private to channel members.
- Chaincode.
- Ordering service nodes.

6.2.1.7 Organizations

The HLF network is made up of peers that are owned and contributed by many organisations in the network. The network exists because organisations contribute their individual resources to the common network. Peers have a digital certificate issued to them by their owning organization's MSP. Peers from various organisations might be there on the same channel.

6.2.2 System Overview

Consider the tiny network depicted in Figure 6.1, in which three organisations O_1, O_2, and O_3, collaborate to create a distributed ledger. Every organisation may be considered as a validating peer. The network initiator is chosen from the validating peers (O_1). The clients submit transaction requests using multiple channels (C_1, C_2, C_3) to validating peers. Validating peers verify the transaction before broadcasting it onto the network. A copy of the ledger L_1 corresponding with channel C_1 is stored on peer node P_1. A copy of ledger L_2 associated with channel C_2 is maintained by peer node P_2. P_3 holds a copy of the ledger L_3 that is associated to channel C_3. Channel C_1 is controlled by the organizations O_1 and O_2 and is configured with chaincode CC_1. Channel $C2$ is controlled by the organizations O_2 and O_3 and is configured with chaincode CC_2. There is an ordering service used by all the organizations. Each organisation has a chosen Certificate Authority (CA) that will issue certificates to its peers.

6.2.3 Transaction Flow

The HLF is a permissioned blockchain in which participants must have certain credentials and are referred to as peers. An MSP is responsible for associating all network peers

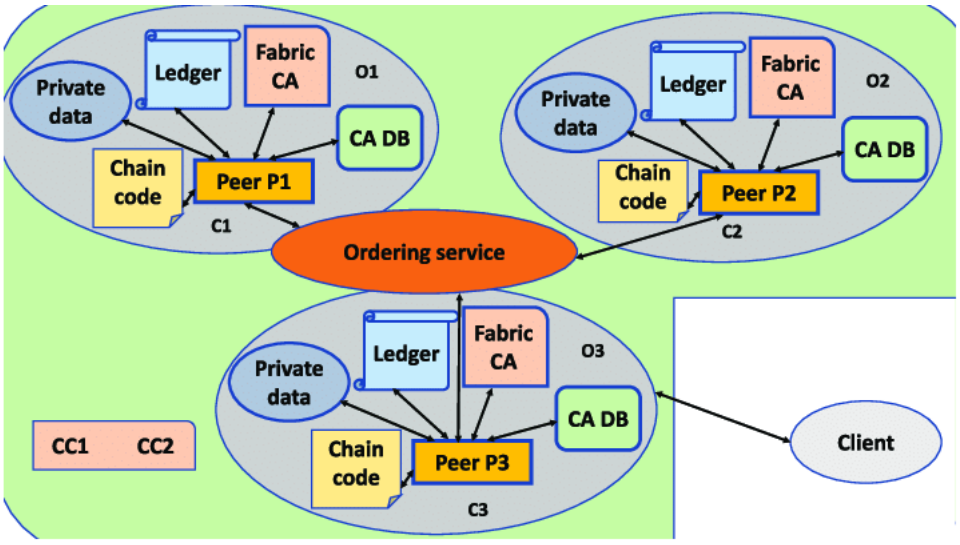


FIGURE 6.1: Hyperledger Fabric System Overview [153]

by giving cryptographic credentials, ensuring that the Hyperledger Fabric's permissioned nature is maintained. A given system's MSP can be any standard certificate authority. Otherwise, the system can create its own CA. The Hyperledger Fabric transaction life cycle is as follows, shown in Figure 6.2.

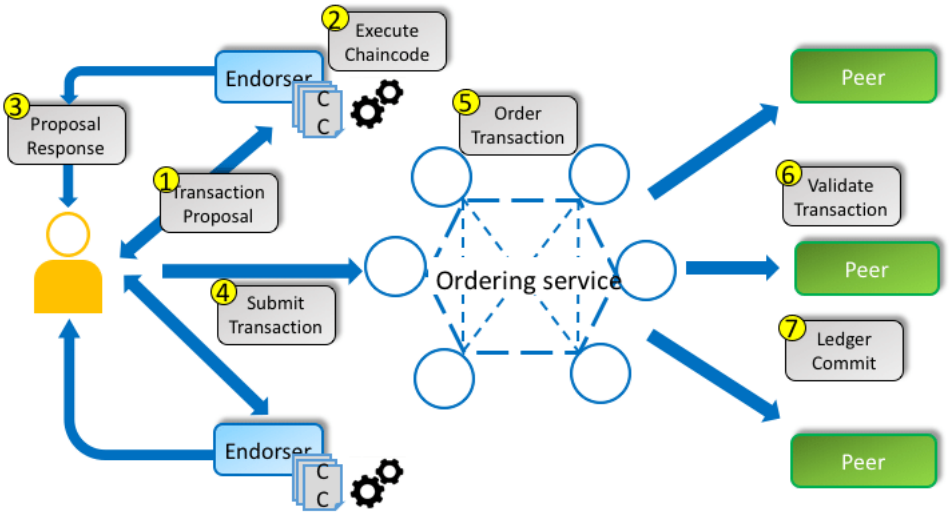


FIGURE 6.2: Hyperledger Fabric Transaction Flow [9]

1. A transaction proposal is prepared by the client and submitted to the endorsing peers for approval.

2. Endorsing peers check the client's authentication and proposal format, run the chaincode associated with them, and prepare the read/write set for the corresponding proposal.

3. Endorsers sign a read/write set and return it to the client as a response to the proposal.

4. The client now sends the endorsed proposal to the ordering peers, along with the read/write set.

5. The ordering peers then reach consensus among themselves by running a consensus algorithm that they have configured, forming a block of transactions, and submitting it to the committing peers.

6. Committing peers verifies the endorsing policy and read/write set versions in state DB, tags transactions as valid or invalid, commits all valid transactions to state DB, and broadcasts the block to all peers to add to their blockchain.

6.3 Proposed Framework

The scalability plays a vital to the success and acceptance of any blockchain technology, it is imperative to provide a scalable blockchain framework. A scalable blockchain framework can manage a significant amount of transactions and users while preserving very high levels of security, privacy, and decentralization.

In this section, a decentralized scalable framework for using HLF is proposed. Three main phases are proposed in our framework – Endorsing phase, Validation phase and Ledger updating phase. Fig.3 depicts the architecture of the proposed framework. Endorsing phase endorses the transaction proposal, checks the endorsing policy and sends the proposal response to the validator. The Validating peer checks the proposal, sends the acknowledgement to the user, collects the transactions into a block and broadcast the block to all the peers to update their ledgers. Every communication in the system is happening through the TLS protocol to enforce anonymity and security. The network has n number of peers in the system and has l endorsers and m validators ($l, m \leq n$).

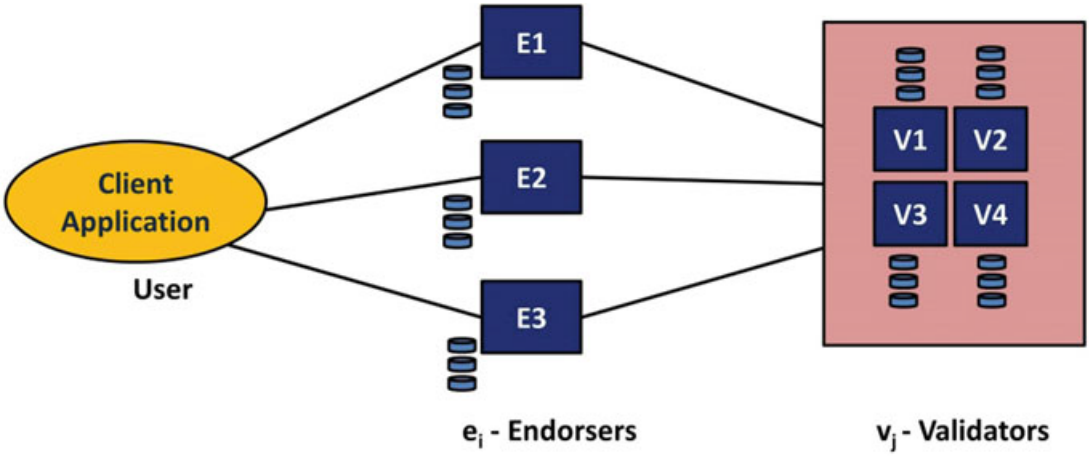


FIGURE 6.3: Architecture of Proposed Framework

6.3.1 Endorsing Phase

In this phase client prepares a transaction proposal T_x and signs the proposal using his secret key u_{sk}, gets the signed proposal σ. Depending on the endorsing policy user may submit the proposal to more than one endorser also. After receiving the proposal, endorser checks the legitimacy of the user by verifying his signature σ. If the user is legitimate, then the endorser executes the proposal on the corresponding chaincode CC, signs the proposal using his secrete key e_{sk}, send it to the validator as a proposal response (σ') and sends the acknowledgement to the corresponding user.

Algorithm 9 Endorsement Algorithm

Require: $T_x,\ \sigma$

Ensure: σ'

1: $res1 \Leftarrow Verify(u_{pk}, T_x, \sigma)$
2: $res2 \Leftarrow CCExec(CC, T_x)$
3: **if** $res1 == true\ and\ res2 == true$ **then**
4: $\quad \sigma' \Leftarrow SigGen(\sigma, e_{pk})$
5: **else**
6: $\quad return\ 0$
7: **end if**

6.3.2 Validation Phase

In this phase, the endorsers $e_i(1 \leq i \leq l)$ submit endorsed proposal to the validator $v_j(1 \leq j \leq m)$ for validation and approval of the proposal to include in the distributed ledger. After receiving the proposal, validator verifies the signatures of both user and endorser and also verifies the endorsing policy specified. Endorsing policy is formulated by the administrator at the time of configuring the system. For instance, if an organization has three endorsing peers A, B, *and* C, the user must be endorsed by all three of them in order to meet the company's endorsing policy. If the validator believes the endorsers are trustworthy and the proposal meets the endorsing policy, the validator marks the transaction proposal or record as valid and considers it for inclusion in the blockchain; if not, the validator marks it as invalid and sends it back to the user with a reasoning. Now all the validators have to achieve the consensus to order the transactions into a block, which comprises of all the records marked as valid. The Kafka mechanism has to be configured at all the validators to achieve the consensus.

Algorithm 10 Validation Algorithm

Require: EP, σ, σ'

Ensure: b *or* ack

1: $res1 \Leftarrow Verify(\sigma, \sigma', e_{pk}, u_{pk})$
2: $res2 \Leftarrow ePolicy(EP)$
3: **if** $res1 == true$ *and* $res2 == true$ **then**
4: $\quad b \Leftarrow createBlock(T_x)$
5: $\quad return\ b$
6: **else**
7: $\quad return\ ack$
8: **end if**

6.3.3 Ledger Updating Phase

Once the consensus is achieved by all the validators $v_j(1 \leq j \leq m)$ and block b is created for all the valid transactions, then b is broadcasted to all the peers $P_k(1 \leq k \leq n)$ of

the particular channel to update their local copy of global data B. Now all the peers p_k update their consistent distributed ledger B to B'.

Algorithm 11 Ledger Updation Algorithm

Require: b
Ensure: B'
1: $B' \Leftarrow append(B, b)$
2: $return\ B'$

6.3.4 Transaction flow of proposed framework

In this section, transaction flow of proposed decentralized framework for record-keeping system in Government using HLF is described. The diagram below depicts the transaction flow of proposed framework.

1. A transaction proposal is prepared by the client and submitted to the endorsing peers for approval.

2. Endorsing peers check the client's authentication and proposal format, run the chain-code on the proposal and prepare the read/write set for the corresponding proposal.

3. Endorsers sign a read/write set and send it to the ordering service as a response to the proposal instead of returning it back to the client.

4. The ordering peers then reach consensus among themselves by running a consensus algorithm that they have configured, forming a block of only valid transactions, and submitting it to the committing peers.

5. Committing peers verifies the endorsing policy and read/write set versions in state DB, commits all valid transactions to state DB, and broadcasts the block to all peers to add to their blockchain.

6.3.5 Main Contribution

The main contributions of our work in this chapter are,

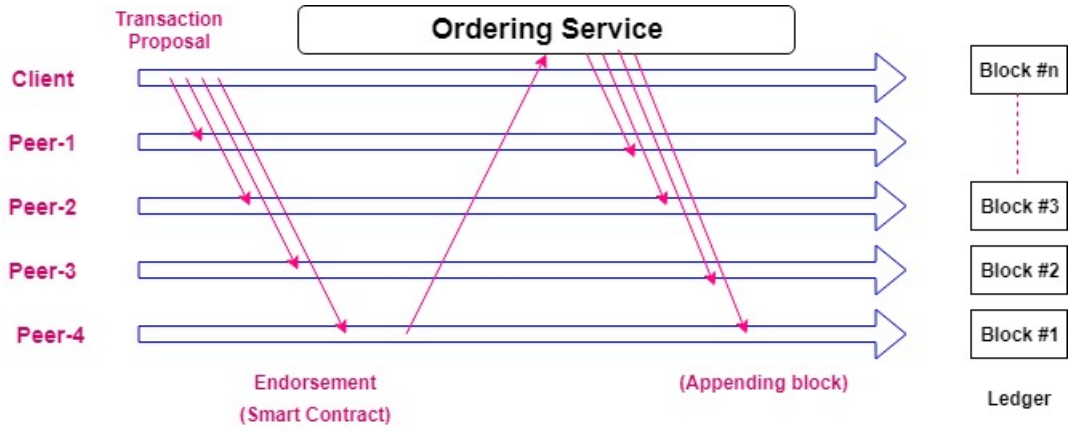


FIGURE 6.4: Transaction Flow of Proposed Framework [9]

1. The proposed framework will validate the documents and creates the blocks only for valid documents. Invalid documents are sent back to the clients.

2. In government, there is no need to keep the ordering service and validating service separately as officers are assumed to be trusted and both the services can be done by the same authority. Even if the officer is malicious, because of the permissioned blockchain setup, users identity will be revealed.

   This will reduce the resource usage overhead on the system when compared to Hyperledger Fabric framework where ordering and validation is done by separate peers.

3. The documents/transactions are passed on in a hierarchical authority in government offices and hence the endorsed transaction is directly submitted for validation rather than sending it back to the client again. This reduces the communication overhead on the system and the validated transaction is included in the ledger directly later. Hence, transaction throughput of the system is going to be increased.

6.3.6 Results and Discussion

In this section, the performance of our proposed framework will be analyzed based on the execution time, latency and throughput of Hyperledger Fabric 1.0.

6.3.6.1 Experimental Setup

A small private blockchain network was built in our lab utilizing hyperledger fabric, which was installed on an AMD Ryzen 7 5800X processor, Ubuntu 20.04.5 LTS operating system, and 8 GB of RAM for the performance assessment and validation of the proposed framework. Version 1.0 of Hyperledger Fabric blockchain framework, the most recent version at the time of our investigations with two organizations is used for the trials. There is one peer, one CA, and one MSP for each organisation. The organizations have connected using only one channel. The experiment is carried out using the SOLO ordering service. We utilized a customised version of the Hyperledger caliper to evaluate the performance for the frameworks.

6.3.6.2 Execution Time

The execution time evaluation is carried out by changing the number of transactions for the various functions and comparison is done for the two platforms. In general, as the number of transactions increases, the execution time also increases. Figure 6.5 illustrates that the query function execution time for our proposed framework are faster than those for Fabric 1.0.

6.3.6.3 Latency

The time interval between the submission of a transaction and its confirmation is referred to as transaction latency. Figure 6.6 shows the average execution time for the invoke and query functions for both the frameworks. The average latency for the proposed framework is better than the average latency of Fabric 1.0 in the query function.

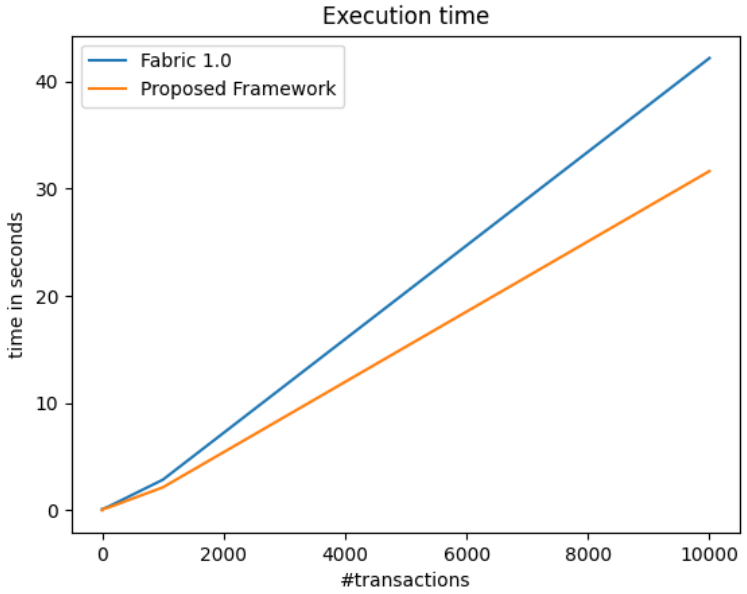


FIGURE 6.5: Comparison of execution time

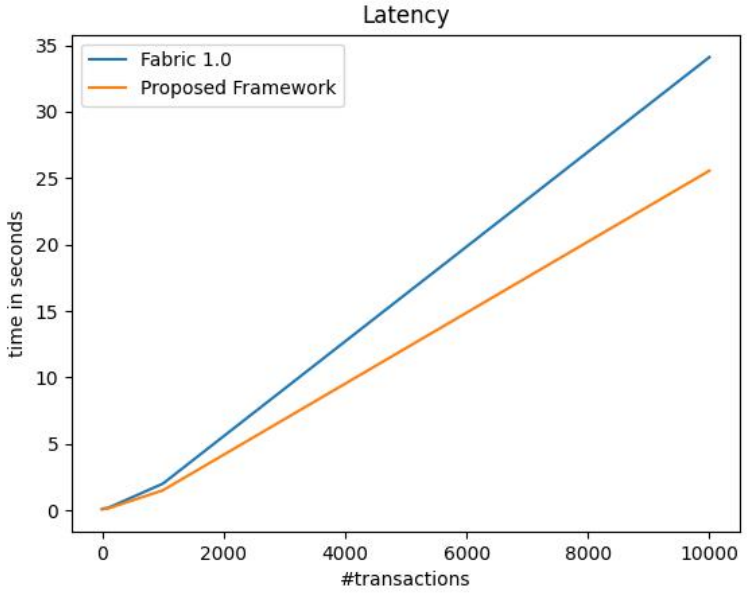


FIGURE 6.6: Comparison of latency

6.3.6.4 Throughput

The rate at which all valid transactions are committed is known as the throughput. Therefore, throughput=total committed transactions/total time. The throughput of each version for running the query function is shown in Figure 6.7. The proposed framework outperforms Fabric 1.0 in terms of throughput across all transaction counts.

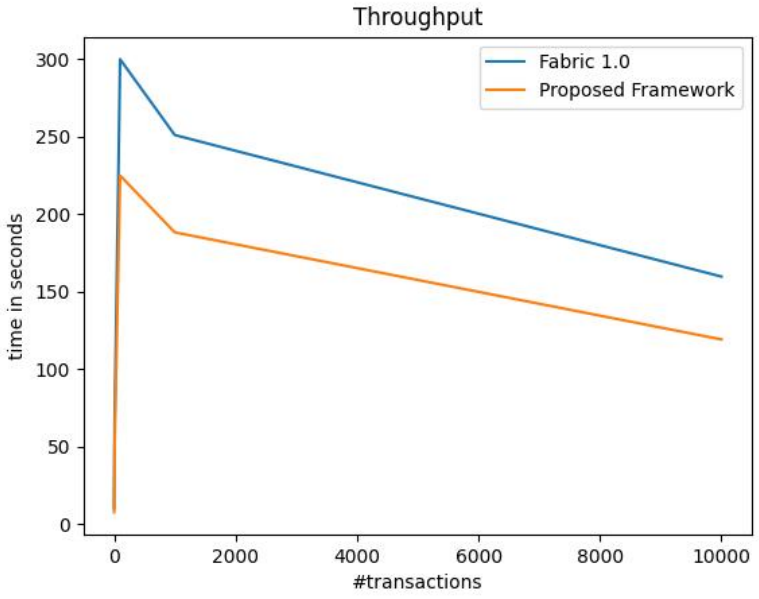


FIGURE 6.7: Comparison of throughput

6.4 Conclusion

A scalable decentralized framework for record keeping system in government using Hyperledger Fabric is proposed. Usage of permissioned blockchain Hyperledger Fabric helps the framework to avoid expensive computations as part of mining. The inbuilt property of Hyperledger Fabric makes all the transactions irrefutable, thereby it avoids all sort of fraudulent transactions and corruption. In addition to that, the ordering and validating services are combined together by taking advantage of government trusted officials. This helps in reducing resource usage overhead. Also, all the transactions flow in hierarchical pattern, except for invalid transactions. This reduces communication overhead compared

to other systems where communication is bidirectional. Embedding invalid documents in the ledger and tracing the states update of the transaction proposed at each hierarchical level can be considered for future work.

Chapter 7

Conclusion and Future Work

The conclusion of this thesis is summarised in this chapter, along with future research directions.

7.1 Conclusion

The following are the four main contributions of this thesis:

- **First Contribution**: Identity verifiable ring signature scheme for privacy protection in public blockchains.

  Although existing ring signatures have everlasting anonymity and seem to benefit users, they cannot identify the message's real signers if any such requirement arises. In Chapter 3, an identity verifiable ring signature scheme is proposed that not only has all of the properties of a ring signature but also the property that the signer can correctly determine who among the possible signers actually signed whenever it is required by proving to the members that he is the indeed signer of the message. One of our future works is to make IVRSS suitable for private and consortium blockchains and the same scheme can be extended for dynamic networks as well.

- **Second Contribution**: A decentralized group signature scheme (DGSS) for privacy protection on private blockchain.

A novel DGSS is proposed to address the identity privacy challenges in blockchain based applications. Lee et al. [92] group signature scheme was extended to DGSS by decentralization of group manager to eliminate the basic requirement of having trust on the group manager and also to improve identity privacy of group member. The security properties like unforgeability, anonymity, unlinkability and treceability for the proposed DGSS are also discussed. The proposed DGSS is more suitable for permissioned blockchain-based applications. Proof of correctness for proposed scheme ensures that the original message can still be reconstructed correctly, even after it is distributed among several group managers. Also a framework of blockchain-based e-auction protocol with DGSS is proposed. However, use of anonymous signatures for public blockchains is being explored as future work.

- **Third Contribution**: Dynamic decentralized group signature scheme for privacy protection on private blockchain.

  A dynamic decentralized group signature scheme is proposed to address the user identity privacy issues in blockchain-based applications. The DGSS is proposed by decentralizing the group manager but it is suitable for static environments only. Hence, it does not allow new members to join at run-time and the group members cannot be revoked. In this paper, the DGSS is extended as a dynamic decentralized group signature scheme. The proposed scheme can join new members in the group at any time and also revoke the group members. The performance of our scheme is more efficient compared to DGSS. This is achieved by reducing multiplication operations of verification algorithm. The proposed scheme is suitable for both blockchain based applications of dynamic setting and as well for the memory constrained devices and light-weight blockchains. The security properties like unforgeability, anonymity, unlinkability and traceability for the proposed scheme are also discussed. The suggested scheme is more suited to permissioned blockchain-based applications. The proof of correctness for the proposed scheme ensures that the original message can still be reconstructed correctly, even after it has been distributed among several group managers. However, making it suitable for the public blockchains will remain as our future work.

- **Fourth Contribution**: A scalable decentralized framework for record-keeping system in Government using hyperledger fabric.

  A novel decentralized framework for record-keeping system in government using Hyperledger Fabric is proposed. The user participating in the system has to prepare his transaction proposal which is to be recorded in the distributed ledger. The user has to submit his/her proposal for endorsement to the endorsers; the endorser endorses it and submits to the validators to check the validity and then include it in the ledger. Every peer maintains the ledger and the newly suggested blocks from the validators are added to it. The usage of permissioned blockchain Hyperledger Fabric helps the framework to avoid expensive computations as part of mining. The inbuilt property of Hyperledger Fabric makes all the transactions irrefutable, thereby avoids all sorts of fraudulent transactions and corruption. In addition to that, the ordering and validating services are combined together by taking advantage of government trusted officials. This helps in reducing resource usage overhead. Also, all the transactions flow in hierarchical pattern, except for invalid transactions. This reduces communication overhead compared to other systems where communication is bidirectional and performance of the system is going to be improved.

7.2 Future Directions

In this section, we suggest some future research directions based on the open research issues.

- The group signature schemes are more suitable for the permissioned blockchain systems since they require a group manager to set up the group. On the other hand, a ring signature is a unique type of group signature and more significant in terms of privacy than a normal group signature because of its unconditional anonymity and unforgeability features. In business activities like as electronic payments, auctions etc. ring signatures are quite important. Hence, the ring signature schemes are more suitable to protect user identity privacy in permissionless blockchains. As the existing ring signature schemes are unable to identify the actual signers of the message, it

is difficult to adopt them in some blockchain-based application like e-auction. Hence, an identity verifiable ring signature scheme is proposed to protect user identity privacy in permissionless blockchain-based applications in Chapter 3. One of our future works is to make IVRSS suitable for private and consortium blockchains.

- The *DGSS* proposed in Chapter 4 is static in nature, which doesn't permit new members to be added into the group and to revoke existing members from the group. Hence, *DGSS* is extended to as a dynamic decentralized group signature scheme that allows to add new members into the network at any time, and also to revoke the existing members from the network. This can be utilized to protect the identity privacy of the signers in real-time distributed applications. In addition to that, the performance of the proposed scheme is better compared to *DGSS*. This is achieved by reducing the number of multiplication operations in the verification algorithm of the proposed scheme. Further, there is a clear scope to reduce the complexity of signing and verification algorithms of *DDGSS* by making any changes in A, B, C *and* D parameters. The *DDGSS* can also be modified to make it suitable for any other blockchain-based applications in future.

- A decentralized group signature scheme is proposed to address the user identity privacy challenge in Chapter 4. Any user in the group can produce an anonymous signature on behalf of the group. Furthermore, the group manager has the power to revoke user anonymity. The group manager is centralized authority, can be malicious which leads towards biased decisions. To address the issue which is occurring due the centralized group manager, a DGSS is proposed by decentralizing the group manager. At the time of identifying actual signer of the message, all the n group mangers has to be present. It means that at the time of identifying signers, if any group manager node fails then the scheme is unable to identify the real signer. Thus, there is clear opportunity to come up with a scheme which has threshold among the group managers or fault tolerant. The *DGSS* can also be modified to make it suitable for any other blockchain-based applications.

Bibliography

[1] Abe, M., Ohkubo, M., and Suzuki, K. (2002). 1-out-of-n signatures from a variety of keys. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 415–432. Springer.

[2] Agarwal, A. and Saraswat, R. (2013). A survey of group signature technique, its applications and attacks. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(10).

[3] Agbo, C. C., Mahmoud, Q. H., and Eklund, J. M. (2019). Blockchain technology in healthcare: a systematic review. In *Healthcare*, volume 7, page 56. MDPI.

[4] Aitzhan, N. Z. and Svetinovic, D. (2018). Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE Transactions on Dependable and Secure Computing*, 15(05):840–852.

[5] Al Jawaheri, H., Al Sabah, M., Boshmaf, Y., and Erbad, A. (2020). Deanonymizing tor hidden service users through bitcoin transactions analysis. *Computers & Security*, 89:101684.

[6] Alamélou, Q., Blazy, O., Cauchie, S., and Gaborit, P. (2017). A code-based group signature scheme. *Designs, Codes and Cryptography*, 82(1):469–493.

[7] Ali, M. S., Vecchio, M., Pincheira, M., Dolui, K., Antonelli, F., and Rehmani, M. H. (2018). Applications of blockchains in the internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 21(2):1676–1717.

[8] Andreev, O. (2014). Blind signatures for bitcoin transactions. *Second draft February*, 22.

[9] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*, pages 1–15.

[10] Androulaki, E., Karame, G. O., Roeschlin, M., Scherer, T., and Capkun, S. (2013). Evaluating user privacy in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 34–51. Springer.

[11] Antonopoulos, A. M. (2014). *Mastering Bitcoin: unlocking digital cryptocurrencies.* " O'Reilly Media, Inc.".

[12] Ateniese, G., Camenisch, J., Joye, M., and Tsudik, G. (2000). A practical and provably secure coalition-resistant group signature scheme. In *Annual international cryptology conference*, pages 255–270. Springer.

[13] Ateniese, G., Song, D., and Tsudik, G. (2002). Quasi-efficient revocation of group signatures. In *International Conference on Financial Cryptography*, pages 183–197. Springer.

[14] Ateniese, G. and Tsudik, G. (1999). Group signatures a la carte. In *SODA*, volume 99, pages 848–849. Citeseer.

[15] Baliga, A., Solanki, N., Verekar, S., Pednekar, A., Kamat, P., and Chatterjee, S. (2018). Performance characterization of hyperledger fabric. In *2018 Crypto Valley conference on blockchain technology (CVCBT)*, pages 65–74. IEEE.

[16] Bellare, M., Micciancio, D., and Warinschi, B. (2003). Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *International conference on the theory and applications of cryptographic techniques*, pages 614–629. Springer.

[17] Bellare, M., Shi, H., and Zhang, C. (2005). Foundations of group signatures: The case of dynamic groups. In *Cryptographers' Track at the RSA Conference*, pages 136–153. Springer.

[18] Bender, A., Katz, J., and Morselli, R. (2006). Ring signatures: Stronger definitions, and constructions without random oracles. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 60–79. Springer.

[19] Bender, A., Katz, J., and Morselli, R. (2009). Ring signatures: Stronger definitions, and constructions without random oracles. *Journal of Cryptology*, 22(1):114–138.

[20] Bernstein, P. A. and Goodman, N. (1981). Concurrency control in distributed database systems. *ACM Computing Surveys (CSUR)*, 13(2):185–221.

[21] Bhosale, J. and Mavale, S. (2018). Volatility of select crypto-currencies: A comparison of bitcoin, ethereum and litecoin. *Annu. Res. J. SCMS, Pune*, 6.

[22] Blass, E.-O. and Kerschbaum, F. (2018). Strain: A secure auction for blockchains. In *European Symposium on Research in Computer Security*, pages 87–110. Springer.

[23] Blum, M., Feldman, P., and Micali, S. (2019). Non-interactive zero-knowledge and its applications. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 329–349. ACM.

[24] Boneh, D., Gentry, C., Lynn, B., and Shacham, H. (2003). Aggregate and verifiably encrypted signatures from bilinear maps. In *International conference on the theory and applications of cryptographic techniques*, pages 416–432. Springer.

[25] Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., and Felten, E. W. (2015). Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*, pages 104–121. IEEE.

[26] Bonneau, J., Narayanan, A., Miller, A., Clark, J., Kroll, J. A., and Felten, E. W. (2014). Mixcoin: Anonymity for bitcoin with accountable mixes. In *International Conference on Financial Cryptography and Data Security*, pages 486–504. Springer.

[27] Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J., and Petit, C. (2015). Short accountable ring signatures based on ddh. In *European Symposium on Research in Computer Security*, pages 243–265. Springer.

[28] Bresson, E., Stern, J., and Szydlo, M. (2002). Threshold ring signatures and applications to ad-hoc groups. In *Annual International Cryptology Conference*, pages 465–480. Springer.

[29] Cai, W., Wang, Z., Ernst, J. B., Hong, Z., Feng, C., and Leung, V. C. (2018). Decentralized applications: The blockchain-empowered software system. *IEEE Access*, 6:53019–53033.

[30] Camenisch, J., Drijvers, M., Lehmann, A., Neven, G., and Towa, P. (2020). Short threshold dynamic group signatures. In *International Conference on Security and Cryptography for Networks*, pages 401–423. Springer.

[31] Castro, M., Liskov, B., et al. (1999). Practical byzantine fault tolerance. In *OsDI*, volume 99, pages 173–186.

[32] Chandran, N., Groth, J., and Sahai, A. (2007). Ring signatures of sub-linear size without random oracles. In *International Colloquium on Automata, Languages, and Programming*, pages 423–434. Springer.

[33] Chang, S., Wong, D. S., Mu, Y., and Zhang, Z. (2009). Certificateless threshold ring signature. *Information Sciences*, 179(20):3685–3696.

[34] Chase, M. (2007). Multi-authority attribute based encryption. In *Theory of cryptography conference*, pages 515–534. Springer.

[35] Chaum, D. and Van Heyst, E. (1991). Group signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 257–265. Springer.

[36] Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90.

[37] Chen, L. and Pedersen, T. P. (1994). New group signature schemes. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 171–181. Springer.

[38] Chen, Y.-H., Chen, S.-H., and Lin, I.-C. (2018). Blockchain based smart contract for bidding system. In *2018 IEEE International Conference on Applied System Invention (ICASI)*, pages 208–211. IEEE.

[39] Chow, S. S. and Yap, W.-S. (2007). Certificateless ring signatures. *Cryptology ePrint Archive*.

[40] Chung, G., Desrosiers, L., Gupta, M., Sutton, A., Venkatadri, K., Wong, O., and Zugic, G. (2019). Performance tuning and scaling enterprise blockchain applications. *arXiv preprint arXiv:1912.11456*.

[41] Conti, M., Kumar, E. S., Lal, C., and Ruj, S. (2018). A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials*, 20(4):3416–3452.

[42] Courtois, N. T. and Mercer, R. (2017). Stealth address and key management techniques in blockchain systems. *ICISSP*, 2017:559–566.

[43] Cruz, J. P. and Kaji, Y. (2017). E-voting system based on the bitcoin protocol and blind signatures. *IPSJ Transactions on Mathematical Modeling and Its Applications*, 10(1):14–22.

[44] Decker, C. and Wattenhofer, R. (2015). A fast and scalable payment network with bitcoin duplex micropayment channels. In *Symposium on Self-Stabilizing Systems*, pages 3–18. Springer.

[45] DeGroot, M. H. (1974). Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121.

[46] Devidas, S., Rukma Rekha, N., and Subba Rao, Y. (2020). Decentralized framework for record-keeping system in government using hyperledger fabric. In *Proceedings of the Third International Conference on Computational Intelligence and Informatics*, pages 663–672. Springer.

[47] Devidas, S., Rukma Rekha, N., and Subba Rao, Y. (2023). Dynamic decentralized group signature scheme for privacy protection in blockchain. In *International Conference on Innovative Computing and Communications*, pages 745–760. Springer.

[48] Devidas, S., YV, S. R., and Rekha, N. R. (2021). A decentralized group signature scheme for privacy protection in a blockchain. *International Journal of Applied Mathematics and Computer Science*, 31(2):353–364.

[49] Dikshit, P. and Singh, K. (2017). Efficient weighted threshold ecdsa for securing bitcoin wallet. In *2017 ISEA Asia Security and Privacy (ISEASP)*, pages 1–9. IEEE.

[50] Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., and Wang, J. (2018). Untangling blockchain: A data processing view of blockchain systems. *IEEE transactions on knowledge and data engineering*, 30(7):1366–1385.

[51] Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., and Tan, K.-L. (2017). Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM international conference on management of data*, pages 1085–1100.

[52] Dodis, Y., Kiayias, A., Nicolosi, A., and Shoup, V. (2004). Anonymous identification in ad hoc groups. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 609–626. Springer.

[53] Du, W. and Atallah, M. J. (2001). Secure multi-party computation problems and their applications: a review and open problems. In *Proceedings of the 2001 workshop on New security paradigms*, pages 13–22.

[54] Dwork, C. (2008). Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer.

[55] Dwork, C., Naor, M., and Sahai, A. (2004). Concurrent zero-knowledge. *Journal of the ACM (JACM)*, 51(6):851–898.

[56] Feng, Q., He, D., Zeadally, S., Khan, M. K., and Kumar, N. (2019). A survey on privacy protection in blockchain system. *Journal of Network and Computer Applications*, 126:45–58.

[57] Fernandez-Vazquez, S., Rosillo, R., De La Fuente, D., and Priore, P. (2019). Blockchain in fintech: A mapping study. *Sustainability*, 11(22):6366.

[58] Fleder, M., Kester, M. S., and Pillai, S. (2015). Bitcoin transaction graph analysis. *arXiv preprint arXiv:1502.01657*.

[59] Fontaine, C. and Galand, F. (2007). A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security*, 2007:1–10.

[60] Fujisaki, E. and Suzuki, K. (2007). Traceable ring signature. In *International Workshop on Public Key Cryptography*, pages 181–200. Springer.

[61] Gai, K., Wu, Y., Zhu, L., Zhang, Z., and Qiu, M. (2019). Differential privacy-based blockchain for industrial internet-of-things. *IEEE Transactions on Industrial Informatics*, 16(6):4156–4165.

[62] Galal, H. S. and Youssef, A. M. (2018). Verifiable sealed-bid auction on the ethereum blockchain. In *International Conference on Financial Cryptography and Data Security*, pages 265–278. Springer.

[63] Gao, W., Hatcher, W. G., and Yu, W. (2018). A survey of blockchain: techniques, applications, and challenges. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–11. IEEE.

[64] Garg, S., Gentry, C., Halevi, S., Sahai, A., and Waters, B. (2013). Attribute-based encryption for circuits from multilinear maps. In *Annual Cryptology Conference*, pages 479–499. Springer.

[65] Gill, M. and Taylor, G. (2004). Preventing money laundering or obstructing business? financial companies' perspectives on 'know your customer'procedures. *British Journal of Criminology*, 44(4):582–594.

[66] Gorbunov, S., Vaikuntanathan, V., and Wee, H. (2015). Attribute-based encryption for circuits. *Journal of the ACM (JACM)*, 62(6):1–33.

[67] Gorenflo, C., Lee, S., Golab, L., and Keshav, S. (2020). Fastfabric: Scaling hyperledger fabric to 20 000 transactions per second. *International Journal of Network Management*, 30(5):e2099.

[68] Green, M. and Miers, I. (2017). Bolt: Anonymous payment channels for decentralized currencies. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 473–489.

[69] Gu, K. and Wu, N. (2018). Constant size traceable ring signature scheme without random oracles. *Cryptology ePrint Archive*.

[70] Guannan, F. and Pan, D. (2016). Research on trusted execution environment building technology based on trustzone. *Netinfo Security*.

[71] Guo, R., Shi, H., Zhao, Q., and Zheng, D. (2018). Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems. *IEEE access*, 6:11676–11686.

[72] Hassija, V., Bansal, G., Chamola, V., Saxena, V., and Sikdar, B. (2019). Blockcom: A blockchain based commerce model for smart communities using auction mechanism. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE.

[73] Ho, S.-S. and Ruan, S. (2011). Differential privacy for location pattern mining. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, pages 17–24.

[74] Hwang, J. Y., Chen, L., Cho, H. S., and Nyang, D. (2015). Short dynamic group signature scheme supporting controllable linkability. *IEEE Transactions on Information Forensics and Security*, 10(6):1109–1124.

[75] Ioannidis, I. and Grama, A. (2003). An efficient protocol for yao's millionaires' problem. In *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, pages 6–pp. IEEE.

[76] Joshi, A. P., Han, M., and Wang, Y. (2018). A survey on security and privacy issues of blockchain technology. *Mathematical foundations of computing*, 1(2):121.

[77] Jouini, M., Rabai, L. B. A., and Aissa, A. B. (2014). Classification of security threats in information systems. *ANT/SEIT*, 32:489–496.

[78] Jung, T., Li, X.-Y., Wan, Z., and Wan, M. (2013). Privacy preserving cloud data access with multi-authorities. In *2013 Proceedings IEEE INFOCOM*, pages 2625–2633. IEEE.

[79] Karame, G. (2016). On the security and scalability of bitcoin's blockchain. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1861–1862.

[80] Kikwai, B. K. (2017). Elliptic curve digital signatures and their application in the bitcoin crypto-currency transactions. *International Journal of Scientific and Research Publications*, 7(11):11.

[81] Kim, H., Lee, Y., Abdalla, M., and Park, J. H. (2021). Practical dynamic group signature with efficient concurrent joins and batch verifications. *Journal of information security and applications*, 63:103003.

[82] Kim, H.-J., Lim, J. I., and Lee, D. H. (2000). Efficient and secure member deletion in group signature schemes. In *International Conference on Information Security and Cryptology*, pages 150–161. Springer.

[83] Kim, S., Kwon, Y., and Cho, S. (2018). A survey of scalability solutions on blockchain. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1204–1207. IEEE.

[84] Kobusińska, A., Brzeziński, J., Boroń, M., Inatlewski, Ł., Jabczyński, M., and Maciejewski, M. (2016). A branch hash function as a method of message synchronization in anonymous p2p conversations. *International Journal of Applied Mathematics and Computer Science*, 26(2):479–493.

[85] Kong, W., Jiang, B., Fan, Q., Zhu, L., and Wei, X. (2018). Personal identification based on brain networks of eeg signals. *International Journal of Applied Mathematics and Computer Science*, 28(4).

[86] Kosba, A., Miller, A., Shi, E., Wen, Z., and Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE symposium on security and privacy (SP)*, pages 839–858. IEEE.

[87] Krenn, S., Samelin, K., and Striecks, C. (2019). Practical group-signatures with privacy-friendly openings. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–10.

[88] Kreps, J., Narkhede, N., Rao, J., et al. (2011). Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB*, volume 11, pages 1–7.

[89] Krishna, V. (2009). *Auction theory*. Academic press.

[90] Lafourcade, P., Nopere, M., Picot, J., Pizzuti, D., and Roudeix, E. (2019). Security analysis of auctionity: a blockchain based e-auction. In *International Symposium on Foundations & Practice of Security FPS 19*, pages 0–0.

[91] Langlois, A., Ling, S., Nguyen, K., and Wang, H. (2014). Lattice-based group signature scheme with verifier-local revocation. In *International workshop on public key cryptography*, pages 345–361. Springer.

[92] Lee, C.-C., Ho, P.-F., and Hwang, M.-S. (2009). A secure e-auction scheme based on group signatures. *Information Systems Frontiers*, 11(3):335–343.

[93] Lewko, A. and Waters, B. (2011). Decentralizing attribute-based encryption. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 568–588. Springer.

[94] Li, C.-Y., Chen, X.-B., Chen, Y.-L., Hou, Y.-Y., and Li, J. (2018). A new lattice-based signature scheme in post-quantum blockchain network. *IEEE Access*, 7:2026–2033.

[95] Li, H. and Xue, W. (2021). A blockchain-based sealed-bid e-auction scheme with smart contract and zero-knowledge proof. *Security and Communication Networks*, 2021.

[96] Li, S., Zhang, Y., Wang, Y., and Sun, W. (2019). Utility optimization–based bandwidth allocation for elastic and inelastic services in peer–to–peer networks. *International Journal of Applied Mathematics and Computer Science*, 29(1):111–123.

[97] Li, X., Jiang, P., Chen, T., Luo, X., and Wen, Q. (2017). A survey on the security of blockchain systems. *Future Generation Computer Systems*.

[98] Libert, B., Ling, S., Mouhartem, F., Nguyen, K., and Wang, H. (2016). Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 373–403. Springer.

[99] Lin, Q., Yan, H., Huang, Z., Chen, W., Shen, J., and Tang, Y. (2018). An id-based linearly homomorphic signature scheme and its application in blockchain. *IEEE Access*, 6:20632–20640.

[100] Ling, S., Nguyen, K., and Wang, H. (2015). Group signatures from lattices: simpler, tighter, shorter, ring-based. In *IACR International Workshop on Public Key Cryptography*, pages 427–449. Springer.

[101] Liu, J. K., Wei, V. K., and Wong, D. S. (2004). Linkable spontaneous anonymous group signature for ad hoc groups. In *Australasian Conference on Information Security and Privacy*, pages 325–335. Springer.

[102] Luo, Q. and Jiang, C.-Y. (2020). A new constant-size group signature scheme from lattices. *IEEE Access*, 8:10198–10207.

[103] Manimaran, P. and Dhanalakshmi, R. (2019). Blockchain-based smart contract for e-bidding system. In *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, pages 55–59. IEEE.

[104] Maxwell, G. (2013a). Coinjoin: Bitcoin privacy for the real world. In *Post on Bitcoin forum*, pages 0–0.

[105] Maxwell, G. (2013b). Coinjoin: Bitcoin privacy for the real world.

[106] Maxwell, G. (2013c). Coinswap: Transaction graph disjoint trustless trading. *CoinSwap: Transactiongraphdisjointtrustlesstrading (October 2013)*.

[107] McFadden, F. R. and Hoffer, J. A. (1991). *Database management*. Benjamin-Cummings Publishing Co., Inc.

[108] Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., and Savage, S. (2013). A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140.

[109] Mercer, R. (2016). Privacy on the blockchain: Unique ring signatures. *arXiv preprint arXiv:1612.01188*.

[110] Miers, I., Garman, C., Green, M., and Rubin, A. D. (2013). Zerocoin: Anonymous distributed e-cash from bitcoin. In *2013 IEEE Symposium on Security and Privacy*, pages 397–411. IEEE.

[111] Miller, A., Bentov, I., Bakshi, S., Kumaresan, R., and McCorry, P. (2019). Sprites and state channels: Payment networks that go faster than lightning. In *International Conference on Financial Cryptography and Data Security*, pages 508–526. Springer.

[112] Monrat, A. A., Schelén, O., and Andersson, K. (2019). A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access*, 7:117134–117151.

[113] Naidu, P. S., Kharat, R., Tekade, R., Mendhe, P., and Magade, V. (2016). E-voting system using visual cryptography & secure multi-party computation. In *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, pages 1–4. IEEE.

[114] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260.

[115] Naor, M. (2002). Deniable ring authentication. In *Annual International Cryptology Conference*, pages 481–498. Springer.

[116] Narayanan, A., Bonneau, J., Felten, E., Miller, A., and Goldfeder, S. (2016). *Bitcoin and cryptocurrency technologies: a comprehensive introduction.* Princeton University Press.

[117] Nasir, M. H., Arshad, J., Khan, M. M., Fatima, M., Salah, K., and Jayaraman, R. (2022). Scalable blockchains—a systematic review. *Future Generation Computer Systems*, 126:136–162.

[118] Natoli, C. and Gramoli, V. (2016). The balance attack against proof-of-work blockchains: The r3 testbed as an example. *arXiv preprint arXiv:1612.09426*.

[119] Noether, S., Mackenzie, A., et al. (2016). Ring confidential transactions. *Ledger*, 1:1–18.

[120] Noether, S., Noether, S., and Mackenzie, A. (2014). A note on chain reactions in traceability in cryptonote 2.0. *Research Bulletin MRL-0001. Monero Research Lab*, 1:1–8.

[121] Ongaro, D. and Ousterhout, J. (2014). In search of an understandable consensus algorithm. In *2014 USENIX Annual Technical Conference (Usenix ATC 14)*, pages 305–319.

[122] Özsu, M. T. and Valduriez, P. (1999). *Principles of distributed database systems*, volume 2. Springer.

[123] Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer.

[124] Pedersen, T. P. (1991). Non-interactive and information-theoretic secure verifiable secret sharing. In *Annual international cryptology conference*, pages 129–140. Springer.

[125] Perera, M. N. S. and Koshiba, T. (2018). Fully dynamic group signature scheme with member registration and verifier-local revocation. In *International conference on mathematics and computing*, pages 399–415. Springer.

[126] Perera, M. N. S., Nakamura, T., Hashimoto, M., Yokoyama, H., Cheng, C.-M., and Sakurai, K. (2022). A survey on group signatures and ring signatures: traceability vs. anonymity. *Cryptography*, 6(1):3.

[127] platon (2022). Platon.

[128] Poon, J. and Dryja, T. (2016). The bitcoin lightning network: Scalable off-chain instant payments.

[129] Prado-Romero, M. A., Doerr, C., and Gago-Alonso, A. (2018). Discovering bitcoin mixing using anomaly detection. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 22nd Iberoamerican Congress, CIARP 2017, Valparaíso, Chile, November 7–10, 2017, Proceedings 22*, pages 534–541. Springer.

[130] Qusa, H., Tarazi, J., and Akre, V. (2020). Secure e-auction system using blockchain: Uae case study. In *2020 Advances in Science and Engineering Technology International Conferences (ASET)*, pages 1–5. IEEE.

[131] Reid, F. and Harrigan, M. (2013). An analysis of anonymity in the bitcoin system. In *Security and privacy in social networks*, pages 197–223. Springer.

[132] Ren, H., Zhang, P., Shentu, Q., Liu, J. K., and Yuen, T. H. (2018). Compact ring signature in the standard model for blockchain. In *International Conference on Information Security Practice and Experience*, pages 50–65. Springer.

[133] Ren, J. and Harn, L. (2008). Generalized ring signatures. *IEEE Transactions on Dependable and Secure Computing*, 5(3):155–163.

[134] Rivest, R. L., Shamir, A., and Tauman, Y. (2001). How to leak a secret. In *International conference on the theory and application of cryptology and information security*, pages 552–565. Springer.

[135] Ron, D. and Shamir, A. (2013). Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*, pages 6–24. Springer.

[136] Ruffing, T., Moreno-Sanchez, P., and Kate, A. (2014). Coinshuffle: Practical decentralized coin mixing for bitcoin. In *Computer Security-ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II 19*, pages 345–364. Springer.

[137] Sakai, Y., Emura, K., Hanaoka, G., Kawai, Y., Matsuda, T., and Omote, K. (2012). Group signatures with message-dependent opening. In *International Conference on Pairing-Based Cryptography*, pages 270–294. Springer.

[138] Sánchez, D. C. (2018). Raziel: Private and verifiable smart contracts on blockchains. *arXiv preprint arXiv:1807.09484*.

[139] Saraswat, V. and Pandey, S. K. (2014). How to leak a secret and reap the rewards too. In *International Conference on Cryptology and Information Security in Latin America*, pages 348–367. Springer.

[140] Sasson, E. B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., and Virza, M. (2014). Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE symposium on security and privacy*, pages 459–474. IEEE.

[141] Sato, M. and Matsuo, S. (2017). Long-term public blockchain: Resilience against compromise of underlying cryptography. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8. IEEE.

[142] Schott, P. A. (2006). *Reference guide to anti-money laundering and combating the financing of terrorism*. World Bank Publications.

[143] Shacham, H. and Waters, B. (2007). Efficient ring signatures without random oracles. In *International Workshop on Public Key Cryptography*, pages 166–180. Springer.

[144] ShenTu, Q. and Yu, J. (2015). A blind-mixing scheme for bitcoin based on an elliptic curve cryptography blind digital signature algorithm. *arXiv preprint arXiv:1510.05833*.

[145] Shrestha, R. and Kim, S. (2019). Integration of iot with blockchain and homomorphic encryption: Challenging issues and opportunities. In *Advances in Computers*, volume 115, pages 293–331. Elsevier.

[146] Sovbetov, Y. (2018). Factors influencing cryptocurrency prices: Evidence from bitcoin, ethereum, dash, litcoin, and monero. *Journal of Economics and Financial Analysis*, 2(2):1–27.

[147] Steffen, S., Bichsel, B., Baumgartner, R., and Vechev, M. (2022). Zeestar: Private smart contracts by homomorphic encryption and zero-knowledge proofs. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1543–1543. IEEE Computer Society.

[148] Stinson, D. R. (2005). *Cryptography: theory and practice*. Chapman and Hall/CRC.

[149] Sun, S.-F., Au, M. H., Liu, J. K., and Yuen, T. H. (2017). Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In *European Symposium on Research in Computer Security*, pages 456–474. Springer.

[150] Sun, Y. and Liu, Y. (2021). An efficient fully dynamic group signature with message dependent opening from lattice. *Cybersecurity*, 4(1):1–15.

[151] Sun, Y., Liu, Y., and Wu, B. (2019). An efficient full dynamic group signature scheme over ring. *Cybersecurity*, 2(1):1–15.

[152] Sun, Y., Sun, Y., Luo, M., Gu, L., Zheng, S., and Yang, Y. (2013). Comment on lee et al.'s group signature and e-auction scheme. *Information Systems Frontiers*, 15(1):133–139.

[153] Swathi, P. and Venkatesan, M. (2021). Scalability improvement and analysis of permissioned-blockchain. *ICT Express*, 7(3):283–289.

[154] Thakkar, P., Nathan, S., and Viswanathan, B. (2018). Performance benchmarking and optimizing hyperledger fabric blockchain platform. In *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 264–276. IEEE.

[155] Tsai, C.-Y., Ho, P.-F., and Hwang, M.-S. (2018). A secure group signature scheme. *IJ Network Security*, 20(2):201–205.

[156] Valdeolmillos, D., Mezquita, Y., González-Briones, A., Prieto, J., and Corchado, J. M. (2019). Blockchain technology: a review of the current challenges of cryptocurrency. In *International Congress on Blockchain and Applications*, pages 153–160. Springer.

[157] Valenta, L. and Rowan, B. (2015). Blindcoin: Blinded, accountable mixes for bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 112–126. Springer.

[158] Wang, D., Zhao, J., and Mu, C. (2021). Research on blockchain-based e-bidding system. *Applied Sciences*, 11(9):4011.

[159] Wang, D., Zhao, J., and Wang, Y. (2020). A survey on privacy protection of blockchain: The technology and application. *IEEE Access*, 8:108766–108781.

[160] Wang, X., Zha, X., Ni, W., Liu, R. P., Guo, Y. J., Niu, X., and Zheng, K. (2019). Survey on blockchain for internet of things. *Computer Communications*, 136:10–29.

[161] Wei, V. K. (2005). Tracing-by-linking group signatures. In *International Conference on Information Security*, pages 149–163. Springer.

[162] Whitman, M. E. and Mattord, H. J. (2021). *Principles of information security*. Cengage Learning.

[163] Wood, G. et al. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32.

[164] Xie, J., Tang, H., Huang, T., Yu, F. R., Xie, R., Liu, J., and Liu, Y. (2019). A survey of blockchain technology applied to smart cities: Research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(3):2794–2830.

[165] Xu, J., Zhang, Z., and Feng, D. (2004). A ring signature scheme using bilinear pairings. In *International Workshop on Information Security Applications*, pages 160–169. Springer.

[166] Xu, S. and Yung, M. (2004). Accountable ring signatures: A smart card approach. In *Smart Card Research and Advanced Applications VI*, pages 271–286. Springer.

[167] Yao, A. C. (1982). Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE.

[168] Yu, Z., Au, M. H., Yu, J., Yang, R., Xu, Q., and Lau, W. F. (2019). New empirical traceability analysis of cryptonote-style blockchains. In *Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers*, pages 133–149. Springer.

[169] Yuan, C., Xu, M.-x., and Si, X.-m. (2017). Research on a new signature scheme on blockchain. *Security and Communication Networks*, 2017.

[170] Zhang, F. and Kim, K. (2002). Id-based blind signature and ring signature from pairings. In *International conference on the theory and application of cryptology and information security*, pages 533–547. Springer.

[171] Zhang, L., Zhang, F., and Wu, W. (2007). A provably secure ring signature scheme in certificateless cryptography. In *International Conference on Provable Security*, pages 103–121. Springer.

[172] Zhang, R., Xue, R., and Liu, L. (2019). Security and privacy on blockchain. *ACM Computing Surveys (CSUR)*, 52(3):1–34.

[173] Zhao, Y., Zhao, J., Jiang, L., Tan, R., Niyato, D., Li, Z., Lyu, L., and Liu, Y. (2020). Privacy-preserving blockchain-based federated learning for iot devices. *IEEE Internet of Things Journal*, 8(3):1817–1829.

[174] Zheng, H., Wu, Q., Xie, J., Guan, Z., Qin, B., and Gu, Z. (2020). An organization-friendly blockchain system. *Computers & Security*, 88:101598.

[175] Zheng, Z., Xie, S., Dai, H.-N., Chen, X., and Wang, H. (2018). Blockchain challenges and opportunities: A survey. *International journal of web and grid services*, 14(4):352–375.

[176] Zhenyu, N., Fengwei, Z., and Weisong, S. (2019). A study of using tee on edge computing. *Journal of Computer Research and Development*, 56(7):1441.

[177] Zhou, Q., Huang, H., Zheng, Z., and Bian, J. (2020). Solutions to scalability of blockchain: A survey. *Ieee Access*, 8:16440–16455.

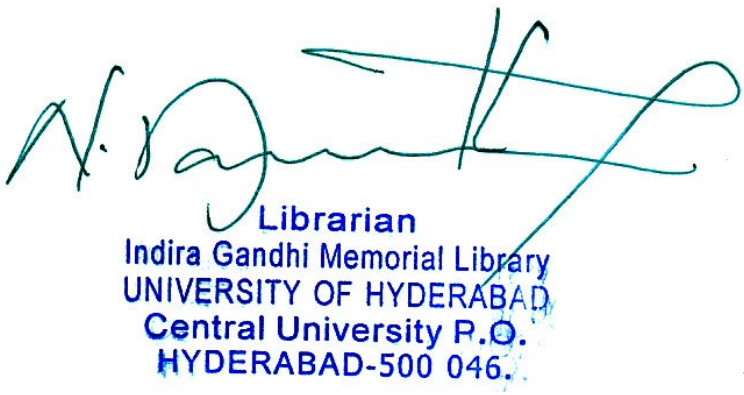
[178] Zhou, X., Yang, X., Wei, P., and Hu, Y. (2007). Dynamic group signature with forward security and its application. In *Sixth International Conference on Grid and Cooperative Computing (GCC 2007)*, pages 473–480. IEEE.

[179] Zhu, Y., Guo, R., Gan, G., and Tsai, W.-T. (2016). Interactive incontestable signature for transactions confirmation in bitcoin blockchain. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 443–448. IEEE.

[180] Ziegeldorf, J. H., Matzutt, R., Henze, M., Grossmann, F., and Wehrle, K. (2018). Secure and anonymous decentralized bitcoin mixing. *Future Generation Computer Systems*, 80:448–466.

Signature Schemes for Privacy Protection in Blockchains

by Devidas S



Submission date: 27-Apr-2023 12:07PM (UTC+0530)
Submission ID: 2076951411
File name: S_Devidas.pdf (1.43M)
Word count: 32262
Character count: 163362

Signature Schemes for Privacy Protection in Blockchains

ORIGINALITY REPORT

19%	11%	15%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES Overall Similarity Index = 19 − (6+3+2) = 8%

1	bibliotekanauki.pl Internet Source This publication is by the student	6%
2	"International Conference on Innovative Computing and Communications", Springer Science and Business Media LLC, 2023 Publication This publication is by the student	3%
3	"Proceedings of the Third International Conference on Computational Intelligence and Informatics", Springer Science and Business Media LLC, 2020 Publication This publication is by the student	2%
4	Submitted to Jawaharlal Nehru Technological University Student Paper	1%
5	Submitted to City University of New York System Student Paper	<1%
6	Abylay Satybaldy, Mariusz Nowostawski. "Review of Techniques for Privacy-Preserving Blockchain Systems", Proceedings of the 2nd	<1%

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

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

Signature Schemes for Privacy Protection in Blockchains

ORIGINALITY REPORT

19%	11%	15%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	bibliotekanauki.pl Internet Source	6%
2	"International Conference on Innovative Computing and Communications", Springer Science and Business Media LLC, 2023 Publication	3%
3	"Proceedings of the Third International Conference on Computational Intelligence and Informatics", Springer Science and Business Media LLC, 2020 Publication	2%
4	Submitted to Jawaharlal Nehru Technological University Student Paper	1%
5	Submitted to City University of New York System Student Paper	<1%
6	Abylay Satybaldy, Mariusz Nowostawski. "Review of Techniques for Privacy-Preserving Blockchain Systems", Proceedings of the 2nd	<1%

	ACM International Symposium on Blockchain and Secure Critical Infrastructure, 2020 Publication	
7	www.mdpi.com Internet Source	<1 %
8	Swathi P., M. Venkatesan. "Scalability improvement and analysis of permissioned-blockchain", ICT Express, 2021 Publication	<1 %
9	Jian Ren, Lein Harn. "An Efficient Threshold Anonymous Authentication Scheme for Privacy-Preserving Communications", IEEE Transactions on Wireless Communications, 2013 Publication	<1 %
10	www.researchgate.net Internet Source	<1 %
11	jwcn-eurasipjournals.springeropen.com Internet Source	<1 %
12	Lecture Notes in Computer Science, 2015. Publication	<1 %
13	Adam Bender. "Ring Signatures: Stronger Definitions, and Constructions Without Random Oracles", Lecture Notes in Computer Science, 2006 Publication	<1 %

14	ebin.pub Internet Source	<1 %
15	eprints.soton.ac.uk Internet Source	<1 %
16	www.ijert.org Internet Source	<1 %
17	hdl.handle.net Internet Source	<1 %
18	Weidong Fang, Wei Chen, Wuxiong Zhang, Jun Pei, Weiwei Gao, Guohui Wang. "Digital signature scheme for information non-repudiation in blockchain: a state of the art review", EURASIP Journal on Wireless Communications and Networking, 2020 Publication	<1 %
19	"Topics in Cryptology – CT-RSA 2020", Springer Science and Business Media LLC, 2020 Publication	<1 %
20	Jian Ren. "Generalized Ring Signatures", IEEE Transactions on Dependable and Secure Computing, 07/2008 Publication	<1 %
21	Submitted to Universidad de Salamanca Student Paper	<1 %

22	Wang, C.H.. "A new ring signature scheme with signer-admission property", Information Sciences, 20070201 Publication	<1 %
23	"Decentralised Internet of Things", Springer Science and Business Media LLC, 2020 Publication	<1 %
24	"Information Security and Cryptology", Springer Science and Business Media LLC, 2020 Publication	<1 %
25	Submitted to Griffth University Student Paper	<1 %
26	Lecture Notes in Computer Science, 2001. Publication	<1 %
27	Lecture Notes in Computer Science, 2003. Publication	<1 %
28	Xiaofang Li, Yurong Mei, Jing Gong, Feng Xiang, Zhixin Sun. "A Blockchain Privacy Protection Scheme Based on Ring Signature", IEEE Access, 2020 Publication	<1 %
29	"Cryptology and Network Security", Springer Science and Business Media LLC, 2016 Publication	<1 %

30	Submitted to University of Wales Institute, Cardiff Student Paper	<1 %
31	"Blockchain and Trustworthy Systems", Springer Science and Business Media LLC, 2021 Publication	<1 %
32	"Information and Communications Security", Springer Nature, 2004 Publication	<1 %
33	Maharage Nisansala Sevwandi Perera, Toru Nakamura, Masayuki Hashimoto, Hiroyuki Yokoyama, Chen-Mou Cheng, Kouichi Sakurai. "A Survey on Group Signatures and Ring Signatures: Traceability vs. Anonymity", Cryptography, 2022 Publication	<1 %
34	acris.aalto.fi Internet Source	<1 %
35	jips-k.org Internet Source	<1 %
36	"Cryptology and Network Security", Springer Science and Business Media LLC, 2018 Publication	<1 %
37	Qingkuan Dong, Xiaoping Li, Yanming Liu. "Two extensions of the ring signature scheme	<1 %

	of Rivest–Shamir–Taumann", Information Sciences, 2012 Publication	
38	Lingling Wang. "A survey of ring signature", Frontiers of Electrical and Electronic Engineering in China, 01/2008 Publication	<1 %
39	link.springer.com Internet Source	<1 %
40	Submitted to Lebanese American University Student Paper	<1 %
41	Lecture Notes in Computer Science, 2006. Publication	<1 %
42	Jian Reny, Lein Harnz, Tongtong Li. "How to leak a secret from multiple sources", MILCOM 2008 - 2008 IEEE Military Communications Conference, 2008 Publication	<1 %
43	"Security and Cryptography for Networks", Springer Science and Business Media LLC, 2010 Publication	<1 %
44	Huaqun Wang. "Cryptanalysis of a Generalized Ring Signature Scheme", IEEE Transactions on Dependable and Secure Computing, 2009 Publication	<1 %

45	Lecture Notes in Computer Science, 2005. Publication	<1 %
46	Lecture Notes in Computer Science, 2014. Publication	<1 %
47	Qassim Nasir, Ilham A. Qasse, Manar Abu Talib, Ali Bou Nassif. "Performance Analysis of Hyperledger Fabric Platforms", Security and Communication Networks, 2018 Publication	<1 %
48	Submitted to Singapore University of Technology and Design Student Paper	<1 %
49	Submitted to The British College Student Paper	<1 %
50	Submitted to University of Exeter Student Paper	<1 %
51	Xun Sun, Jian-hua Li, Shu-tang Yang, Gong-liang Chen. "Non-interactive identity-based threshold signature scheme without random oracles", Journal of Zhejiang University-SCIENCE A, 2008 Publication	<1 %
52	"Wireless Algorithms, Systems, and Applications", Springer Nature, 2006 Publication	<1 %
53	Lecture Notes in Computer Science, 1999. Publication	

		<1 %
54	Submitted to University of Essex Student Paper	<1 %
55	Submitted to VIT University Student Paper	<1 %
56	vdoc.pub Internet Source	<1 %
57	"Advances in Cryptology – EUROCRYPT 2016", Springer Science and Business Media LLC, 2016 Publication	<1 %
58	Liehuang Zhu, Zijian Zhang, Chang Xu. "Secure and Privacy-Preserving Data Communication in Internet of Things", Springer Nature, 2017 Publication	<1 %
59	Lu, Yanrong, Lixiang Li, Haipeng Peng, and Yixian Yang. "A biometrics and smart cards-based authentication scheme for multi-server environments : A biometrics and smart cards-based authentication scheme for multi-server environments", Security and Communication Networks, 2015. Publication	<1 %
60	Namita Tiwari, Amit Virmani, Ashutosh Tripathi. "Chapter 7 An Efficient Group	<1 %

	Signature Scheme Based onECDLP", Springer Science and Business Media LLC, 2023 Publication	
61	Submitted to University of Edinburgh Student Paper	<1 %
62	website60s.com Internet Source	<1 %
63	www.tradefi.com Internet Source	<1 %
64	"Information and Communications Security", Springer Science and Business Media LLC, 2011 Publication	<1 %
65	"Wireless Algorithms, Systems, and Applications", Springer Science and Business Media LLC, 2018 Publication	<1 %
66	scholarbank.nus.edu.sg Internet Source	<1 %
67	Jianhong Zhang. "A Novel Efficient Group Signature Scheme with Forward Security", Lecture Notes in Computer Science, 2003 Publication	<1 %

Exclude quotes	On	Exclude matches	< 14 words

Exclude bibliography On