Privacy Preserving Decision-Making over Blockchain

Jiajie Zhang, BSc (Hons)
School of Computing and Communications
Lancaster University

A thesis submitted for the degree of
Doctor of Philosophy

August 26, 2024

I dedicate this thesis to my mother, Fangyuan Hou and my husband, Pierre Ciholas, the one who gave me life and the one who made me free.

Declaration

I declare that the work presented in this thesis is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. This thesis does not exceed the maximum permitted word length of 80,000 words including appendices and footnotes, but excluding the bibliography. A rough estimate of the word count is: 69935

Jiajie Zhang

Privacy Preserving Decision-Making over Blockchain

Jiajie Zhang, BSc (Hons).
School of Computing and Communications, Lancaster University

A thesis submitted for the degree of *Doctor of Philosophy*. August 26, 2024

Abstract

Decentralised decision-making systems are now a crucial building block of various applications. These systems allow participants to collectively decide the direction and outcomes of their projects, reaching decision agreements in a democratic and inclusive manner. In this thesis, we present our work on a comprehensive and inclusive decentralised collaborative decision-making system while ensuring privacy. The system core foundation is a 2-stage voting scheme based on choice architecture principles. Crucially, this decision-making system is built to be compatible with existing blockchains and their infrastructure so it is adaptable and easy to deploy. Furthermore, this system implements liquid democracy and delegative voting, allowing the stakeholders to vote directly on proposals or delegate their voting powers to experts. The combination of wisdom of the crowd and expert knowledge enhances collaborative intelligence, resulting in more informed, and therefore improved decision-making.

To ensure the privacy of voters, our system ensures that even when a minority of the voting committee members are dishonest, it is computationally impossible for any participant to reveal the voting preferences or delegations of voters with a significant probability. These privacy assurances are crucial to preserve the integrity of the decision-making process and protect participants.

Concurrent multiple voting events is an important feature, therefore in this thesis we introduce a distributed batch key generation protocol. This protocol allows participants to generate multiple keys simultaneously, thereby minimizing communication costs with an amortised complexity of $\mathcal{O}(n)$ per key, where n represents the number of participants. Additionally, the system is built to support an evolving committee feature, which allows voting committee members to be changed during the voting process. This ensures that the decision-making process remains adaptable and aligned with the evolving interests and expertise within the blockchain community through flexibility.

We thoroughly analyse the security of our system and demonstrate its resilience under the universally composable (UC) framework. By conducting a deep investigation of previous systems, we identify gaps such as non-private ballots and insecure and/or inefficient voting methods which we address in our proposed system successfully. We validate the efficiency of our proposed system and implement a demonstration written using the programming language Scala. The system is then benchmarked, and the results indicate that our system can effectively handle large numbers of participants, while maintaining high efficiency

throughout the decision-making process.

We believe that this community-inclusive decentralized collaborative decision-making system with privacy assurance contributes to the advancement of blockchain governance. This system also ensures that the principles of decentralization and democratic decision-making are upheld. Our proposed system provides incentives for the participants while also ensuring their active engagement in the decision-making process by using a novel reputation management scheme. Our research provides a practical and efficient solution for blockchain applications requiring transparent and secure decision-making processes.

Publications

Contributing Publications

The thesis is supported by a series of publications that have made substantial contributions to its advancement. Below is a compilation of publications that directly contribute to this thesis:

- **Jiajie Zhang**, Bingsheng Zhang, Andrii Nastenko, Hamed Balogun, and Roman Oliynykov. "Privacy-Preserving Decision-Making over Blockchain." In IEEE Transactions on Dependable and Secure Computing, 2022.
- **Jiajie Zhang**. "Dynamic Scalable Distributed Key Generation and Maintenance on Permissionless Blockchains." Under Submission.

Additional Publications

Throughout my PhD education, I have participated in several publications that, although not directly relevant to my current thesis, have enhanced my research experience. These articles demonstrate my active involvement and cooperation within the academic community, emphasising a wide array of subjects investigated throughout my PhD studies. Below is a compilation of these supplementary publications:

- Ciholas Pierre, Jose Miguel Such, Angelos K. Marnerides, Benjamin Green, **Jiajie Zhang**, and Utz Roedig. "Fast and furious: outrunning Windows kernel notification routines from user-mode." In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 67-88. Springer, Cham, 2020;
- **Jiajie Zhang**, Bingsheng Zhang, and Bincheng Zhang. "Defending adversarial attacks on cloud-aided automatic speech recognition systems." In Proceedings of the Seventh International Workshop on Security in Cloud Computing, pp. 23-31. 2019;
- **Jiajie Zhang**, Bingsheng Zhang, and Jiancheng Lin. "Recessive Social Networking: Preventing Privacy Leakage against Reverse Image Search." In 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 211-219. IEEE, 2019.

Acknowledgements

Completing my thesis signifies a turning point in my academic career, and I am debtful to the numerous people whose constant support, direction, and encouragement has driven me ahead. Their combined efforts have affected not only the nature of my work but also my personal development as a researcher and person.

First of all, I owe most gratitude to Prof. Bingsheng Zhang, my main supervisor. His great understanding and experience in blockchain and cryptography has been absolutely crucial in guiding my work. He laid the fundamental stones of my academic search in this exciting topic and exposed me to the complex universe of zero-knowledge proofs.

I particularly like to thank Dr. Antonios Gouglidis, my second supervisor. Over my PhD, his consistent support and direction has been a lighthouse. His friendliness and approachability have given me the confidence and ease I need to negotiate the most difficult stages of my study. His presence has always been comforting, which helps to ease the difficult research choreography.

A particular thank you for their insightful comments and direction, Prof. Cong Wang and Dr. Muhammad Bilal. Their helpful comments during the viva have really improved my work. I also owe Dr. Haris Rotsos for chairing the viva, an event I consider to be among the most enlightening and educational ones of my academic life.

Working with eminent colleagues-Prof. Roman Oliynykov, Dr. Andrii Nastenko, and Dr. Dymitro Kaidalov-has really improved my experience at IOHK. Their friendship, teamwork, and pragmatic wisdom have been rather inspirational. Rich conversations and shared memories with them have greatly enhanced my study and helped me to link it with practical uses.

I would want to thank every buddy of mine at SCC, B55, and B59. Having your company has made my workplace not only efficient but also fun. One of the pleasures of my PhD has been the time I spend with you.

The constant support of Ovini Gunasekera has been pillar of my PhD path. One great source of strength has been your ability to lighten any circumstance with humour and your ongoing support. As essential to my achievement as the research itself has been, the memories we have produced and the laughter we have shared will always be priceless. One present I will always treasure is your friendship.

Specifically thanks Angela Cheung and Dr. Peter Linwood. My first phase of the PhD was full of difficulties; your unwavering support and encouragement helped me get through those trying circumstances.

My beloved sister and brother, Sissi (Xiaoxi Ding) and Nick (Yi Zhen), have been a continual source of inspiration, providing a welcome break from the demands of academics via their unflinching companionship. You have brought the United Kingdom to feel like home. Your presence has been consoling and joyful, which helps me to remember and find heartwarming my path here.

Thank you also to my EY family: Dr. Duncan Westland, Ilyas Ridhuan, Miranda Wood, Dr. Zac Youell, Dr. Swati Rawal, for their relentless support. One major driving force has been your confidence in my ability. When I consider the great help that has highlighted the last year of my PhD journey-a crucial period full of obstacles and successes-I have to especially thank Ilyas. Beyond simple encouragement, Ilyas's direction reminded me of the particular road each person travels and the need of not measuring oneself against others. He has been the best counsellor, whose insight and support have helped me to regain my self-belief and focus.

Judith Elgie has been a remarkable pillar of strength and inspiration on this road, where academic rigour sometimes blended with personal struggles. When I most needed clarity and perspective, you really have a great capacity to clear the fog in trying circumstances. Thank you Judith!

Remarkable single mother my mother has shown tenacity and commitment, thereby imparting to me the real meaning of will and fortitude. Her insightful advice and caring have been my lighthouse, offering direction and solace amid trying circumstances. Her influence in my life goes well beyond family ties; she is my inspiration, mentor, constant supporter in whatever I do. Thank you Mama, I love you!

To my aunt, Xiuchuan Hou, your faith in my talents and your continuous encouragement have been a great source of strength and inspiration during this demanding academic road.

To my Pierre, your unwavering faith in my abilities and your ability to bring joy and balance into our lives has been transformative. Your presence has not only enriched my days but also infused my academic endeavors with a sense of purpose and fulfillment that transcends the pages of this thesis. In you, I have found not just a partner, but a source of endless encouragement and a beacon of hope, lighting up both my personal path and professional aspirations.

Over my PhD, the great help I received goes beyond personal relationships to my dear feline friends, SHA-1, Shadow, and Meimei. Their presence offers a special kind of support only pets can, and it has been a continual source of pleasure and delight. Their unwavering affection and company have been essential on this road; they are not just the best cats on earth but also treasured parts of my life.

To everyone listed as well as all those who have travelled this road, my sincere thanks are unbounded.

Formless Shapeless, yet it can crash.

Contents

List of Figures

List of Tables

List of Abbreviations

AMT Authenticated Multi-point evaluation Trees.

B2B Business-to-Business.

B2C Business-to-Consumer.

BIP Bitcoin Improvement Proposal.

CBA Committee-Based Assembly.

DBKG Distributed Batched Key Generation.

DCR Decred.

DDH Decisional Diffie-Hellman.

DDoS Distributed Denial-of-Service.

DKG Distributed Key Generation.

DoS Denial-of-Service.

DSKM Dynamic Scalable Distributed Key Management.

E-Voting Electronic Voting.

EIP Ethereum Improvement Proposal.

HIM Hyper-Invertible Matrix.

IK-CPA Indistinguishability of Keys under Chosen-Plaintext Attack.

IoT Internet-of-Things.

ITI Interactive Turing Instance.

ITM Interactive Turing Machine.

JF-DKG Joint-Feldman Distributed Key Generation.

LEG Lifted ElGamal Encryption.

MCD Multi-Collateral Dai.

NIZK Non-Interactive Zero-Knowledge.

OTP One Time Pad.

OVN Open Vote Network.

PB Participatory Budgeting.

PID Participant Identifier.

PKE Public Key Encryption.

PVSS Publicly Verifiable Secret Sharing.

RO Random Oracle.

SHVZK Special Honest Verifier Zero-Knowledge.

SID Session Identifier.

TC Technical Committee.

TSV Two-Stage Voting.

UTXO Unspent Transaction Output.

VRF Verifiable Random Function.

VSS Verifiable Secret Sharing.

ZIP Zcash Improvement Proposal.

ZKP Zero-Knowledge Proof.

Chapter 1

Introduction

> Decentralisation is not a technological construct; it is a mindset, a way of organising, a culture, a way of doing, and a way of living in the world. It is the re-imagining of power, the shifting of boundaries, and the democratisation of decision-making.
>
> Primavera De Filippi, "Blockchain and the Law: The Rule of Code"

1.1 Overview

Emerging as a revolutionary tool, blockchain technology marks a new era of decentralised applications capable of upsetting accepted paradigms in many different fields. Decentralised decision-making systems stand out as especially exciting initiatives in this transforming terrain, ready to change traditional centralised decision-making procedures into more open, participatory, accountable systems. These systems seek to democratise decision-making by using the fundamental ideas of blockchain, therefore empowering a wider range of stakeholders and providing open, safe, and trustless substitutes for group decisions across many spheres. From e-democracy [1] to Internet-of- Things (IoT) information sharing decision-making [2] and beyond, these systems have the ability to transform how communities and businesses come to agreement and distribute funds. Existing methods, then, run against scalability, essentially the capacity to support a rising number of participants/projects, handle rising transaction volumes, adaptability to dynamic participation, and ensuring fair representation of diverse opinions, so impeding their practicality and effectiveness.

This thesis seeks to address these issues and develop a decentralised decision-making system on the blockchain that can accommodate a great number of players, adapt to changing dynamics, and guarantee fair representation of many points of view on the blockchain. We discuss the reasons behind this research in the ensuesing sections, explain its aims and objectives in Section 1.2, highlight the important contributions made by this work in Section 1.3, and show an outline of the structure of this thesis in Section 1.4.

1.2 Problem Statement and Thesis Goals

1.2.1 Current Limitations in Decision-Making Systems

The conventional centralised decision-making processes frequently encounter problems associated with a lack of transparency, potential biases, and limited participation [3]. Centralised decision-making often fails to effectively represent the interests and preferences of all stakeholders, resulting in community disengagement and rejection [4]. Blockchain technology provides a novel solution by facilitating decentralised decision-making that functions on a distributed, transparent, and unchangeable ledger. By utilising the concepts of blockchain, individuals may collaboratively make decisions without depending on a central governing body, promoting a feeling of ownership and inclusiveness.

The exponential growth of decentralised decision-making systems on the blockchain has generated enthusiasm and ingenuity in the field of collaborative decision-making [5, 6, 7]. These solutions have the potential to make decision processes more democratic, decrease centralisation, and improve openness and accountability. As the use of blockchain technology increases, there is a greater need for strong, secure, confidential, and comprehensive decision-making systems. The main difficulties involve creating a system that can efficiently and securely manage a significant number of participants over an extended period of decision-making [8, 9, 10], guaranteeing equal representation of different viewpoints and interests [11, 10, 12], and motivating participants to actively engage in the decision-making process [13, 14, 15, 16].

Scalability is a significant difficulty faced by many existing decentralised decision-making systems. As the number of players increases, these systems frequently struggle with maintaining operational efficiency and reducing transaction costs. The increasing computational and communication burdens can impede the system's efficiency in managing a growing number of participants effectively. Furthermore, in numerous cases, the periods of decision-making are of significant duration, occasionally lasting for up to a month. The lengthy period of this process requires a significant number of participants to either be online or keep confidential information for the entire term, which increases the risk of potential corruption. Although permissionless blockchain systems are inherently decentralised, they should also be able to adapt to changes in participant involvement. Managing the smooth inclusion and exclusion of participants, while maintaining security and fairness, is a complex and significant task.

Democracy fundamentally depends on the inclusion of a wide range of voices and viewpoints. Nevertheless, attaining fair and just representation in decentralised decision-making systems poses significant difficulties. In the absence of suitable processes, these systems can unintentionally distort decision outcomes, showing preference towards specific groups or excessively magnifying the impact of a small number of influential players.

First priorities are the respect of participants' vote privacy and the security of the

decision-making process. Many decision-making systems already in use shockingly lack appropriate protections to safeguard vote privacy. Finding a balance between personal privacy and system security while using blockchain's inherent openness calls for careful thought. Furthermore coexisting in today's vast blockchain ecosystem are several platforms and applications. To effectively use the transforming power of blockchain technologies, security must not be compromised by ensuring perfect compatibility across these different systems and decentralised decision-making protocols.

In summary, decentralised decision-making systems have substantial obstacles that hinder their efficiency and scalability:

- Scalability Challenges: As participant numbers increase, existing systems struggle to maintain performance without incurring prohibitive costs, revealing a critical need for scalable solutions that do not compromise operational efficiency.
- Privacy and Security Concerns: Current systems often fail to adequately protect vote confidentiality and system integrity, making them vulnerable to attacks and reducing trust among users.
- Insufficient Dynamic Participation and Representation: Many systems do not effectively adapt to changes in participant numbers or ensure fair representation, leading to potential biases and unequal influence among stakeholders.

1.2.2 Hypothesis

This thesis argues that by employing an integrated strategy that incorporates dynamic committees, effective cryptographic protocols, and a strong reputation management system, it is possible to significantly address and surpass these limits. The theory is based on the assumption that:

"A multi-faceted architectural design incorporating advanced cryptographic techniques and dynamic governance mechanisms can significantly enhance the scalability, privacy, and equitable representation in blockchain-based decentralised decision-making systems."

1.2.3 Goals and Objectives

The desire to create a complete design strategy and effective cryptographic protocols for a decentralised decision-making system able to overcome the constraints previously drives this thesis. By carefully tackling these issues, the intention is to equip communities, businesses, and stakeholders the capacity to participate in fair and effective group decision-making, all within the limits of a trustless and decentralised paradigm.

Consequently, the fundamental goal of this thesis is to conceive, construct, and actualise a decentralised decision-making system placed inside the blockchain domain, which skilfully

negotiates the complex array of obstacles described before. This proposed system seeks to provide an atmosphere where a varied assembly of participants can engage in effective, private, and safe decision-making, all the while firmly maintaining the cardinal values of openness and inclusivity.

Given this background, this thesis aims to meet and finally conquer these challenges, so helping to produce decentralised decision-making systems. Examining the complicated junction between technical innovation, theoretical constructions, and pragmatic application helps the thesis to provide pragmatic solutions that boost the practicability, efficiency, and security of these systems. By means of a thorough research of cryptography techniques, architectural design principles, and governance paradigms, the aim is to build an all-encompassing framework that not only powers an array of stakeholders but also promotes cooperative intelligence inside the decision-making process. Using the potential inherent in blockchain technologies, this thesis aims to start the dawn of a new era distinguished by transparent, detailed, public verifiable decision-making techniques across a wide spectrum of fields.

The overarching goals of this research are framed to demonstrate the practical implementation and effectiveness of these solutions:

- Demonstrate Scalability: Prove that the system can scale to handle thousands of participants with minimal impact on transaction costs and system performance.
- Enhance Privacy and Security: Validate the effectiveness of cryptographic protocols in safeguarding voter privacy while ensuring the integrity and transparency of the decision-making process.
- Promote Equitable Representation and Dynamic Adaptation: Show how the system can adapt to participant turnover and activity changes, ensuring continuous fair representation and engagement.

The more specific objectives of this thesis are as follows:

- Scalability Enhancement: The primary objective of this research is to develop innovative architectural paradigms and technological solutions that enhance the scalability of decentralised decision-making systems. It is essential that the system is able to accommodate the increasing number of participants without compromising efficiency or incurring exorbitant transaction costs as the number of participants increases.
- Facilitating Dynamic Participation: The decentralised nature of blockchain systems promotes flexible participant engagement; however, this dynamic involvement requires seamless transitions as individuals enter or depart the decision-making process. The objective of this research is to create mechanisms that ensure the integrity of the

decision-making process in the face of the fluidity of participation dynamics, thereby facilitating seamless transitions for participants. The objective is to guarantee that the system remains inclusive and accessible as participants' roles change.

- Equitable Representation Empowerment: Ensuring fair and equal representation of a wide range of ideas is of utmost importance in upholding the principles of democracy. This study aims to provide strategies that guarantee equitable and impartial representation in the proposed decision-making system. The aim is to create an atmosphere where choices are made collectively and are representative of the wider community by using methods that prevent individual groups from having too much influence and avoid power being concentrated in a few hands.

- Privacy-Conscious Decision-Making: This research places significant emphasis on maintaining a delicate balance between protecting the privacy of participants and ensuring the security of the decision-making process. Ensuring the confidentiality of participants' votes and strengthening the security of the system are key components of the suggested approach. The objective is to establish a secure and reliable decision-making environment by combining cryptographic approaches with strong security protocols.

- Interoperability Harmonisation: The intricate terrain of blockchain technology encompasses a plethora of platforms and applications. It is crucial to have smooth compatibility between the decentralised decision-making system and the wider blockchain ecosystem, while also prioritising security, in order to fully realise its potential. The objective of this research is to create an architecture that can seamlessly interact with current blockchain projects, therefore improving the ability of blockchain technology to collaborate across several fields.

By addressing these critical issues through innovative technological contributions, this thesis aims to advance the state of decentralised decision-making systems, making them more robust, secure, and inclusive for diverse applications.

1.3 Contribution statements

This thesis makes several substantial contributions to the field of decentralised decision-making systems on the blockchain. The overarching hypothesis of this work is thereby supported by the fact that each contribution is intended to resolve specific challenges identified in the problem statement.

- For Scalability: The thesis directly addresses the scalability issue by introducing a distributed batched key generation technique that substantially reduces the computational and communication overhead associated with large participant bases.
- For Privacy and Security: We suggest a two-stage voting mechanism that is augmented with non-interactive zero-knowledge (NIZK) proofs for the purposes of privacy and security. This configuration guarantees the privacy of ballots and their verifiability, thereby improving the system's security and trust.
- For Dynamic Participation and Representation: The reputation management scheme, in conjunction with the evolving committee mechanism, guarantees that the system adapts dynamically to changes in participation while preserving fairness and inclusivity.

In the following section, we delve deeper into the specific contributions of this thesis, each designed to address distinct challenges within decentralised decision-making systems. These contributions not only demonstrate the practical implementation of our theoretical frameworks but also showcase their impact on enhancing scalability, privacy, and dynamic participation. Detailed below are the pivotal advancements made through this work:

- **Systemic Design:**
  - **Contribution:** We propose a comprehensive systemic design for decentralised decision-making, encompassing the pre-voting, voting, and post-voting epochs. We support participatory budgeting in the decision-making system. Namely, proposals, voters and experts are associated with tags/fields. Each field has its own fixed budget, the shortlist and winning proposals are tallied independently of the other fields. Taking blockchain development funding as an example [17], the fields will at lease have Marketing (*e.g.*, activities like conference and advertisement for marketing growth), Technology adoption (*e.g.*, platform integration), Development and security (*e.g.*, security incident response), Organisation and management (*e.g.*, team coordination), Support (*e.g.*, user support and documentation) and General (*e.g.*, charity).
  - **Novelty:** Unlike existing models, this system design facilitates seamless interactions between participants at each stage, enhancing fairness, transparency, and efficiency. The inclusion of participatory budgeting is particularly innovative,

as it allows stakeholders to allocate resources directly, reflecting a truly democratic process within decentralised frameworks.

- **Evolving Committee Mechanism:**
  - **Contribution:** We introduce an evolving committee mechanism [18] during the distributed key generation process to allow the system to adapt dynamically to participant turnover and maintains security through periodic reassignment of keys. Periodically, a new committee will be selected by cryptographic sortition [18, 19], and the secret keys are re-shared to the new committee while keeping the public keys unchanged. This mechanism addresses the challenge of dynamic participation by enabling the system to handle changes in the committee's composition. The evolving committee enhances system resilience and fault tolerance against changing participation dynamics and potential malicious behaviours in long decision-making period.
  - **Novelty:** This approach addresses the dynamic nature of participant engagement in decentralised systems. It enhances system resilience and fault tolerance against changing participation dynamics and potential malicious behaviours over extended periods.
- **Reputation Management Scheme:**
  - **Contribution:** We design a reputation management scheme that objectively evaluates participants' contributions and behaviours across different roles and fields. This scheme incentives active engagement and diverse participation, ensuring a balanced and robust decision-making ecosystem. By providing reputation scores associated with field labels, participants are encouraged to contribute their expertise to various decision-making domains, fostering a rich and dynamic environment.
  - **Novelty:** The scheme introduces a method for quantitatively assessing and rewarding contributions in a decentralised setting, which is crucial for maintaining participant engagement and ensuring system integrity.
- **Efficient Cryptographic Protocols:**
  - **Contribution:** We develop efficient cryptographic protocols, such as the Distributed Batched Key Generation (DBKG) protocol and Two-Stage Voting (TSV) scheme, to ensure privacy, security, and verifiability while minimising communication overhead and execution times. These protocols strike a balance between cryptographic security and practical scalability, making them suitable for real-world decentralised decision-making scenarios. We analyse the security

of our proposed protocols under the Universal Composability (UC) framework [20]. Therefore, all the proposed protocols can be used with other UC secure protocols without lossing security.

Our DBKG protocol can generate multiple keys simultaneously, achieving amortised complexity of $\mathcal{O}(n)$ per key. The TSV scheme significantly reduce the community's voting effort. In the first stage (preferential voting), the voters and experts announce their preferences to specify their preferred proposals. Depending on fund availability, a shortlist will be produced. In the second stage (threshold voting), the voters and experts vote on each shortlisted proposal for YES/NO/ABSTAIN; those proposals that receives more than threshold supports will be funded. Besides that, the batched either 1 or 0 zero knowledge proof to validate the encrypted ballots from voters and experts reduced the communication costs. We add non-interactive zero-knowledge (NIZK) proofs to make the ballots ciphertext publicly verifiable by leveraging the sigma protocol and the Fiat-Shamir heuristic.

– **Novelty:** These protocols offer a new level of efficiency and security in decentralised decision-making. The DBKG protocol, in particular, reduces the complexity of key generation and distribution, a critical improvement over existing solutions.

- **Prototype Implementation and Evaluation:**

  – **Contribution:** We create a fully functional prototype of the proposed decision-making system and conduct extensive evaluations to demonstrate its practicality, efficiency, and scalability in real-world scenarios. The prototype serves as a proof of concept and showcases the effectiveness of the proposed design and cryptographic protocols in a tangible implementation.

  The implementation of the proposed decision-making system is in Scala programming language for benchmarking in the real world environment. The main functionalities covered include proposal submission, voters/experts registration, voting committee members and their corresponding deposit lock, randomised selection of the voting committee members among voters, distributed key generation ballots casting, joint decryption of tally with recovery in case of faulty/malicious committee members, reward payments, deposit paybacks and penalties for faulty actors. All implemented protocols are fully decentralised and are resilient up to 50% of malicious participants. We launched a testnet comprising a dozen full nodes successfully operating hundreds of polling periods with different parameters.

  – **Novelty:** The prototype's successful deployment on a testnet, involving multiple nodes and polling periods, showcases the real-world applicability of the proposed

system, providing tangible evidence of its advantages and robustness.

Through the advancements presented in this thesis, we aim to significantly advance the state-of-the-art in decentralised decision-making systems on the blockchain. Our approach not only supports the underlying hypothesis but also expands upon current research by offering innovative, practical solutions to complex challenges faced in real-world applications. By skillfully integrating advanced cryptographic techniques with novel governance models, this work lays a robust foundation for the continued evolution and implementation of blockchain technologies.

The contributions detailed herein enable a future where decision-making processes are not only transparent and participatory but also inherently secure. This enhances trust and fosters a stronger consensus among diverse stakeholder groups. As we leverage the unique capabilities of blockchain and sophisticated cryptographic methods, we anticipate a paradigm shift toward more democratic and efficient decision-making frameworks across various domains.

This thesis not only demonstrates the viability of these sophisticated systems in theoretical and simulated environments but also emphasises their practical applicability. The proposed models and technologies promise to revolutionise the way decisions are made, managed, and respected in a blockchain context, ultimately contributing to a more equitable and responsive digital society.

In the following chapters, we will delve into the details of the proposed design, cryptographic protocols, and evaluation results, providing a comprehensive understanding of our approach's effectiveness and potential. Additionally, we will discuss the implications of our research and outline future directions to further advance the field of decentralised decision-making systems on the blockchain.

1.4 Thesis outline

As illustrated in Figure 1.1, this thesis is structured into several key chapters, each addressing specific aspects of the research and contributing to the overall goal of designing a robust and efficient decentralised decision-making system on the blockchain.

- Chapter 2. Preliminaries - Cryptography Tools

  In this foundational chapter, we lay the groundwork by introducing the necessary cryptographic tools and concepts used throughout the thesis. We explore various cryptographic techniques, such as encryption, security analysis, and zero-knowledge proofs, which form the building blocks of our proposed decentralised decision-making system.

- Chapter 3. Related Work - Reviewing Existing Designs

  To gain insights into the current state of decentralised decision-making systems, this chapter reviews and analyses existing designs and approaches. We examine the strengths and limitations of various systems, including treasury systems, e-voting protocols, and blockchain governance models. By studying related work, we can identify gaps and opportunities for improvement in our proposed system.

- Chapter 4. System Design - Overview of the Proposed System

  In this pivotal chapter, we present an in-depth overview of our proposed decentralised decision-making system on the blockchain. We outline the fundamental elements, such as the two-stage voting scheme, liquid democracy features, and reputation management scheme. By providing a holistic view of the system design, we set the foundation for the subsequent chapters that explore each component in detail.

- Chapter 5. Building Block: Distributed Key Generation - Methods for Generating Distributed Keys

  To ensure secure and private voting in our system, this chapter focuses on the distributed key generation (DKG) process. We investigate different methods for generating distributed keys, enabling efficient and robust encryption of ballots. By exploring DKG protocols, we lay the groundwork for a trustworthy and tamper-resistant voting process.

- Chapter 6. Building Block: Two Stage Voting Scheme - Demonstrating the Voting Process

  In this chapter, we delve into the intricacies of our two-stage voting scheme, inspired by choice architecture principles. We explain how voters and experts participate in the decision-making process, cast their ballots, and validate the voting results. By

presenting the voting scheme, we demonstrate the transparency and verifiability of our proposed system.

- Chapter 7. Building Block: Evolving Committee - Changing Participants during Protocol Execution

  One critical aspect of our decentralised decision-making system is the ability to accommodate dynamic participation. In this chapter, we outline the evolving committee mechanism, allowing for the addition and removal of participants during the protocol execution. By ensuring the system's adaptability, we promote inclusivity and diverse representation.

- Chapter 8. Building Block: Reputation Management Scheme - Managing Entity Reputation

  Reputation is a vital aspect of any decision-making system, influencing participants' roles and influence. In this chapter, we introduce a reputation management scheme that objectively evaluates participants' contributions and behaviours across different roles and fields. By implementing a comprehensive reputation system, we encourage active and honest participation in the system.

- Chapter 9. Implementation and Performance - Evaluating Proposed Methods

  To validate the practicality and efficiency of our proposed system, this chapter focuses on the implementation and performance evaluation. We conduct benchmark tests and analyse the cryptographic protocols' execution time and communication costs. By assessing the system's performance, we gain valuable insights into its real-world viability.

- Chapter 10. Conclusion and Future Works - Summarising Contributions and Future Directions

  In this final chapter, we provide a comprehensive summary of the contributions made in this thesis. We reflect on the achievements and challenges encountered during the research and discuss potential future directions for advancing decentralised decision-making systems on the blockchain. By offering a conclusive outlook, we emphasise the importance of continuous research in this promising field.

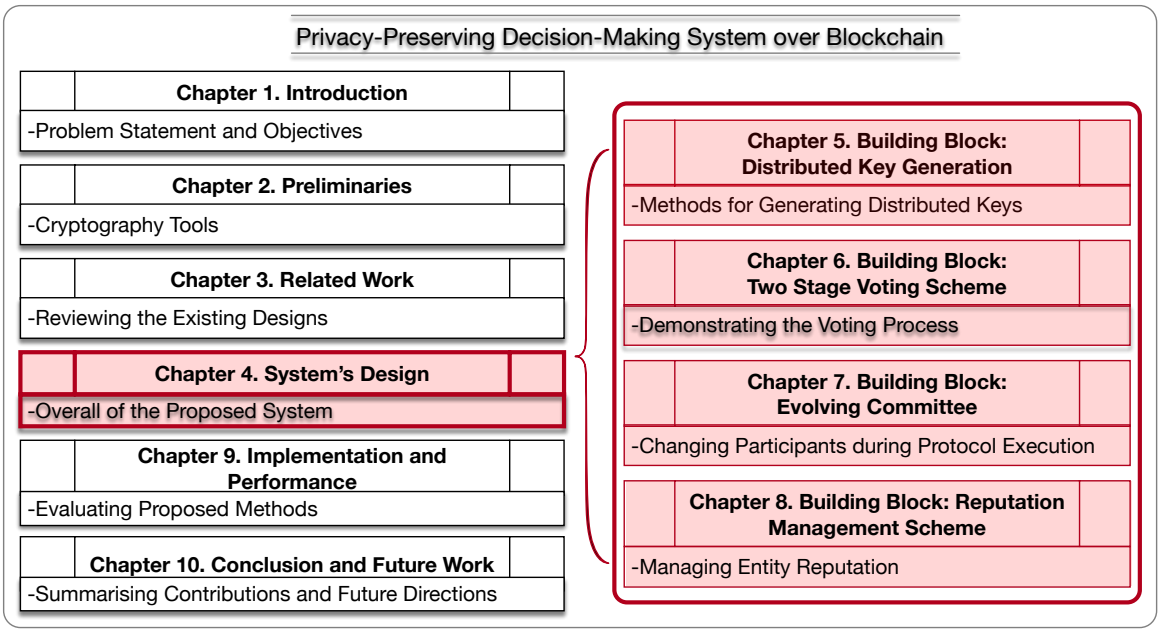

Figure 1.1: Thesis Road Map

Chapter 2

Preliminaries

> There is no security on this earth; there is only opportunity.
>
> General Douglas MacArthur

2.1 Overview

This chapter presents the cryptographic building blocks and key concepts. Section 2.2 defines important notations that will be used throughout the thesis. The security analysis of the proposed protocols is analysed under the Universal Composability (UC) framework [20], which is further elaborated in Section 2.3. This section also provides a definition of UC security and offers an intuitive understanding of its principles. Public-Key Encryption [21] as the main encryption scheme in decision-making systems is formalised in Section 2.4, including Lifted Elgamal Encryption. Section 2.5 focuses on Commitment schemes, particularly Pedersen Commitment[22]. Commitment schemes are essential in secret sharing and zero-knowledge proofs. In this section, the Pedersen Commitment scheme is explained in detail, highlighting its significance in cryptographic applications.

Section 2.6 explains the concept of Secret Sharing, which is utilised in Distributed Key Generation to distribute key shares among participants. The section covers various Secret Sharing schemes, including Shamir's Secret Sharing[23], Verifiable Secret Sharing[24], and Hyper-Invertible Matrix based Secret Sharing[25]. Section 2.7 delves into Zero-Knowledge Proofs, a technique used to prove a statement without revealing any additional information. This section introduces different aspects of Zero-Knowledge Proofs, including the Σ-protocol, Special Honest Verifier Zero-Knowledge, Non-Interactive Zero Knowledge, Proof of Knowledge, and the Schwartz-Zippel Lemma. The aim is to provide a comprehensive understanding of the various methods and techniques used in Zero-Knowledge Proofs, as

outlined in [26].

Section 2.8 focuses on Blockchain technology, which serves as the broadcast channel and peer-to-peer channel in various protocols. This section provides a standardised explanation of Blockchain properties, the Blockchain model, and Blockchain functionality. The goal is to establish a clear understanding of how Blockchain operates within cryptographic protocols. Section 2.9 demonstrates Distributed Key Generation and its application in generating encryption keys, particularly in the context of Discrete-Log (DLog) based cryptographic systems. Lastly, Section 2.10 clarifies the concept of Cryptographic Sortition, which is used for self-election. Additionally, it introduces the Verifiable Random Function, which is employed to realise cryptographic sortition effectively.

2.2 Notations

In this section, we introduce various notations and symbols used throughout this thesis. We start by defining the set of natural numbers as $\mathbb{Z}_N$, which includes numbers from 0 to $N-1$, where N is greater than 1. The $:=$ symbol indicates assignment, where$A := B$ means that A is initialised with the value of B, while $A = B$ indicates that A is equivalent to B.

We use indexed labels, such as $a^{(x)}$, to denote values indexed by a label x, and a^x represents the value of a power of x. A set with size n is denoted as $\{a_1, \ldots, a_n\}$, and $\langle a_1, \ldots, a_n \rangle$ is a set with fixed order. The notations $[a, b]$ and (a, b) represent intervals. $[a, b]$ includes continuous integers between a and b, $[b]$ abbreviates $[1, b]$ (for example, $[3] = 1, 2, 3$), and (a, b) represents either continuous or discontinuous integers between a and b, (for example, $(1, \ldots, 3)$ can be either $\{1, 3\}$ or $\{1, 2, 3\}$). $\mathbf{V}^{[\ell]} := \{x_1, \ldots, x_\ell\}$ is a vector of length ℓ, and $[\ell]$ can be omitted for simplicity. A matrix with m rows and n columns is represented as $\mathbf{M}^{[m \times n]}$.

Given an invertible element b such that $bx \equiv 1 \pmod{N}$ with a unique inverse x, where $\mathsf{gcd}(b, N) = 1$, $(\mathbb{Z}_N)^*$, we define $(\mathbb{Z}_N)^*$ as the set $\{b \in \{1, \ldots, N-1\} | \mathsf{gcd}(b, N) = 1\}$, which consists of all elements b in the range from 1 to $N-1$ that are coprime with N. Denote a finite set as $\mathbb{F}$. When we say $x \leftarrow \mathbb{F}$, it means that x is uniformly and randomly selected from $\mathbb{F}$, with each element having a probability of $\Pr[x = f] = 1/|\mathbb{F}|$ for all $x \in \mathbb{F}$. Furthermore, $\mathbb{F}$ can be efficiently sampled. The statistical distance between two random variables a and b from $\mathbb{F}$, denoted by $\Delta(a, b) = 0$, is calculated as $\Delta(a, b) := 2^{-1} \cdot \sum_{c \in \mathbb{F}} |\Pr[a = c] - \Pr[b = c]|$ for all $c \in \mathbb{F}$. If $\Delta(a, b) = 0$, it means that a and b are considered equally distributed random variables.

The participants are represented as a set $\mathsf{P}^{(r)} = \{P_1^{(r)}, \cdots, P_i^{(r)}\}$, which consists of i Interactive Turing Machines (ITMs), where r is a time-related index. $P_{k \in [i]}^{(r)}$ is used to denote both the machine itself and its identity. Throughout this thesis, the security parameter is denoted by κ. An event is said to be negligible (as defined in Definition 1), if its probability of occurrence is smaller than the inverse of any polynomial function of κ.

Definition 1 (Negligible Function, [21])**.** *A function,* $\mathsf{negl}(\kappa)$*, from the natural numbers to the non-negative real numbers is negligible, if for all sufficiently large* κ *and every* $c > 0$*, it holds that* $\mathsf{negl}(\kappa) \leq 1/(\kappa^c)$.

2.3 Universal Composability

The Universal Composability (UC) Framework, introduced by Canetti in [20], is employed to establish the security of protocols. It ensures the indistinguishability of the environment $\mathcal{Z}$'s views between the real-world execution and the ideal-world execution. In the UC framework, an ideal functionality is utilised to represent the desired security properties of a protocol. This functionality can be seen as an entirely honest third party that carries out

the task ideally. The ideal functionality explicitly captures the influence of adversaries and the knowledge they can gain from the protocol's real-world execution. By comparing the real-world execution with the ideal-world execution, the security of the protocol can be rigorously proven.

Composable security is a crucial property that ensures the protocols remain secure even when combined with other protocols or used as building blocks in larger systems. The security of a protocol should not depend on the specific details of how it interacts with other protocols but rather on its individual security guarantees in isolation. By achieving composable security, cryptographic protocols can be confidently used as building blocks to construct more complex systems without compromising their security properties. This property allows for a modular and scalable approach to design secure systems, as each component can be analysed and proven secure independently, and their security guarantees hold when they are composed together.

2.3.1 Composable Security

Prior cryptographic protocols typically define security in a stand-alone setting, considering only a single execution of the protocol. However, this approach fails to address potential threats that may arise in real-world execution environments where multiple protocols interact with each other. As a result, the initial security definitions need to be extended to encompass these new challenges and considerations in the context of a more complex execution environment. For example, in the case of encryption, the semantic security definition [27] has been augmented to include additional security against chosen ciphertext attacks [28]. Similarly, in the context of zero-knowledge protocols [29], the definition has been extended to handle "resetting" attacks [30].

While enhancing security by adding new security features can address specific threats, it often leads to more complex security definitions. As a consequence, the security analysis becomes more intricate, and the verification of security properties becomes more challenging. Moreover, these complex security definitions are often limited to known security properties and specific execution conditions. They may not fully capture all potential threats or vulnerabilities that arise in real-world scenarios. As the landscape of cryptography and cybersecurity continuously evolves, new security challenges may emerge, necessitating further extensions to existing security definitions or even the creation of entirely new ones.

Figure 2.1 illustrates the negative impact on security when combining the Needham-Schroeder Key Exchange [31] and the One Time Pad (OTP, [32]) protocols. In this scenario, two parties, P_1 and P_2, intend to use OTP with a key generated by the Needham-Schroeder Key Exchange to encrypt a message. In Figure 2.1-(a), P_1 and P_2 execute the Needham-Schroeder Key Exchange protocol, generating a secure key, b. Due to identity checks and public encryption, b remains untamperable. Moreover, the attacker can only obtain the encryption of the generated key, denoted as C, ensuring the secrecy of b. In Figure 2.1-(b),

P_1 encrypts a message M (where $M \in M_1, M_2$) with the key b, computed as $D := M \oplus b$. P_2 can retrieve the original message M by computing $M := D \oplus b$.

At first glance, these protocols may seem secure because the Needham-Schroeder Key Exchange is secure and the OTP is unconditionally secure [33]. However, Figure 2.1-(c) demonstrates an attack on this protocol. Suppose there is an attacker capable of intercepting the ciphertext, $D := \mathsf{Enc}_b(M)$, sent by P_1. The attacker can append D with M_1 by computing $D' := D \oplus M_1$. If $M = M_1$, then $D' = b$, otherwise $D' \neq b$. As a result, the original message can be guessed by the attacker. Hence, the composition of a secure Needham-Schroeder Key Exchange protocol with a secure OTP protocol is, in fact, insecure. This example highlights the importance of carefully analysing the security implications when combining cryptographic protocols, as the security of the resulting composition may not be as strong as expected. It emphasises the need for comprehensive security analysis in multi-protocol environments to avoid vulnerabilities and attacks like the one demonstrated in Figure 2.1.

Overall, addressing security in multi-protocol and multi-environment settings is an ongoing challenge in cryptography. Researchers strive to strike a balance between enhancing security to cover emerging threats while maintaining simplicity and generality in the security definitions to ensure their broad applicability across various cryptographic protocols and execution scenarios.

Indeed, the Universal Composability (UC) framework, proposed by Canetti in [20], addresses the challenges of security when combining cryptographic protocols. The UC framework ensures the security of any UC-secure protocol when composed with other UC-secure protocols, providing strong guarantees for secure composition. The three key properties of the UC framework are as follows:

- Formulation of Security: The UC framework offers a comprehensive approach to formulate the security of cryptographic interactive protocols. It provides a unified and formal language to describe and analyse the security properties of protocols in a modular and composable manner.

- Individual Analysis and Design: Under the UC framework, each cryptographic protocol is analysed, designed, and proven in isolation. This means that the security of a protocol can be established independently of its interactions with other protocols. Protocols can be rigorously assessed for security properties without the need to consider all possible combinations with other protocols.

- Secure Composition: One of the most significant advantages of the UC framework is that it guarantees secure composition. This means that when UC-secure protocols are combined, their security properties are preserved, and the resulting composite system remains secure. The UC framework ensures that the security of the combined system does not rely on specific details of individual protocols but rather on their individual UC security properties.

By adhering to the principles of the UC framework, cryptographic protocols can be designed, analysed, and combined with confidence in various environments. This approach provides a solid foundation for building complex and secure systems by focusing on individual protocol security. It simplifies the analysis and design process, making it more manageable and reliable. As a result, the UC framework significantly advances the field of secure protocol composition and contributes to the development of robust and dependable cryptographic systems.

The UC framework guarantees strong composability, meaning that if a protocol is proven secure under the UC framework, its security remains preserved even when executed with other protocols. This property is formally stated in Theorem 1. In other words, when designing a protocol, one can directly use another UC functionality as an oracle, which can be replaced by the protocol that realises that functionality.

Overall, the UC framework revolutionises the way cryptographic protocols are analysed and designed. It provides a unified and formal language for defining security, ensuring that protocols can be securely combined and used as building blocks in larger systems. As a result, the UC framework greatly enhances the reliability and security of cryptographic systems, making them more resilient to potential attacks and providing a solid foundation for building secure and trustworthy systems in various real-world applications.

Theorem 1 (Composition Theorem, [20])**.** *For any protocol, ρ, with the realised functionality, $\mathcal{G}$, in the $\mathcal{F}$-hybrid model, $\mathcal{F}$ may be used as a subroutine. The composed protocol, ρ^{π}, replacing $\mathcal{F}$ with a secure protocol, π, also securely realises $\mathcal{G}$ in the real model.*

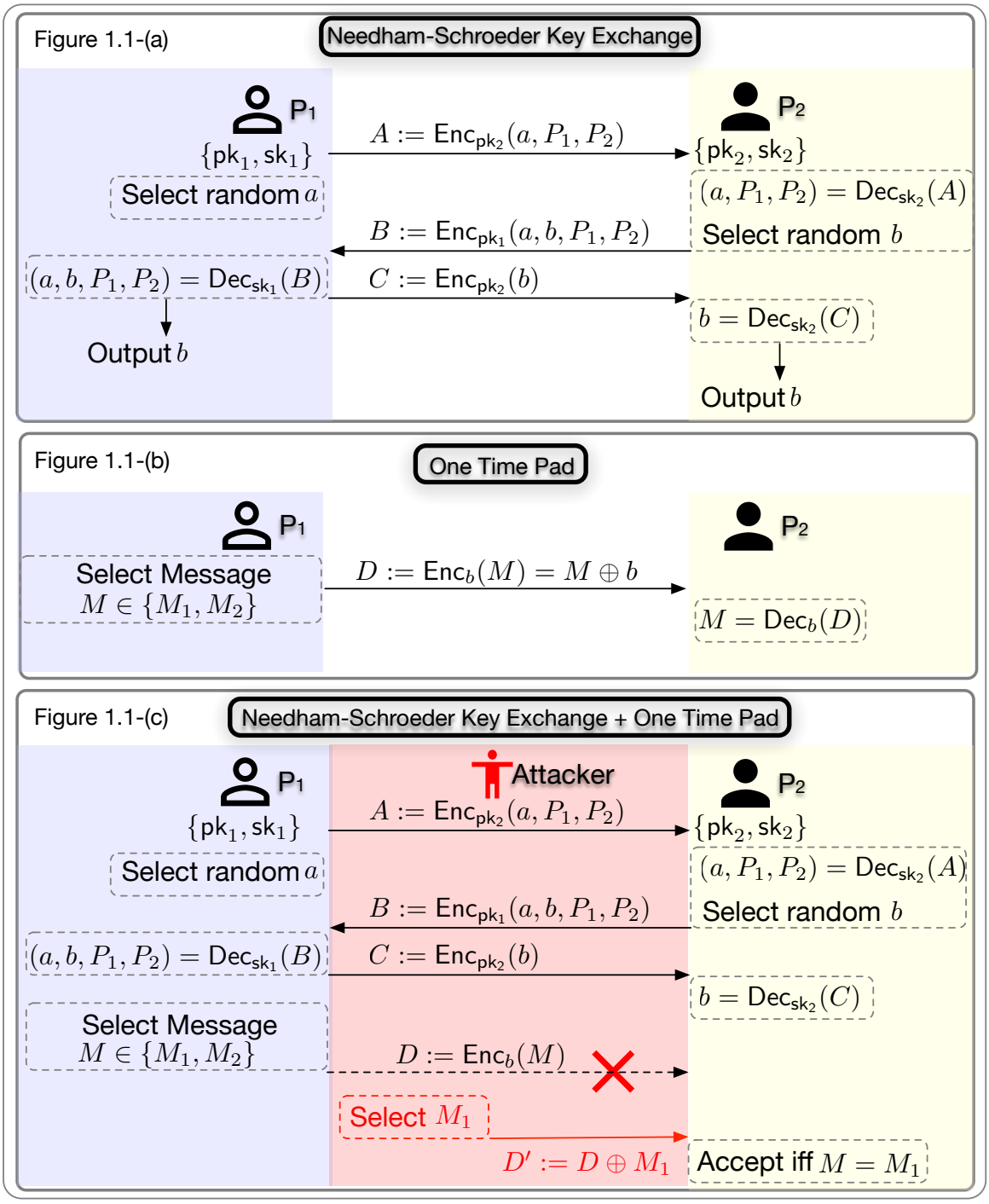

Figure 2.1: Composition Attack Example.

2.3.2 Intuition of Composable Security

The Universal Composability (UC) framework operates by defining two parallel worlds: the Ideal World and the Real World. The main goal is to demonstrate the indistinguishability of these two worlds, ensuring that the real protocol is as secure as the ideal functionality under all possible environments. The role of the Distinguisher, also known as the Environment, is to model everything happening in the universe apart from the protocol execution. In the Ideal World, a trusted third party known as the Ideal Functionality is created to perform the desired computation based on the functional requirements of the protocol. The Ideal Functionality interacts with the Ideal Adversary (Simulator), the Environment, and "dummy" participants whose role is to simply forward or pass messages between the Ideal Adversary and the Environment. In contrast, in the Real World, the actual protocol is implemented and interacts with real participants, the real Adversary, and the Environment. The Environment observes the interactions between the protocol, participants, and the real Adversary.

The security of the protocol is proven by demonstrating that the Environment cannot distinguish between the Ideal World and the Real World better than random guessing, regardless of the possible strategies employed by the real Adversary. This means that the protocol remains secure. If the two worlds are indistinguishable and the protocol's security holds for all possible environments, then the protocol is considered UC-secure. This property ensures that the protocol can be safely composed with other UC-secure protocols without compromising its security guarantees. By employing this rigorous approach, the UC framework provides a robust method for proving the security of cryptographic protocols and verifying their composability in real-world systems.

Achieving and proving composable security of a cryptographic protocol involves the construction of an ideal functionality that performs the ideal/secure version of the real protocol, while ensuring it leaks no extra information beyond what is defined by the protocol. The ultimate goal is to demonstrate that the security of the real protocol is at least as strong as the security of the ideal functionality. To achieve composable security, the following steps are typically taken:

- Ideal Functionality Construction: The first step is to construct an ideal functionality that represents the desired security properties of the real protocol. This ideal functionality serves as an "ideal world" counterpart to the real protocol. It should embody the most secure version of the protocol and should not reveal any additional information beyond what is explicitly defined by the protocol.

- Leakage Simulation: The next step involves demonstrating that the information leaked by the ideal functionality can be simulated during the execution of the real protocol. This means that any information that can be learned from the ideal functionality in the ideal world can also be learned from the execution of the real protocol in the real world.

- Proving Security: Finally, the security of the real protocol is proven by showing that no more information can be learned from the protocol than from the ideal functionality. Since the ideal functionality is designed to be secure, any information leakage from the real protocol should be no greater than what can be learned from the ideal functionality.

By following these steps and establishing the equivalence between the real protocol and the ideal functionality, the composable security of the protocol is demonstrated. This approach provides a rigorous and formal framework for proving the security of cryptographic protocols in a modular and composable manner. It ensures that the security of the protocol is not compromised when it is used in combination with other protocols, making it a powerful tool for designing and analysing secure cryptographic systems.

In the UC framework, for any possible PPT real/hybrid world adversary, $\mathcal{A}$, an ideal world PPT simulator, S, is constructed to present an indistinguishable view to the environment, $\mathcal{Z}$. The real world protocol is denoted by Π and modelled by n Interactive Turing Machines (ITMs, also known as participants) denoted by $\mathsf{P} := (P_1, \cdots, P_n)$, and the adversary, $\mathcal{A}$ (also an ITM).

Interactive Turing Machines (ITMs) represent the static objects or programs in the protocol. During the system execution, the interactions of ITMs are defined by instances of ITMs, referred to as ITI (Interactive Turing Machine Instance). Each ITI possesses a unique identifier to distinguish it from other ITIs for the same ITM in the same system. This unique identifier is composed of two components:

- Session Identifier (SID): The SID represents the particular instance of the ITI. It is used to track and identify a specific run or session of the protocol. Each time the protocol is executed, a new session is created with a unique SID.

- Participant Identifier (PID): The PID indicates the specific participant involved in the protocol instance. It identifies which ITM is acting as a participant in a given session.

The combination of SID and PID serves as a unique identifier for each ITI during the protocol execution. This unique identifier is essential for distinguishing between different instances of ITIs associated with the same ITM and participant in the system. By having a unique identifier for each ITI, the UC framework can accurately track and manage the interactions among participants and the environment throughout the protocol's execution. It ensures that the state and behaviour of each ITI are correctly accounted for, preventing any confusion or ambiguity that could arise when multiple instances of the same ITM are involved in the protocol. The use of the SID and PID also facilitates the proper execution and operation of the UC framework, as it enables clear identification and differentiation of individual instances of ITIs. This precise tracking is crucial for ensuring the correctness and security of the protocol, as well as for providing a reliable foundation for the security analysis and composability guarantees of cryptographic protocols within the UC framework.

In the ideal world of the UC framework, there exists an ideal functionality, denoted as $\mathcal{F}$, which is treated as an honest participant. $\mathcal{F}$ interacts with other participants securely and faithfully, adhering to the specifications defined by its functionality design. It acts as a trusted third party that performs the desired computation based on the functional requirements of the protocol. In this ideal world, the available communication resources for the participants during protocol execution are precisely defined by the input and output behaviours of $\mathcal{F}$. The interactions with $\mathcal{F}$ are completely secure and trusted, ensuring that all participants can communicate and exchange information in a reliable and confidential manner.

The ideal world is considered as an absolutely secure environment, free from any external threats or vulnerabilities. In cases where the adversary gains control over some of the participating participants, it can only access the internal states of the corrupted participants. The adversary's control is limited to reading or modifying the internal state of these corrupted participants, as allowed by the adversary's adversarial power. By having this ideal world with a trusted functionality and secure communication, the UC framework provides a reference point for evaluating the security of the real-world protocol. The goal is to show that the real-world protocol behaves in a way that is indistinguishable from the ideal functionality, even when facing adversaries and potential attacks. This ensures that the protocol remains secure and performs as expected in a wide range of scenarios, allowing for composability with other protocols while maintaining its security guarantees.

The objective of proving security under the UC framework is to design a protocol, denoted as Π, that behaves exactly like the ideal functionality, $\mathcal{F}$, even in the presence of the adversary, $\mathcal{A}$. The goal is to show that Π realises the functionality $\mathcal{F}$ securely. To prove this, an environment, $\mathcal{Z}$, is introduced, which can provide inputs to all participants in the protocol represented by the set P. The environment, $\mathcal{Z}$, is allowed to interact with both the real execution and the ideal execution of the protocol.

In the real execution, the protocol Π is executed involving all participants in P and the adversary $\mathcal{A}$. The environment, $\mathcal{Z}$, observes the outputs produced by this real execution. On the other hand, in the ideal execution, the ideal functionality $\mathcal{F}$ is executed with dummy participants and the ideal adversary S. The environment, $\mathcal{Z}$, also interacts with the ideal execution and observes the outputs. The goal of the security proof is to ensure that the environment, $\mathcal{Z}$, cannot distinguish between the outputs received from the real execution and those received from the ideal execution with a significant advantage. Formally, this is expressed as:

$$\mathsf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}} := \mathsf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}}(\kappa, z), \tag{2.1}$$

where z is the input polynomial in security parameter, κ. The ideal execution is the execution of the ideal functionality, $\mathcal{F}$, involving dummy participants, P, and the ideal adversary, S:

$$\mathsf{EXEC}_{\mathcal{F},\mathsf{S},\mathcal{Z}} := \mathsf{EXEC}_{\mathcal{F},\mathsf{S},\mathcal{Z}}(\kappa, z). \tag{2.2}$$

By demonstrating this indistinguishability property for all possible environments, adversaries,

and ideal adversaries, the protocol Π is proven to securely realise the ideal functionality $\mathcal{F}$ in the UC framework. This proof ensures that the protocol is secure and behaves as expected, even when facing potential attacks and adversarial influence, making it suitable for composability and secure integration into larger cryptographic systems.

In the UC setting, a protocol Π *UC-realises* a functionality $\mathcal{F}$ (Definition 2) if, for every PPT adversary $\mathcal{A}$, there exists a PPT ideal adversary S such that no PPT environment $\mathcal{Z}$ can distinguish between a real execution of Π and an ideal execution of $\mathcal{F}$ with a significant advantage. The execution of the two worlds, the real world (where Π is executed) and the ideal world (where $\mathcal{F}$ is executed), is shown in Figure 2.2.

Definition 2 (UC Realisation, [20])**.** *Let π and ϕ be two* PPT *protocols. π is said to UC-realise ϕ if for any* PPT *adversary, $\mathcal{A}$, there exists a* PPT *adversary,* S*, such that for any* PPT *$\mathcal{Z}$:*

$$\mathsf{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}} \approx \mathsf{EXEC}_{\mathcal{F},\mathsf{S},\mathcal{Z}}. \tag{2.3}$$

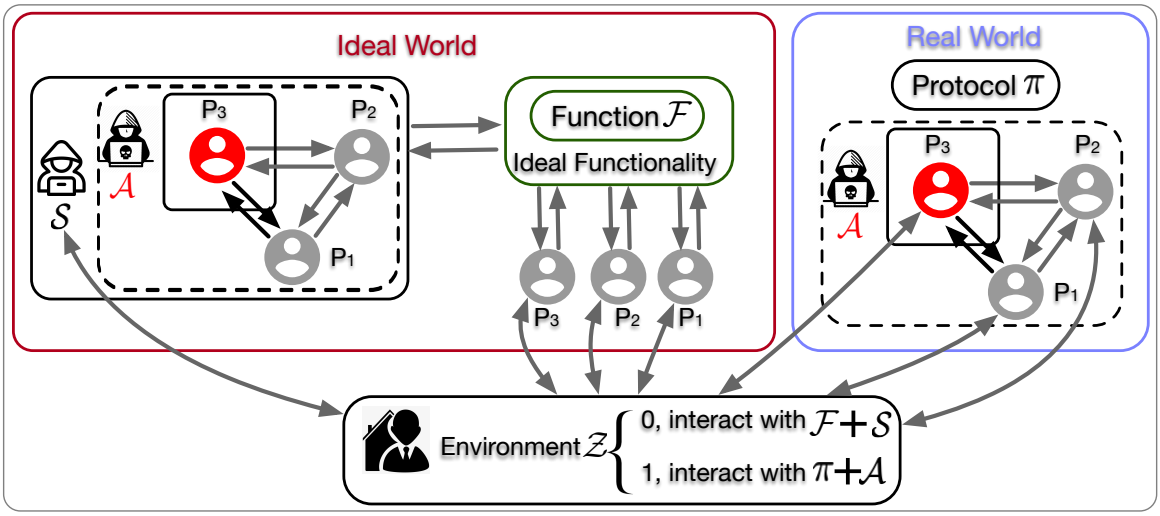

Figure 2.2: Ideal World and Real World in the UC Framework.

In the UC framework, security is established through simulation. The main idea is to construct a simulator, denoted as S, which utilises the adversary $\mathcal{A}$ as a black-box and provides an indistinguishable view to the environment $\mathcal{Z}$. Specifically, when dealing with honest participants, the simulator S simulates the messages to the environment $\mathcal{Z}$ by following the protocol execution as it would occur in the real world. However, when corrupted participants are involved, the simulator S needs to simulate the messages to $\mathcal{Z}$ on behalf of honest participants. In such cases, S may need to "rewind" or backtrack to extract relevant information, such as witness values, to correctly simulate the interactions with $\mathcal{Z}$. This is necessary when S does not know the real inputs or internal states of the corrupted

participants. The security proof can investigate resilience against static corruption in the Random Oracle (RO) model. This involves demonstrating the indistinguishability between the real/hybrid world executions (involving the actual protocol, participants, and adversaries) and the ideal world executions (involving the ideal functionality and ideal adversaries). By showing that the simulator S can effectively simulate the interactions with the environment $\mathcal{Z}$ and provide an indistinguishable view of the two worlds, the security of the protocols is established. This proof ensures that the protocols are resistant to attacks and adversaries, and it is based on the concept of indistinguishability, which is a fundamental property in cryptographic security.

Overall, the security analysis in the UC framework, with the use of simulation and indistinguishability, provides a rigorous and formal approach to prove the security guarantees of cryptographic protocols against static corruption in the presence of adversaries. It allows for the assessment of the protocol's resilience in realistic scenarios and its suitability for secure composability in complex cryptographic systems.

2.4 Public-Key Encryption

Public-Key Encryption (PKE, Definition 3) is a cryptographic scheme that enables secure communication between participants without requiring them to have agreed on any secret information beforehand. PKE schemes leverage mathematical problems such as the Discrete Logarithm problem (Definition 4) and the Decisional Diffie-Hellman assumption (DDH assumption, Definition 5) to achieve their security. These mathematical assumptions ensure the computational hardness of certain operations, making it difficult for adversaries to break the encryption and deduce the private key from the public key. However, it's important to acknowledge potential limitations, as there are scenarios where DDH may not hold[34], leading to vulnerabilities in protocols. In this work, the reliance on DDH highlights the need for ongoing evaluation and potential exploration of alternative cryptographic assumptions to ensure robustness against potential attacks.

Definition 3 (Public-Key Encryption, [21])**.** *A Public-Key Encryption scheme,* PKE*, is a triple of* PPT *algorithms,* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, such that*

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Gen}(1^\kappa)$. *The key-generation algorithm,* $\mathsf{PKE.Gen}$*, takes security parameter,* κ*, as an input, outputs public key,* pk*, and secret key,* sk*. It is assumed that* pk *and* sk *each should have at least length* κ*;*

- $c \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}}(m)$. *The encryption algorithm,* $\mathsf{PKE.Enc}$*, takes public key,* pk*, and a message,* m*, from a message space,* $\mathcal{M}$*, as inputs, and outputs cipher-text,* c*.* $\mathsf{PKE.Enc}$ *needs to be probabilistic to achieve meaningful security;*

- $m = \mathsf{PKE.Dec}_{\mathsf{sk}}(c)$. *The deterministic decryption algorithm,* PKE.Dec*, takes secrete key,* sk*, and cipher-text,* c*, as inputs, and outputs either message,* m*, or a special symbol,* $\perp$*, denoting failure.*

Definition 4 (Discrete Logarithm Problem, [21])**.** *Let* κ *be the security parameter,* $\mathbb{G}$ *be a cyclic group of order,* q*,* g *is a generator of* $\mathbb{G}$*,* $\mathcal{G}$ *be a group-generation algorithm,* $\mathcal{A}$ *be the adversary, the game,* $\mathsf{Exp}^{\mathsf{DLog}}_{\mathcal{G},\mathcal{A}}(1^\kappa)$*, is defined as follows:*

- $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^\kappa)$. *The challenger runs* $\mathsf{Gen}(1^\kappa)$ *to generate group elements,* $\mathbb{G}, q, g$*;*
- *The challenger selects* $h \in \mathbb{G}$*;*
- $x \leftarrow \mathcal{A}(\mathbb{G}, q, g, h)$. $\mathcal{A}$ *outputs* $x \in \mathbb{Z}_q$*;*
- *The output of* $\mathsf{Exp}^{\mathsf{DLog}}_{\mathcal{G},\mathcal{A}}(1^\kappa)$ *is 1 if and only if* $g^x = h$*.*

Discrete Logarithm problem is hard relative to $\mathcal{G}$ *if for all* PPT *adversary,* $\mathcal{A}$*, there exists a negligible function,* negl*, that*

$$\Pr[\mathsf{DLog}(\mathcal{A}, \mathcal{G}, 1^\kappa) = 1] \leq \mathsf{negl}(\kappa). \tag{2.4}$$

Definition 5 (Decisional Diffie-Hellman Assumption, [21])**.** *The Decisional Diffie-Hellman assumption is hard relative to* $\mathbb{G}$ *if for all* PPT *adversary,* $\mathcal{A}$*, there exists a negligible function,* negl*, such that*

$$\left| \Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{x \cdot y}) = 1] \right| \leq \mathsf{negl}(\kappa), \tag{2.5}$$

where in each case the probabilities are taken over the experiment in which $\mathbb{G}(1^n)$ *outputs* $(\mathbb{G}, q, g)$*, and then* $\{x, y, z\} \in \mathbb{Z}_q$ *are chosen uniformly. When* z *is uniform in* $\mathbb{Z}_q$*,* g^z *is uniformly distributed in* $\mathbb{G}$*.*

2.4.1 Homomorphic Public-Key Encryption

Lifted ElGamal Encryption (LEG) is a variant of ElGamal Encryption that serves as a candidate for additively homomorphic PKE. The protocols proposed in this context involve relatively small messages, making the resolution of the discrete logarithm problem in LEG feasible. The security of LEG is based on the concept of Indistinguishability of Keys under Chosen-Plaintext Attack (IK-CPA) as defined in Definition 7. With its additively homomorphic property, we can encrypt the messages sent in the protocols and enable batch verification in Zero-Knowledge Proofs.

By employing LEG in the cryptographic protocols, the system can ensure the confidentiality of messages while facilitating secure computations and batch verification. The combination of LEG's security properties and additively homomorphic property makes it a valuable tool in achieving privacy and efficiency in the proposed protocols.

Definition 6 (Lifted ElGamal Encryption, [35])**.** *A Lifted ElGamal Encryption scheme,* LEG*, is a set of four* PPT *algorithms,* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Add}, \mathsf{Dec})$*, such that*

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{LEG.Gen}(1^\kappa)$*. The key generation algorithm,* LEG.Gen*, takes* κ *as an input, picks secret key,* $\mathsf{sk} \leftarrow (\mathbb{Z}_q)^*$*, sets public key as* $\mathsf{pk} := g^{\mathsf{sk}}$*, and outputs key pair,* $(\mathsf{pk}, \mathsf{sk})$*;*
- $(c_1, c_2) \leftarrow \mathsf{LEG.Enc}_{\mathsf{pk}}(m; r)$*. The encryption algorithm,* LEG.Enc*, picks randomness,* $r \leftarrow (\mathbb{Z}_q)^*$*, and outputs cipher-text,* $(c_1, c_2) := (g^r, g^m \cdot \mathsf{pk}^r)$*;*
- $(\prod_{i=1}^{\ell} c_{i,1}, \prod_{i=1}^{\ell} c_{i,2}) \leftarrow \mathsf{LEG.Add}((c_{1,1}, c_{1,2}), \ldots, (c_{\ell,1}, c_{\ell,2}))$*.* LEG *is additively homomorphic;*
- $m \leftarrow \mathsf{LEG.Dec}_{\mathsf{sk}}(c_1, c_2)$*. The decryption algorithm,* LEG.Dec*, takes cipher-text,* (c_1, c_2)*, as an input, and outputs message,* $m = \mathsf{DLog}(c_2 \cdot c_1^{-\mathsf{sk}})$*.*

Definition 7 (IK-CPA Security, [36])**.** *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme,* κ *be the security parameter, and* $b \in \{0, 1\}$*,* $\mathcal{A}$ *be the adversary that runs in two stages. The game,* $\mathsf{Exp}^{\mathsf{IK\text{-}CPA}-b}_{\mathsf{PKE},\mathcal{A}}(1^\kappa)$*, is defined as follows:*

- $\{(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1)\} \leftarrow \mathsf{PKE.Gen}(1^\kappa)$*. The challenger runs* $\mathsf{Gen}(1^\kappa)$ *to generate two pairs of keys,* $(\mathsf{pk}_0, \mathsf{sk}_0)$ *and* $(\mathsf{pk}_1, \mathsf{sk}_1)$*, then sends* $(\mathsf{pk}_0, \mathsf{pk}_1)$ *to* $\mathcal{A}$*;*
- $(x, s) \leftarrow \mathcal{A}(\textsc{Find}, \mathsf{pk}_0, \mathsf{pk}_1)$*. In the* FIND *stage,* $\mathcal{A}$ *takes two public keys as inputs, outputs a message,* x*, and a stage information,* s*;*
- $y \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_b}(x)$*. The challenger randomly picks* pk_b *from* $\{\mathsf{pk}_0, \mathsf{pk}_1\}$*, then computes* $y \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_b}(x)$ *and sends* y *to* $\mathcal{A}$*;*
- $d \leftarrow \mathcal{A}(\textsc{Guess}, y, s)$*. In the* GUESS *stage,* $\mathcal{A}$ *decides which key was used to encrypt* x *based on the cipher-text,* y*, and stage information,* s*, then outputs its decision,* $d \in \{0, 1\}$*;*
- *Return* d*.* $\mathcal{A}$ *returns* d *as the output of* $\mathsf{Exp}^{\mathsf{IK\text{-}CPA}-b}_{\mathsf{PKE},\mathcal{A}}(1^\kappa)$*.*

The advantage of an adversary who wins the aforementioned game is defined as:

$$\mathsf{Adv}^{\mathsf{IK\text{-}CPA}}_{\mathsf{PKE}}(1^\kappa, \mathcal{A}) = \Pr[\mathsf{Exp}^{\mathsf{IK\text{-}CPA}-1}_{\mathsf{PKE},\mathcal{A}}(1^\kappa) = 1] - \Pr[\mathsf{Exp}^{\mathsf{IK\text{-}CPA}-0}_{\mathsf{PKE},\mathcal{A}}(1^\kappa) = 1], \tag{2.6}$$

$$\mathsf{Adv}^{\mathsf{IK\text{-}CPA}}_{\mathsf{PKE}}(1^\kappa, \mathcal{A}) \leq \mathsf{negl}(\kappa). \tag{2.7}$$

Theorem 2 (IK-CPA Security of LEG, [36])**.** *Let* $\mathcal{G}$ *be a prime-order-group generator. If the DDH assumption is a computational hardness assumption for* $\mathcal{G}$*, then the (Lifted) ElGamal scheme,* LEG*, is* IK-CPA *secure. Besides, for any adversary,* $\mathcal{A}$*, there exists a distinguisher,* $\mathcal{D}$*, such that for any* κ*:*

$$\mathsf{Adv}^{\mathsf{IK\text{-}CPA}}_{\mathsf{LEG},\mathcal{A}} \leq 2 \cdot \mathsf{Adv}^{DDH}_{\mathcal{G},\mathcal{D}}(\kappa) + \frac{1}{2^{\kappa-2}}. \tag{2.8}$$

Definition 8 (IND-CPA Security)**.** *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public encryption scheme,* $b \in \{0,1\}$*,* κ *be the security parameter,* $\mathcal{A}$ *be the adversary that runs in two stages. The game,* $\mathsf{Exp}^{\mathsf{IND\text{-}CPA}-b}_{\mathsf{PKE},\mathcal{A}}(1^\kappa)$*, is defined as the following:*

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Gen}(1^\kappa)$*. The challenger runs* $\mathsf{Gen}(1^\kappa)$ *to generate a pair of keys,* $(\mathsf{pk}, \mathsf{sk})$*, and sends* pk *to* $\mathcal{A}$*;*
- $(m_0, m_1) \leftarrow \mathcal{A}(\textsc{Find})$*. In the* FIND *stage,* $\mathcal{A}$ *outputs two messages,* m_0 *and* m_1*;*
- $c \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}}(m_b)$*. The challenger randomly picks* $m_b \leftarrow \{m_0, m_1\}$*, encrypts* m_b *with* pk *and sends the cipher-text,* c*, to* $\mathcal{A}$*;*
- $d \leftarrow \mathcal{A}(\textsc{Guess}, c)$*. In the* GUESS *stage,* $\mathcal{A}$ *decides which message was encrypted in* c*, and outputs its decision,* $d \in \{0,1\}$*;*
- *Return* d*,* $\mathcal{A}$ *returns* d *to the challenger.*

The advantage of an adversary who wins the aforementioned game is defined as:

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}} = Pr[\mathsf{Exp}^{\mathsf{IND\text{-}CPA}-1}_{\mathsf{PKE},\mathcal{A}}(\kappa) = 1] - Pr[\mathsf{Exp}^{\mathsf{IND\text{-}CPA}-0}_{\mathsf{PKE},\mathcal{A}}(\kappa) = 1]. \tag{2.9}$$

The public-key encryption, PKE*, is* IND-CPA *secure if* $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}}(\cdot)$ *is negligible for any adversary,* $\mathcal{A}$*, with polynomial time complexity of* κ*,*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}} \leq \mathsf{negl}(\kappa). \tag{2.10}$$

Theorem 3 (IND-CPA Security of LEG)**.** *Let* $\mathcal{G} \in \mathbb{G}$ *be a prime-order-group generator. If the DDH problem is computationally hard for* $\mathcal{G}$*, then the (Lifted) ElGamal scheme,* LEG*, is* IND-CPA *secure. Besides, for any adversary,* $\mathcal{A}$*, the following advantage of adversary,* $\mathcal{A}$*, is* $\mathsf{negl}(\kappa)$*:*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{LEG}}(1^\kappa, \mathcal{A}) := \left| Pr[\mathbb{G}^{\mathsf{LEG},\mathcal{A}}_{0,DDH} = 1] - Pr[\mathbb{G}^{\mathsf{LEG},\mathcal{A}}_{1,DDH} = 1] \right|. \tag{2.11}$$

The main distinction between the Lifted ElGamal Encryption scheme (LEG) and the traditional ElGamal encryption scheme is the way in which they encrypt the messages. In LEG, instead of directly encrypting the plaintext m, it encrypts g^m, where g is a generator of the group used in the encryption. As a result, decryption of ciphertexts (c_1, c_2) under LEG requires computing the discrete logarithm to retrieve the original plaintext:

$$m = \mathsf{DLog}(g^m). \tag{2.12}$$

However, computing the discrete logarithm is a computationally challenging problem, which adds a level of security to the encryption scheme. If the messages in the protocols are relatively small, a bidirectional mapping can be used to transform the original messages into

a smaller message space. This mapping allows for more efficient decryption of ciphertexts without compromising the security of the encryption scheme. By applying this mapping, the decryption process becomes more manageable while maintaining the confidentiality and integrity of the encrypted messages.

Let p be a perfect power of 2 (*e.g.*, 2^3), $\mathbf{d}^{[\ell]} := \{d_1, \ldots, d_\ell\}$, and $\mathbf{p}^{[\ell]} := \{p, \ldots, p^\ell\}$. Denote the encoder function and decode function by

$$\begin{aligned} \mathbf{d}^{[\ell]} &= \mathsf{Encode}(m), \\ m &= \mathsf{Decode}(\mathbf{d}^{[\ell]}) \end{aligned} \tag{2.13}$$

where $\mathbf{d}^{[\ell]} \cdot \mathbf{p}^{[\ell]} = m$. For clarity, $(\mathbf{c_1}^{[\ell]}, \mathbf{c_2}^{[\ell]}) = \mathsf{LEG.Enc}_{\mathsf{pk}}(\mathbf{d}^{[\ell]}; \mathbf{r}^{[\ell]})$, $\mathbf{d}^{[\ell]} = \mathsf{LEG.Dec}_{\mathsf{sk}}(\mathbf{c_1}^{[\ell]}, \mathbf{c_2}^{[\ell]})$ are used to represent ℓ encryption and decryption operations of ℓ elements in $\mathbf{d}^{[\ell]}$.

2.5 Commitment

In the distributed key generation protocol and related zero-knowledge proofs, certain participants need to hide secret information initially and later reveal it securely. It is crucial to ensure that this secret information remains unchanged during the revealing process and that, prior to revealing, only the secret owner knows this information. To meet these requirements, a commitment scheme can be used.

A commitment scheme, as introduced in [22], can be thought of as a digital envelope. In this scheme, a participant (denoted as P_1) "commits" a message m to a commitment string c and then passes this commitment to another participant (denoted as P_2). Informally, in a commitment scheme, P_1 seals a message m in an envelope (the commitment string c) and hands over the envelope to P_2. Afterwards, P_1 sends both the commitment string c and an opening string s to P_2, thereby convincing P_2 that the message m is indeed committed in c.

A commitment scheme has two essential properties, Hiding (Definition 9) and Binding (Definition 10):

- *Hiding*: This property ensures that given the commitment string c, an adversary cannot determine the committed message m. In other words, the commitment conceals the underlying message.

- *Binding*: This property ensures that for any committed message m, it is computationally infeasible for an adversary to find two distinct opening strings s_1 and s_2 such that both strings open to the same commitment c. In other words, the commitment prevents any manipulation or changing of the committed value without being detected during the revealing process.

By satisfying both the Hiding and Binding properties, a commitment scheme provides a secure and reliable mechanism for participants to commit to a message and later reveal it while ensuring confidentiality and integrity.

Definition 9 (Commitment Hiding Game, [21])**.** *Let* $\mathsf{C} = (\mathsf{Gen}, \mathsf{Com})$ *be a commitment scheme,* $m \in \mathcal{M}$ *be a message,* κ *be the security parameter,* $\mathcal{A}$ *be the adversary. The commitment hiding game,* $\mathsf{Exp}^{\mathsf{Hiding}-b}_{\mathsf{C},\mathcal{A}}(1^\kappa)$*, is defined as the following:*

- $\mathsf{ck} \leftarrow \mathsf{C.Gen}(1^\kappa)$*. The challenger runs* $\mathsf{C.Gen}(1^\kappa)$ *to generate a commitment key,* ck*, and sends* ck *to* $\mathcal{A}$*;*
- $(m_0, m_1) \leftarrow \mathcal{A}(\textsc{Find})$*. In the* FIND *stage,* $\mathcal{A}$ *outputs two messages,* m_0 *and* m_1*;*
- $c \leftarrow \mathsf{C.Com}_{\mathsf{ck}}(m_b; r)$*. The challenger randomly picks* $r \leftarrow (\mathbb{Z}_q)^*$*,* $m_b \leftarrow \{m_0, m_1\}$*, then commits* m_b *with* ck *and* r*, and sends the commitment,* c*, to* $\mathcal{A}$*;*
- $d \leftarrow \mathcal{A}(\textsc{Guess}, c)$*. In the* GUESS *stage,* $\mathcal{A}$ *decides which message was committed in* c*, and outputs its decision,* $d \in \{0, 1\}$*;*
- *The output of* $\mathsf{Exp}^{\mathsf{Hiding}-b}_{\mathsf{C},\mathcal{A}}(1^\kappa)$ *is 1 if and only if* $d = b$*.*

Definition 10 (Commitment Binding Game, [21])**.** *Let* $\mathsf{C} = (\mathsf{Gen}, \mathsf{Com})$ *be a commitment scheme,* $m \in \mathcal{M}$*,* κ *be the security parameter,* $\mathcal{A}$ *be the adversary. The commitment binding game,* $\mathsf{Exp}^{\mathsf{Binding}}_{\mathsf{C},\mathcal{A}}(1^\kappa)$*, is defined as the following:*

- $\mathsf{ck} \leftarrow \mathsf{C.Gen}(1^\kappa)$*: The challenger runs* $\mathsf{Gen}(1^\kappa)$ *to generate a commitment key,* ck*, and sends* ck *to* $\mathcal{A}$*;*
- $(c, m_0, r_0, m_1, r_1) \leftarrow \mathcal{A}(\textsc{Find})$*: In the* FIND *stage,* $\mathcal{A}$ *outputs two messages,* m_0 *and* m_1*, two randomnesses,* r_0 *and* r_1*, and a commitment,* c*;*
- *The output of* $\mathsf{Exp}^{\mathsf{Binding}}_{\mathsf{C},\mathcal{A}}(1^\kappa)$ *is 1 if and only if* $m_0 \neq m_1$ *and* $\mathsf{C.Com}_{\mathsf{ck}}(m_0; r_0) = \mathsf{C.Com}_{\mathsf{ck}}(m_1; r_1) = c$*.*

Theorem 4 (Secure Commitment, [21])**.** *A commitment scheme,* C*, is secure if for all* PPT *adversary,* $\mathcal{A}$*, there is a negligible function,* negl*, such that*

$$\begin{aligned} &\mathsf{Adv}^{\mathsf{Hiding}}_{\mathsf{C}}(1^\kappa, \mathcal{A}) = Pr[\mathsf{Exp}^{\mathsf{Hiding}-b}_{\mathsf{C},\mathcal{A}}(1^\kappa) = 1] \leq 2^{-1} + \mathsf{negl}(\kappa), \\ &\mathsf{Adv}^{\mathsf{Binding}}_{\mathsf{C}}(1^\kappa, \mathcal{A}) = Pr[\mathsf{Exp}^{\mathsf{Binding}-b}_{\mathsf{C},\mathcal{A}}(1^\kappa) = 1] \leq \mathsf{negl}(\kappa). \end{aligned} \tag{2.14}$$

The Pedersen Commitment scheme (Definition 11, [37]) is a specific type of commitment scheme used in the context of Distributed Key Generation and Zero-Knowledge Proofs, following the work in [38]. The Pedersen Commitment scheme is based on the Discrete Logarithm problem and possesses two crucial properties: unconditional perfect hiding and computational binding.

Definition 11 (Pedersen Commitment)**.** *A Pedersen Commitment,* PC*, is a set of four* PPT *algorithms,* $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Open}, \textsc{Verify})$*, such that*

- $\mathsf{PC.Gen}(1^\kappa)$*. The commitment key generation algorithm,* PC.Gen*, takes* 1^κ *as the input, and outputs a commitment key,* ck*;*
- $\mathsf{PC.Com}_{\mathsf{ck}}(m; r)$*. The commitment algorithm,* PC.Com*, takes a message* $m \in \mathcal{M}$*, and a randomness,* $r \leftarrow (\mathbb{Z}_q)^*$ *as inputs, and outputs a commitment,* $c := g^m \cdot \mathsf{ck}^r$*;*
- $\mathsf{PC.Open}(c)$*. The open algorithm,* PC.Open*, takes the commitment,* c*, as an input, and outputs* $d := (m, r)$*;*
- $\mathsf{PC.}\textsc{Verify}(c, d)$*. The verify algorithm,* PC.VERIFY*, takes the commitment,* c*, and opening string,* d*, as inputs, and returns 1 if and only if* $c = g^m \cdot \mathsf{ck}^r$*;*
- $\mathsf{PC.Com}_{\mathsf{ck}}(m_1; r_1) \cdot \mathsf{PC.Com}_{\mathsf{ck}}(m_2; r_2) = \mathsf{PC.Com}_{\mathsf{ck}}(m_1 + m_2; r_1 + r_2)$*, Pedersen commitment is additively homomorphic.*

Let κ *be the security parameter. A scheme,* PC*, is a Pedersen Commitment scheme if the following properties hold:*

- *Perfect Hiding:* PC *is perfect hiding if for any adversary,* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{l}\mathsf{ck} \leftarrow \mathsf{PC.Gen}(1^\kappa); (\{m_0, m_1\} \leftarrow \mathcal{A}(\mathsf{ck}); \\ b \in \{0,1\}; B \leftarrow \mathsf{PC.Com}_{\mathsf{ck}}(m_b; r); \mathcal{A}(B) = b\end{array}\right] = 2^{-1}. \tag{2.15}$$

- *Computational Binding:* PC *is computational binding if for any adversary,* $\mathcal{A}$*, the advantage of adversary,* $\mathsf{Adv}^{\mathsf{Binding}}_{\mathsf{PC}}(1^\kappa, \mathcal{A})$*, is*

$$\Pr\left[\begin{array}{l}\mathsf{ck} \leftarrow \mathsf{PC.Gen}(1^\kappa); \\ (m_0, r_0, m_1, r_1) \leftarrow \mathcal{A}(\mathsf{ck}) : m_0 \neq m_1 \\ \wedge \mathsf{PC.Com}_{\mathsf{ck}}(m_0; r_0) = \mathsf{PC.Com}_{\mathsf{ck}}(m_1; r_1)\end{array}\right] = \mathsf{negl}(\kappa). \tag{2.16}$$

2.6 Secret Sharing

Secret sharing is a fundamental concept in threshold cryptography, and it was independently proposed in [23] and [39]. Secret sharing allows a secret to be distributed among multiple participants in a way that any qualified subset of participants can reconstruct the original secret, but no subset of participants with fewer members can learn any information about the secret. This property is crucial in ensuring the security and robustness of distributed systems, as it prevents any single point of failure and ensures that no single participant holds the entire secret.

Secret sharing is used in distributed key generation, where cryptographic protocols involve multiple participants collaborating to generate cryptographic keys securely. There are three different approaches to share secrets:

- Shamir's Secret Sharing [23]: Shamir's secret sharing is used to share one secret among n participants, with a computation and communication cost of $\mathcal{O}(n^2)$. This method is based on polynomial interpolation and is widely used in various cryptographic applications.
- Verifiable Secret Sharing [24]: Verifiable secret sharing is used to ensure the honesty of dealers in the secret sharing process. It enables participants to verify that the secret shares they receive are indeed valid and consistent with the commitments made by the dealers.
- Hyper-Invertible Matrix-based Secret Sharing [25]: This approach is used to share a secret among n participants with an amortised computation and communication cost of $\mathcal{O}(n)$. It leverages the properties of hyper-invertible matrices to achieve efficient secret sharing with reduced computational overhead.

By employing these different secret sharing techniques, the distributed key generation protocols can securely generate cryptographic keys among multiple participants, ensuring that no single entity holds the entire secret and enhancing the overall security and resilience of the cryptographic systems.

Hyper-Invertible Matrix (HIM, Definition 12, Construction 1, [25]) is a matrix that any square submatrix formed by removing rows and columns of this matrix is invertible. The symmetry property of HIM enables that any subset of n inputs/outputs can be used to linearly derive the remaining inputs/outputs, which is formalised in Theorem 5.

Definition 12 (Hyper-Invertible Matrix, [25])**.** *An $m \times n$ matrix, $\mathbf{M}^{[m \times n]}$, is a hyper-invertible matrix if for any index sets, $\mathbf{I} \subseteq \{1, \ldots, m\}$ and $\mathbf{O} \subseteq \{1, \ldots, n\}$, with $|\mathbf{I}| = |\mathbf{O}| > 0$, the matrix, $\mathbf{M}^{\mathbf{I}}_{\mathbf{O}}$, is invertible of which the rows, $i \in \mathbf{I}$, and columns, $j \in \mathbf{O}$.*

Construction 1. *Given fixed elements, $\{\alpha_i\}_{i=1}^n$ and $\{\beta_i\}_{i=1}^n$, from $\mathbb{Z}_q$, let $F(z)$ be a polynomial which maps $\{x_i\}_{i=1}^n$ to $\{y_i\}_{i=1}^n$, where $F(\alpha_i) := x_i$ for $i \in [n]$, $y_i = F(\beta_i)$ for $i \in [n]$:*

$$y_i = F(\beta_i) = \sum_{j=1}^{n} \prod_{k=1, k \neq j}^{n} \frac{\beta_i - \alpha_k}{\alpha_j - \alpha_k} \cdot x_j = \sum_{j=1}^{n} \lambda_{i,j} \cdot x_j, \tag{2.17}$$

where $\{\lambda_{i,j}\}_{i=1,j=1}^{n,n}$ is a hyper-invertible matrix, $\mathbf{HIM}^{[n \times n]}$.

Theorem 5 (Hyper-Invertible Matrix Mapping, [25])**.** *Let $\mathbf{M}^{[n \times n]}$ be a $n \times n$ hyper-invertible matrix, given $(y_1, \ldots, y_n) = \mathbf{M} \cdot (x_1, \ldots, x_n)$, for any index sets $\mathbf{A}, \mathbf{B} \subseteq \{1, \ldots, n\}$ where $|\mathbf{A}| + |\mathbf{B}| = n$, there exists an invertible liner function which can map $(\{x_i\}_{i \in \mathbf{A}}, \{y_i\}_{i \in \mathbf{B}})$ to the values $(\{x_i\}_{i \neq \mathbf{A}}, \{y_i\}_{i \neq \mathbf{B}})$.*

To help understand HIM, the following example is given to show how to construct a 3*3 Hyper-Invertible Matrix and explain the properties of a Hyper-Invertible Matrix (HIM):

- **Step 1**: Fix 2 groups containing 3 elements/variables:

  Group 1: $a_1 = 1, a_2 = 2, a_3 = 3$;

  Group 2: $b_1 = 4, b_2 = 5, b_3 = 6$;

- **Step 2**: Define a polynomial such that $F(a_i) = x_i$. Defining this polynomial as $F(z) = z^2 + 1$, therefore $x_1 = (a_1)^2 + 1 = 2, x_2 = 5, x_3 = 10$;

- **Step 3**: Compute HIM parameters. Following the equation 2.17, HIM parameters are defined as $\lambda_{i,j} = \prod_{k=1,k\neq j}^{n} \frac{\beta_i - \alpha_k}{\alpha_j - \alpha_k}$. We can compute that

$$\begin{aligned}
\lambda_{11} &= ((b_1 - a_2)/(a_1 - a_2)) * ((b_1 - a_3)/(a_1 - a_3)) = 1;\\
\lambda_{11} &= ((b_1 - a_2)/(a_1 - a_2)) * ((b_1 - a_3)/(a_1 - a_3)) = 1;\\
\lambda_{12} &= ((b_1 - a_1)/(a_2 - a_1)) * ((b_1 - a_3)/(a_2 - a_3)) = -3;\\
\lambda_{13} &= ((b_1 - a_2)/(a_3 - a_2)) * ((b_1 - a_1)/(a_3 - a_1)) = 3;\\
\lambda_{21} &= ((b_2 - a_2)/(a_1 - a_2)) * ((b_2 - a_3)/(a_1 - a_3)) = 3;\\
\lambda_{22} &= ((b_2 - a_1)/(a_2 - a_1)) * ((b_2 - a_3)/(a_2 - a_3)) = -8;\\
\lambda_{23} &= ((b_2 - a_1)/(a_3 - a_1)) * ((b_2 - a_2)/(a_3 - a_2)) = 6;\\
\lambda_{31} &= ((b_3 - a_2)/(a_1 - a_2)) * ((b_3 - a_3)/(a_1 - a_3)) = 6;\\
\lambda_{32} &= ((b_3 - a_1)/(a_2 - a_1)) * ((b_3 - a_3)/(a_2 - a_3)) = -15;\\
\lambda_{33} &= ((b_3 - a_1)/(a_3 - a_1)) * ((b_3 - a_2)/(a_3 - a_2)) = 10;
\end{aligned}$$

  Therefore, the 3*3 Hyper-Invertible Matrix is

$$\begin{bmatrix} 1 & -3 & 3 \\ 3 & -8 & 6 \\ 6 & -15 & 10 \end{bmatrix}$$

When evaluating the predefined polynomial on (b_1, b_2, b_3) and get (y_1, y_2, y_3), it results that

$$y_i = \sum_{j=1}^{3} \lambda_{i,j} * x_j$$

for $i, j \in [3]$. That is, HIM parameters map (x_1, x_2, x_3) to (y_1, y_2, y_3). We can form 4 square matrices from

$$\begin{bmatrix} 1 & -3 & 3 \\ 3 & -8 & 6 \\ 6 & -15 & 10 \end{bmatrix}$$

and compute the determinant value denoted by det:

$$m_1 = \begin{bmatrix} 1 & -3 \\ 3 & -8 \end{bmatrix}, \mathsf{det}(m_1) = 1;$$

$$m_2 = \begin{bmatrix} -3 & 3 \\ -8 & 6 \end{bmatrix}, \mathsf{det}(m_2) = 6;$$

$$m_3 = \begin{bmatrix} 3 & -8 \\ 6 & -15 \end{bmatrix}, \mathsf{det}(m_3) = 3;$$

$$m_4 = \begin{bmatrix} -8 & 6 \\ -15 & 10 \end{bmatrix}, \mathsf{det}(m_4) = 10;$$

It is easy to infer that that all these sub-matrices are invertible. As defined in Definition 12, this matrix is a Hyper-Invertible Matrix because every square matrix by removing rows and columns from this matrix is invertible.

HIM can be used to detectably generate uniform random sharing by [25]. For completeness, $\mathsf{DoubleShareRandom}(d, d')$ protocol is recapped from [25] in Figure 2.3 to generate T independent random sharings $r_1, \ldots, r_T$ based on HIM. Let t be the threshold, applying n sharings to a $n \times n$ hyper-invertible matrix results in n sharings with the following properties:

Property 1. *([25]) If any (up to t) of the inputs sharings are broken, then this can be seen in every subset of t output sharings.*

Property 2. *([25]) If any $n - t$ input sharings are uniformly random, then every subset of size $n - t$ of output sharings is uniformly random.*

DoubleShareRandom(d, d')

Preparation:

Assume that each participant, $P_i \in \mathsf{P}$, holds a predefined HIM, $\mathbf{M}^{[n\times n]}$, and they want to generate random values $\{r_1, \ldots, r_T\}$.

Secret Share:

$P_i \in \mathsf{P}$ does the following:

- Select $s_i \leftarrow \mathbb{Z}_q$;
- Share s_i to P with $[s_i]_{d,d'}$.

Apply HIM:

$P_i \in \mathsf{P}$ locally computes $([r_1]_{d,d'}, \ldots, [r_n]_{d,d'}) = \mathbf{M}^{[n\times n]} \cdot ([s_1]_{d,d'}, \ldots, [s_n]_{d,d'})$.

Check:

- For $i = T+1, \ldots, n$, $P_j \in \mathsf{P}$ sends its share of $[s_i]_{d,d'}$ to P_i;
- $P_i \in \mathsf{P}$ checks that all $d-$shares lie on a polynomial with degree d, all $d'-$shares lie on another polynomial with degree d', and the free term of these two polynomials are the same.

Output:

Output remaining T double-sharings $[r_1]_{d,d'}, \ldots, [r_T]_{d,d'}$.

Figure 2.3: DoubleShareRandom(d, d'), [25]

2.7 Zero-Knowledge Proof

Zero-Knowledge Proof (ZKP) is a powerful cryptographic concept introduced in [26]. In a ZKP, a prover can convince a verifier that a certain assertion is true without revealing any additional information beyond the validity of the assertion. The prover demonstrates knowledge of a solution to a problem, without actually revealing the solution itself. This property makes ZKPs highly valuable in scenarios where privacy and confidentiality are paramount.

Zero-Knowledge Proofs can be used to prove the validity of messages in threshold encryption and the validation of ballots in a voting system without disclosing any additional information except the truth of the statements. In these protocols, if the message consists

of group elements (for exmaple, elements in an elliptic curve group), the participants always check their group membership as a preliminary step. However, to avoid unnecessary repetition and for brevity, this step can be omitted when describing the protocols.

By utilising Zero-Knowledge Proofs, the participants can efficiently prove the correctness of their actions and the validity of their messages without revealing sensitive information or compromising the security of the overall system. This ensures that the protocols are both secure and privacy-preserving, making them well-suited for applications where confidentiality and trust are critical.

2.7.1 Sigma-Protocol

Sigma-Protocol (Σ-Protocol, [40]) is a fundamental building block in Zero-Knowledge Proof. It is a 3-round interactive protocol between a prover, $\mathcal{P}$, and a verifier, $\mathcal{V}$. Given a common input, x, $\mathcal{P}$ tries to convince $\mathcal{V}$ that it knows some value, w, related to x, while revealing nothing except this assertion.

In the first round of Σ-Protocol, $\mathcal{P}$ sends a commitment, c, of x to $\mathcal{V}$. In the second round, $\mathcal{V}$ picks a random challenge, e, and sends it to $\mathcal{P}$. In the third round, $\mathcal{P}$ computes a response, z, based on the challenge and sends it back to $\mathcal{V}$, $\mathcal{V}$ then can decide to either accept or reject the response. $V_\Sigma(x, c, e, z) = 1$ indicates that (c, e, z) is accepted for x, if and only if $\mathcal{V}$ accepts (c, e, z).

Let $\mathcal{R}$ be a polynomial time decidable binary relation, Σ-Protocol is formalised in Definition 13.

Definition 13 (Σ-Protocol)**.** *The protocol $(\mathcal{P}, \mathcal{V}, V_\Sigma)$ is a Σ-Protocol for a relation $\mathcal{R}$, if it is a 3-round protocol that satisfies:*

- *Completeness. If $\mathcal{P}$ and $\mathcal{V}$ follow the protocol on input, x, and $\mathcal{P}$ gets another private input, w, where $(x, w) \in \mathcal{R}$, then $\mathcal{V}$ always accepts (c, e, z). $V_\Sigma(x, c, e, z) = 1$;*
- *Special Soundness. Given two accepting transcripts, (c, e, z) and (c, e', z'), where $e \neq e'$, there exists a* PPT *knowledge extractor procedure, $\mathcal{E}$, that computes a witness, w, such that $(x, w) \in \mathcal{R}$;*
- *Special Honest Verifier Zero-Knowledge (SHVZ, Definition 15). Given x and a fixed challenge, e, there exists a* PPT *simulator, $\mathcal{S}$, that computes triples, (c, e, z), with the same distribution as a valid transcript.*

2.7.2 Special Honest Verifier Zero-Knowledge

Let $\mathcal{R}$ be a polynomial time decidable binary relation. w is said to be a witness for a statement, x, if $(x, w) \in \mathcal{R}$. Language $\mathcal{L} := \{x \mid \exists w : (x, w) \in \mathcal{R}_\mathcal{L}\}$ is defined as the set of all statements, x, that have a witness, w, for the relation, $\mathcal{R}$. Let a prover, $\mathcal{P}$, and a verifier,

$\mathcal{V}$, be two PPT interactive algorithms and $\tau \leftarrow \langle \mathcal{P}(x,w), \mathcal{V}(x) \rangle$ be the public transcript produced by $\mathcal{P}$ and $\mathcal{V}$. Upon performing the protocol, $\mathcal{V}$ accepts the proof if and only if $\Delta(x,\tau) = 1$, where Δ is a public predicate function.

Definition 14. *For a relation, $\mathcal{R}$, $(\mathcal{P},\mathcal{V})$ is a perfectly complete argument if for all non-uniform* PPT *interactive adversaries, $\mathcal{A}$, the following holds:*

- *Perfect Completeness:*

$$\Pr\left\{ \begin{array}{l} (x,w) \leftarrow \mathcal{A}; \tau \leftarrow \langle P(x,w), V(x) \rangle : \\ (x,w) \in \mathcal{R}_{\mathcal{L}} \vee \Delta(x,\tau) = 1 \end{array} \right\} = 1. \tag{2.18}$$

- *Computational Soundness: $(\mathcal{P},\mathcal{V})$ is computational sound if for any adversary, $\mathcal{A}$, the advantage of adversary,* $\mathsf{Adv}^{\mathsf{sound}}_{SHVZ}(1^{\kappa},\mathcal{A})$*, is*

$$\Pr\left\{ \begin{array}{l} x \leftarrow \mathcal{A}; \tau \leftarrow \langle \mathcal{A}, V(x) \rangle : \\ x \notin \mathcal{L} \wedge \Delta(x,\tau) = 1 \end{array} \right\} = \mathsf{negl}(\kappa). \tag{2.19}$$

Let $V(x;r)$ denote that the verifier, $\mathcal{V}$, is executed on input x with random coin, r. A proof/argument $(\mathcal{P},\mathcal{V})$ is called *public coin* if the verifier, $\mathcal{V}$, randomly and independently picks its challenges of the messages sent by the prover, $\mathcal{P}$.

Definition 15 (Special Honest Verifier Zero-Knowledge, [41, 42, 43])**.** *A public coin proof/argument $(\mathcal{P},\mathcal{V})$ is a perfect Special Honest Verifier Zero-Knowledge (SHVZK) for a relation $\mathcal{R}$, if there exists a* PPT *simulator, $\mathcal{S}$, such that*

$$\Pr\left\{ \begin{array}{l} (x,w,r) \leftarrow \mathcal{A}; \\ \tau \leftarrow \langle P(x,w), V(x;r) \rangle : \\ (x,w) \in \mathcal{R}_{\mathcal{L}} \wedge \mathcal{A}(\tau) = 1 \end{array} \right\} \approx \Pr\left\{ \begin{array}{l} (x,w,r) \leftarrow \mathcal{A}; \\ \tau \leftarrow \mathcal{S}(x;r) : \\ (x,w) \in \mathcal{R}_{\mathcal{L}} \wedge \mathcal{A}(\tau) = 1 \end{array} \right\}. \tag{2.20}$$

Given an adversary, $\mathcal{A}$, that produces an acceptable argument with probability, p, there exists a witness-extended extractor (Definition 16), $\mathcal{E}$, that produces a similar argument with probability, p and outputs a witness.

Definition 16 (Extractor)**.** *A public coin proof/argument $(\mathcal{P},\mathcal{V})$ has a witness extended extractor, $\mathcal{E}$, if for all* PPT *$\mathcal{P}^*$ there exists an expected polynomial time extractor $\mathcal{E} = \mathcal{E}^{\mathcal{P}^*}$ such that for all* PPT *adversary, $\mathcal{A}$:*

$$\Pr\left\{ \begin{array}{l} (x,\psi) \leftarrow \mathcal{A}; \\ \tau \leftarrow \langle \mathcal{P}(x,\psi), \mathcal{V}(x) \rangle : \\ \mathcal{A}(\tau) = 1 \end{array} \right\} \approx \Pr\left\{ \begin{array}{l} (x,\psi) \leftarrow \mathcal{A}; \\ (\tau,w) \leftarrow \mathcal{E}^{\langle \mathcal{P}^*(x,\psi), \mathcal{V}(x) \rangle}(x,\psi) : \\ \mathcal{A}(\tau) = 1 \\ \wedge (\Delta(x,\tau) = 0 \vee (x,w) \in \mathcal{R}) \end{array} \right\}. \tag{2.21}$$

2.7.3 Non-Interactive Zero-Knowledge

Σ-protocol can be transformed to Non-Interactive Zero-Knowledge (NIZK) proof system based on Fiat-Shamir transformation ([44]). Let $\mathsf{hash} : \{0,1\}^* \to \mathbb{Z}_q$ be a collision-resistant hash function, instead letting verifier, $\mathcal{V}$, generate a random challenge, this challenge can be generated by hash based on the statement and the commitment.

The following properties hold for a NIZK protocol from a Σ-protocol based on Fiat-Shamir transformation:

- If the Σ-protocol is sound, then the NIZK protocol is also sound;
- If the Σ-protocol is SHVZK, then the NIZK protocol does not reveal any information about $\mathcal{P}$'s witness.

2.7.4 Proof of Knowledge

Let $\mathbb{G}$ be a cyclic group of prime order, q, with generator, $g \in \mathbb{G}$. In a Proof of Knowledge protocol (also known as Schnorr's identification protocol, [45]), the prover, $\mathcal{P}$, has a secret key, $x \in \mathbb{Z}_q$, and a public key, $y := g^x$. $\mathcal{P}$ wants to prove its identity to a verifier, $\mathcal{V}$, and convinces $\mathcal{V}$ that it knows the values, x. The specific construction is given in Figure 2.4. Assume that Discrete Logarithm problem is hard, Proof of Knowledge is secure against eavesdropping attack.

Proof of Knowledge Protocol

CRS: $g \in \mathbb{G}$, $\mathbb{Z}_q$
Statement: y
Witness: x
Protocol:

- $\mathcal{P}$ does the following:
  - Select $a \leftarrow \mathbb{Z}_q$;
  - Compute $A := g^a$;
  - $\mathcal{V} \leftarrow \mathcal{P} : \langle A \rangle$.
- $\mathcal{P} \leftarrow \mathcal{V} : e \leftarrow \{0,1\}^\kappa$
  /* For NIZK, set $e \leftarrow \mathsf{hash}(\langle g, y, A \rangle)$ */;
- $\mathcal{P}$ does the following:
  - Compute $Z := a + e \cdot x$;
  - $\mathcal{V} \leftarrow \mathcal{P} : \langle Z \rangle$.

Verification:

- For NIZK, $\mathcal{V}$ sets $e \leftarrow \mathsf{hash}(\langle g, y, A \rangle)$;
- $\mathcal{V}$ checks if $y^e \cdot A = g^Z$ holds.

Figure 2.4: Proof of Knowledge, Schnorr's identification protocol.

2.7.5 Schwartz-Zippel lemma

The Schwartz-Zippel theorem (Theorem 6, [46]) is used to prove the soundness in the Zero-Knowledge Proof.

Theorem 6 (Schwartz-Zippel Theorem, [46])**.** *Let f be a non-zero multivariate polynomial of degree, d, over $\mathbb{Z}_p$, then the probability of $f(x_1, \ldots, x_n) = 0$ evaluated with random variables, $x_1, \ldots, x_n$, from $\mathbb{Z}_p$ is at most $p^{-1}d$.*

Based on the Schwartz-Zippel theorem, it is feasible to derive that for two multivariate polynomials, f_1, f_2, and random variables, $x_1, \ldots, x_n$ from $\mathbb{Z}_p$, $f_1(x_1, \ldots, x_n) - f_2(x_1, \ldots, x_n) = 0$, one can assume that $f_1 = f_2$. For the reason that if $f_1 \neq f_2$, the probability that the above equation holds is bounded by $\frac{1}{p}\max(d_1, d_2) = \mathsf{negl}(\kappa)$.

2.8 Blockchain

2.8.1 Blockchain Properties

Initially, Blockchain was designed to support cryptocurrency like bitcoin. Blockchain, as the ledger, is a trustless decentralised database which is maintained by miners. Transactions are stored in blocks on blockchain in chronological order, and blocks are built by miners by appending one block to another. Blockchain has the following properties:

- **Decentralisation**. Blockchain synchronises replicated databases and guarantees a consistent view to all participation nodes. Information on blockchain can be validated without introducing a centralised third party;
- **Immutability**. Once a transaction is agreed and recorded on chain, it is sealed in the ledger and can not be modified anymore;
- **Transparency**. The ledger stores all the transactions, every change on blockchain is viewable and all the history can be traced back.

2.8.2 Blockchain Model

Following [13], the abbreviate blockchain model is given in Figure 2.5. At a high level, a blockchain model has the following four attributes:

- *Coin.* One coin can be spent only once, of which all the value must be consumed. Each coin has the followings 4 attributes:
  - **Coin ID**: Implicit attribute. Every coin has a unique ID that can be used to identify this coin;
  - **Value**: The value of the coin;
  - **Cond**: The spending conditions of the coin;
  - **Payload**: Any auxiliary information.
- *Address.* Formally, an address is either a public key, pk, or hash of a public key, $\mathsf{hash}(\mathsf{pk})$. Recall that each coin on blockchain has a spending condition, this condition is actually a valid signature under the corresponding public key, pk, of the address. An address is simply referred to generic representation of spending condition. Sender can create a new coin of which the spending condition is the recipient's address, so that this coin can only be spent by the correct recipient.
- *Transaction.* A transaction is denoted by $\mathsf{Tx}(A; B; C)$, where A is the set of input coins, B is the set of output coins, and C is the $\mathsf{Payload}$ field. Note that the verification

data is not explicitly described for simplicity. One transaction may take one or more unspent coins as the input, which are denoted by $\{\mathsf{In}_i\}_{i=1}^{n}$, it outputs one or more new coins, denoted as $\{\mathsf{Out}_j\}_{j=1}^{m}$. Every transaction needs transaction fee, which is the difference between the value of all input coins and the value of all output coins:

$$\sum_{i=1}^{n} \mathsf{In}_i.\mathsf{Value} - \sum_{j=1}^{m} \mathsf{Out}_j.\mathsf{Value}. \tag{2.22}$$

It is assumed that all the participants possess enough coins to pay the transaction fees throughout the protocol execution. The *Verification data* field of a transaction stores all the essential verification data, which fulfils the spending conditions of all input coins, $\{\mathsf{In}_i\}_{i=1}^{n}$. Additionally, there is a Payload field in each transaction to store any auxiliary information.

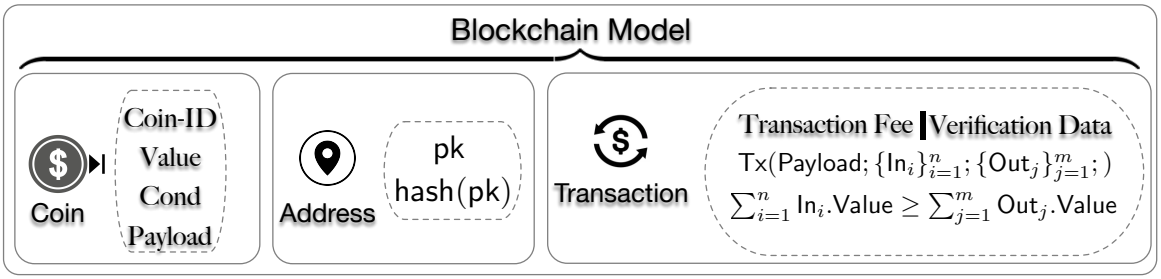

Figure 2.5: Blockchain Model

2.8.3 Blockchain Functionality

To model the communication process in the blockchain, this thesis takes the blockchain functionality from [47] which specifies data reading, writing and validity. For easy reference, the blockchain functionality, $\mathcal{F}_{\mathrm{BC}}$, is presented in Figure 2.6.

Blockchain Functionality$\mathcal{F}_{\text{BC}}$

Parameter: Successor relationship $succ : \{0,1\}^* \times \{0,1\}^* \leftarrow \{0,1\}$ The functionality, $\mathcal{F}_{\text{BC}}$, does the following:

- Upon receiving (INIT), set STORAGE $:= \emptyset$;
- Upon receiving (Read, ID), output STORAGE$[ID]$ or $\perp$ if not found;
- Upon receiving $(\mathsf{Write}, ID, input)$ from P, do the following:
  - Set $val :=$ STORAGE$[ID]$ or $\perp$ if not found;
  - If $succ(val, input) = 1$,then set STORAGE$[ID] := val||(input, P)$ and output $(\text{"}Receipt\text{"}, ID)$.
- Upon receiving $(\text{" } \in \text{"}, ID, val)$, output TRUE if $val \in$ STORAGE$[ID]$; output FALSE otherwise.

Figure 2.6: Blockchain Functionality, $\mathcal{F}_{\text{BC}}$, [47].

In this model, $succ$ limits the criteria of data added to blockchain storage. Data on blockchain is tied with a unique ID, which is used in both reading and writing interfaces. Additionally, $\mathcal{F}_{\text{BC}}$ also supports data ascertaining.

2.9 Distributed Key Generation

Unlike VSS, Distributed Key Generation (DKG, Definition 17, [48]) guarantees that even without trusted dealer no one can learn the secret, s, throughout the protocol execution. In essence, DKG can be considered as performing multiple VSS instances to generate keys without trust dealer setup. Distributed shares (referred to as partial secret keys) are added up to generate a global secret key. The distribution of generated global key pair (global public key and global secret key) is subject to uniformly random distribution.

Definition 17 (Distributed Key Generation, [48])**.** *An (t, n)-DKG protocol allows n participants to jointly generate a global secret key, s, and a global public key, g^s, such that k participants can reconstruct s, where $k \geq t$. A DKG scheme consists of the following two phases:*

- *Sharing: Each participant, P_i, distributes its secret value, z_i, to other participants. At the end of Sharing phase, each participant, P_j, holds a share, s_i, as a pre-decided linear combination of the distributed secret s. s_i is referred to as partial secret key;*
- *Extraction: Each participant broadcasts its partial public key. The global public key is extracted by multiplying the partial public keys owned by the honest participants.*

2.9.1 DKG in DLog setting

This thesis concentrates DKG for Discrete-Log Based Cryptosystem based on [38]. More specifically, given a cyclic group, $\mathbb{G}$ with order q, a DKG in DLog setting generates secret sharing of a global secret, s, and global public key g^s. For completeness, the DKG from [38] is presented in Figure 2.7.

Gennaro's DKG

Generating x

- Each participant, P_i, performs a Pedersen-VSS of a random values z_i as a dealer:
  - P_i chooses two random degree-t polynomials, $f_i(z)$, $f'_i(z)$, over $\mathbb{Z}_q$, $f_i(z) = a_{i,0} + a_{i,1} \cdot z + \ldots + a_{i,t} \cdot z^t$, $f'_i(z) = b_{i,0} + b_{i,1} \cdot z + \ldots + b_{i,t} \cdot z^t$, where $z_i = a_{i,0} = f_i(0)$;
  - P_i broadcasts $C_{i,k} = g^{a_{i,k}} \cdot h^{b_{i,k}}$ for $k \in [0, t]$. P_i computes the shares $s_{i,j} = f_i(j), s'_{i,j} = f'_i(j)$ for $j \in [n]$, and sends $s_{i,j}, s'_{i,j}$ to P_j;
  - P_j verifies the shares it received from other participants. For each $i \in [n]$, P_j checks if $g^{s_{i,j}} \cdot h^{s'_{i,j}} = \prod_{k=0}^{t} (C_{i,k})^{j^k}$. If this check fails for index i, P_j broadcasts a complaint against P_i;
  - P_i as a dealer received a complaint from P_j broadcasts the values, $s_{i,j}, s'_{i,j}$, that satisfy the check equation in last step.
- Any party that either received more than t complaints or answered a complaint without satisfying the check is marked as disqualified by the rest participants;
- Each participant builds a set of non-disqualified participants QUAL.

Extracting $y = g^x$

- Each participant, $P_i \in \mathsf{QUAL}$, exposes $y_i = g^{(z_i)}$ via Feldman-VSS:
  - $P_i \in \mathsf{QUAL}$ broadcasts $A_{i,k} = g^{a_{i,k}}$ for $k \in [0, t]$;
  - P_j verifies the values broadcast by the other participants in QUAL. For each $i \in \mathsf{QUAL}$, P_j checks if $g^{s_{i,j}} = \prod_{k=0}^{t} (A_{i,k})^{j^k}$. If the check fails for P_i, P_j complains against P_i by broadcasting $s_{i,j}, s'_{i,j}$;
  - For participants receive at least one complaint, the other participants reconstruct its $z_i, f_i(z), \{A_{i,k}\}_{k \in [0,t]}$;
  - For all participants in QUAL, set $y_i = A_{i,0} = g^{z_i}$, compute $y = \prod_{P_i \in \mathsf{QUAL}} y_i$.

Figure 2.7: Gennaro's DKG, [38].

2.9.2 Threshold Decryption

Figure 2.8 shows how to decrypt a cipher-text encrypted by the global public key using DKG in DLog setting and lifted Elgamal encryption on blockchain. For simplicity, the encoding details in lifted Elgamal encryption is skipped. Given cipher-texts $(c_1, c_2) := \mathsf{LEG.Enc}_{\mathsf{gpk}}(m; r)$, qualified participant, P_i, computes its decryption share, $R_i = (c_1)^{\mathsf{psk}_i}$, proves $\log_{g^r} R_i = \log_g \mathsf{ppk}_i$ with Logarithm Equality NIZK proof (Figure 2.9), π_i, and posts (R_i, π_i) to $\mathcal{F}_{\text{BC}}$.

Any t participants who showed correct Logarithm Equality NIZK proof in last step, can jointly decrypt the message by

$$\tau := c_2 \cdot (\prod_{P_k \in \mathsf{R}} R_k^{\mathcal{L}_k(0)})^{-1}, \tag{2.23}$$

where $\mathcal{L}_k(0) := \prod_{P_k \in \mathsf{R}, k \neq j} \frac{0-k}{j-k}$ are Lagrange Coefficients. The message can be computed by

$$m = \mathsf{DLog}_g(\tau). \tag{2.24}$$

Threshold Decryption

Setup: For $i \in \mathsf{QUAL}$, each participant, P_i, holds $\mathsf{psk}_i, \{\mathsf{ppk}_j\}_{j \in \mathsf{QUAL}}$.

Decryption:

- For $i \in \mathsf{QUAL}$, each participant, P_i, does the following:
  - Compute decryption share $R_i = (c_1)^{\mathsf{psk}_i}$;
  - Generate Logarithm Equality NIZK proof π_i, Cf. Figure 2.9:

    $$\pi_i \leftarrow \mathsf{NIZK}\left\{ \ (g, \mathsf{ppk}_i, R_i, c_1), (\mathsf{psk}_i, r) : \mathsf{ppk}_i = g^{\mathsf{psk}_i} \ \wedge R_i = (c_1)^{\mathsf{psk}_i} \ \wedge c_1 = g^r \ \right\}$$

  - Send $(\mathsf{Write}, (R_i, \pi_i))$ to $\mathcal{F}_{\text{BC}}$.
- Denote set R as the index set of any t participants who showed correct NIZK proof in last step, $\{P_k\}_{k \in \mathsf{R}}$ jointly decrypt the cipher-text as follows:
  - Compute $\tau := c_2 \cdot (\prod_{k \in \mathsf{R}} (R_k)^{\gamma_k})^{-1}$, where $\{\gamma_k\}_{k \in \mathsf{R}}$ are Lagrange Coefficients;
  - Output $m = \mathsf{DLog}_g(\tau)$.

Figure 2.8: Threshold Decryption.

Logarithm Equality Proof For integrity, Logarithm Equality Proof is given in Figure 2.9, of which the construction details can be found in Chaum and Pedersen's work [49].

Logarithm Equality ZK argument

CRS: $\{g, \mathsf{ppk}_i\} \in \mathbb{G} \setminus \{1\}$.
Statement: R_i, c_1
Witness: psk_i, r
Protocol:

- $\mathcal{P}$ does the following:
  - Select $a \leftarrow \mathbb{Z}_q$;
  - Compute $A := (c_1)^a$, $B := g^a$;
  - $\mathcal{V} \leftarrow \mathcal{P} : \langle A_B \rangle$.
- $\mathcal{V} \rightarrow \mathcal{P} : e \leftarrow \mathbb{Z}_q$;
  /* For NIZK, set $e \leftarrow \mathsf{hash}(\langle A_B \rangle)$ */
- $\mathcal{P}$ does the following:
  - Compute $z := a + e \cdot \mathsf{psk}_i$;
  - $\mathcal{V} \leftarrow \mathcal{P} : \langle z \rangle$.

Verification:

- $\mathcal{V}$ sets $e \leftarrow \mathsf{hash}(\langle A_B \rangle)$ for NIZK;
- $\mathcal{V}$ checks if the following hold:
  - $(c_1)^z = A \cdot (R_i)^e$;
  - $g^z = B \cdot (\mathsf{ppk}_i)^e$.

Figure 2.9: Logarithm Equality ZK argument for Threshold Decryption.

Theorem 7 (Logarithm Equality)**.** *Assume the DDH problem is hard. The protocol described in Figure 2.9 is an honest verifier zero-knowledge argument of knowledge of* psk_i *such that:*

- $\mathsf{ppk}_i = g^{\mathsf{psk}_i}$*;*
- $R_i = (c_1)^{\mathsf{psk}_i}$*;*
- $c_1 = g^r$.

Proof of Theorem 7.
Completeness.
$V_i = g^{\mathsf{psk}_i}$ is computed based on the property of polynomial function. Therefore, the followings can be easy to infer:

$$\begin{aligned} (c_1)^z &= (c_1)^{a+e\cdot\mathsf{psk}_i} = A \cdot (R_i)^e, \\ g^z &= g^{a+e\cdot\mathsf{psk}_i} = B \cdot (\mathsf{ppk}_i)^e. \end{aligned} \tag{2.25}$$

Soundness. The soundness is proved through an argument of knowledge (AoK), by showing that it has a witness-extended emulator. Assuming that there exists a PPT witness-extended extractor, $\mathcal{E}$, runs $\langle \mathcal{P}^*, \mathcal{V} \rangle$ to get transcripts. In addition, if $\mathcal{P}$ is able to make an acceptable argument, then $\mathcal{E}$ can also succeed with the same probability. $\mathcal{E}$ rewinds the protocol to the challenge phase (e) and runs it with fresh challenges. More specifically, $\mathcal{E}$ first gives a challenge e_1, then $\mathcal{E}$ can get:

$$z^{(1)} := a + e_1 \cdot \mathsf{psk}_i. \tag{2.26}$$

Afterwards, $\mathcal{E}$ rewinds the protocol to challenge phase feeding new challenge $e_{\alpha,2}$ and gets

$$z^{(2)} := a + e_2 \cdot \mathsf{psk}_i. \tag{2.27}$$

For $v \in \mathsf{T}$, by computing $(z^{(1)} - z^{(2)})/(e_1 - e_2)$, $\mathcal{E}$ gets witness, psk_i.

Zero Knowledge.

In terms of special honest verifier zero-knowledge, a simulator, S, is constructed that takes the challenge, $e \leftarrow \mathbb{Z}_q$, and statement, $\{V_i, R_i, c_1\}$, as inputs, it should output a simulated transcript the distribution of which is indistinguishable from the real one. In detail, for $v \in \mathsf{T}$, S first picks $z \leftarrow \mathbb{Z}_q$, and computes

$$\begin{aligned} A &= (R_i)^{-e} \cdot (c_1)^z, \\ B &= g^z \cdot (\mathsf{ppk}_i)^{-e}. \end{aligned} \tag{2.28}$$

After that, S outputs the simulated transcripts, $\langle \{A, B\} \rangle$. Since $\{A, B\}$ are uniquely determined for fixed elements from group G, they follow the same distribution in real argument. Therefore, simulated transcript has the same distribution as real transcript in a real argument. □

2.10 Cryptographic Sortition

Cryptographic Sortition is a technique used to randomly select participants or validators from a larger group in a secure and verifiable manner. It plays a crucial role in various decentralised systems, such as blockchain networks, where a subset of participants needs to be chosen to perform specific tasks, such as block validation or consensus. Inspired by [19, 18], a slightly different version of cryptographic sortition ideal functionality is given in Figure 2.10. This functionality interacts with a set of participants, $\mathsf{P} := \{P_1, \ldots, P_n\}$, and an adversary, S. Ideally, cryptographic sortition could be modelled as a "perfect" lottery functionality that selects the nodes based on their sortition weight. In the Preparation stage, $\mathcal{F}^n_{Sortition}$ generates a bit, b_i, for participant, P_i, where the probability that $b_i = 1$ is equal to a predefined sortition weight, w_i. Since adversary is in control of the public input,

communication, and the key pairs of corrupted participants, adversary can keep refresh the Sortition process until it is satisfied with the sortition results of corrupted participants.

Query REVEAL and query REFRESH are used to model adversary's power, where it can obtain the sortition results of the participant(s) it corrupted, and restart the whole sortition scheme until it is satisfied with the sortition results. Upon ending the Preparation stage, all participants can send REVEAL to obtain their sortition results. The sortition results are available to all participants. When someone wants to verify the sortition results of the others, it can simply send VERIFY to get the proof of the sortition result of any participant.

$\mathcal{F}^n_{Sortition}$

$\mathcal{F}^n_{Sortition}$ interacts with a set of participants, $\mathsf{P} := \{P_1, \ldots, P_n\}$, and an adversary, S. It is parameterised with $w \in (0, 1)$.

$\mathcal{F}^n_{Sortition}$ does the following:

1. **Preparation:**
   (a) Upon receiving $(\text{INIT}, \mathsf{sid})$ from $P_i \in \mathsf{P}$, choose a random independent bit, b_i, with $Pr[b_i = 1] := w$;
   (b) Upon receiving $(\text{REVEAL}, \mathsf{sid}, P_i)$ from S, send message $\langle b_i, P_i, \delta_i \rangle$ to S;
   (c) Upon receiving $(\text{REFRESH}, \mathsf{sid})$ from S, repeat Step 1a.
2. **Lottery:**
   (a) Upon receiving $(\text{REVEAL}, \mathsf{sid})$ from S, send $\langle b_i, \delta_i \rangle$ to P_i for $i \in [n]$.
3. **Verification:**
   (a) Upon receiving $(\text{VERIFY}, \mathsf{sid}, P_i)$ from $P_j \in \{\mathsf{P} \cup \{\mathsf{S}\}\}$, send π_i to P_j.

Figure 2.10: The cryptographic sortition functionality, $\mathcal{F}^n_{Sortition}$.

Verifiable Random Function (VRF, Definition 18,[50]) is used to implement cryptographic sortition. It is a pseudo-random function that provides publicly verifiable proofs for the correctness of the output. Intuitively, the pseudo randomness property states that the function value can not be distinguished from a random value, even after seeing any other function values together with the corresponding proofs.

Given any input string, x, $VRF_{\mathsf{sk}}(x)$ returns two values: a l-bit-long hash value that is uniquely determined by sk and x (indistinguishable from random value without knowledge of sk), and a proof, π, which enables public verification of this hash value based on pk.

Definition 18 (Verifiable Random Function, [51])**.** *A function family* $F_{(\cdot)}(\cdot) : \{0,1\}^l \to \{0,1\}^{l_{VRF}}$ *is a family of VRFs, if there exist algorithms,* $(\mathsf{Gen}, \mathsf{VRF}, \mathsf{VerifyVRF})$*, such that* Gen *outputs a pair of keys,* $(\mathsf{pk}, \mathsf{sk})$*,* $\mathsf{VRF}_{\mathsf{sk}}(x)$ *outputs a pair,* $(F_{\mathsf{sk}}(x), \pi_{\mathsf{sk}}(x))$*, where* $F_{\mathsf{sk}}(x)$

is the output value of the function and $\pi_{\mathsf{sk}}(x)$ *is the proof for verifying correctness, and* $\mathsf{VerifyVRF}_{\mathsf{pk}}(x, y, \pi)$ *verifies that* $y = F_{\mathsf{sk}}(x)$ *using the proof,* π*, returns 1 if* y *is valid, and 0 otherwise.*

Verifiable Random Function has the following properties:

- *Uniqueness: No values* $(\mathsf{pk}, x, y_1, y_2, \pi_1, \pi_2)$ *can satisfy* $\mathsf{VerifyVRF}_{\mathsf{pk}}(x, y_1, \pi_1) = \mathsf{VerifyVRF}_{\mathsf{pk}}(x, y_2, \pi_2)$ *unless* $y_1 = y_2$*;*
- *Provability: If* $(y, \pi) = \mathsf{VRF}_{\mathsf{sk}}(x)$*, then* $\mathsf{VerifyVRF}_{\mathsf{pk}}(x, y, \pi) = 1$*;*
- *Pseudo-randomness: Given the first input* 1^κ*, for any* PPT *algorithm,* $A = (A_E, A_J)$*, which runs for a total of* $s(\kappa)$ *steps without querying oracle on* x*:*

$$\Pr\left\{ b = b' \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa); \\ (x, st) \leftarrow A_E^{\mathsf{VRF}(\cdot)(\mathsf{pk})}; \\ y_0 = \mathsf{VRF}_{\mathsf{sk}}(x); y_1 \leftarrow \{0,1\}^{\ell_{\mathsf{VRF}}}; \\ b \leftarrow \{0,1\}; b' \leftarrow A_J^{\mathsf{VRF}(\cdot)(y_b, st)}; \end{array} \right\} \leqslant \frac{1}{2} + \mathsf{negl}(\kappa). \tag{2.29}$$

2.11 Summary

In this section, we have covered various cryptographic building blocks that are central to the protocols proposed in the privacy-preserving decision-making system. These building blocks play a crucial role in ensuring the security, privacy, and integrity of the system. The key points discussed in this section are summarised as follows:

- UC Framework: The UC framework was introduced as a powerful tool for proving the security of cryptographic protocols. It allows for the modular design and analysis of protocols, ensuring that their security properties are preserved even when composed with other protocols.
- Public-Key Encryption: PKE enables secure communication between parties without the need to agree on a shared secret beforehand. It was highlighted that the secrecy of the private key is essential in PKE.
- Commitment: Commitment schemes were explained as a way to hide secret information while later revealing it securely. This property is important in distributed key generation and other protocols where information needs to be kept secret until the appropriate time.
- Secret Sharing: Secret sharing was presented as a technique to distribute a secret among multiple participants in a secure manner. Different methods of secret sharing, including Shamir's, verifiable secret sharing, and hyper-invertible matrix-based secret sharing, were mentioned.

- Zero-Knowledge Proof: Zero-Knowledge Proof (ZKP) allows a prover to convince a verifier of the truth of a statement without revealing any additional information. ZKPs are crucial in validating messages and ensuring privacy in protocols.
- Blockchain: The properties of blockchain, such as decentralisation, immutability, and transparency, were outlined. Blockchain serves as a secure and distributed ledger, providing a reliable record of transactions in decentralised systems.
- Distributed Key Generation: DKG protocols enable a group of participants to jointly generate a private/public key pair without relying on a central authority. DKG is crucial in ensuring the security and trustworthiness of the decision-making system.
- Cryptographic Sortition: Cryptographic Sortition was introduced as a method for randomly selecting participants in a decentralised and verifiable manner. It is often used in consensus mechanisms to select block validators or leaders.

In succinct synthesis, this section serves as the bedrock of knowledge necessary for comprehending the protocols introduced in this thesis. It functions as the scaffolding for the ensuing chapters, wherein the aforementioned cryptographic building blocks are deftly harnessed to engineer secure and efficacious privacy-preserving decision-making protocols. In particular, the following chapter will delve into an exploration of related work, examining the landscape of existing designs and laying the groundwork for the innovative solutions that this thesis proposes.

Chapter 3

Related Work

> The past is behind, learn from it. The future is ahead, prepare for it. The present is here, live it.
>
> Thomas S. Monson

3.1 Overview

Research has consistently shown that group decision-making can lead to better collaborative intelligence and more efficient outcomes compared to individual decision-making. Decisions made by diverse groups tend to be more comprehensive and beneficial for the overall organisation [52, 53, 54, 55, 56, 57]. Group decision-making systems can generally be categorised into centralised and decentralised ones. In centralised systems, decisions are made by a central authority, while in decentralised systems, decision-makers are distributed across the organisation.

Decentralised decision-making ensures that various stakeholders participate actively in the decision-making process, allowing them to provide input and contribute to the final decision. This inclusivity fosters collaborative efforts and ensures that decision-making is not limited to a single central authority. By avoiding centralised management and dictatorship, decentralised decision-making allows for fair representation and participation of all stakeholders.

In recent times, blockchain technology has emerged as a powerful tool to enhance trust in decentralised decision-making processes [58, 59, 60, 61]. Blockchain's decentralised nature supports collaborative intelligence and decision-making without the need for intermediaries. Its transparency ensures fairness, and immutability ensures that decision-making data remains tamper-proof.

The impetus behind this thesis revolves around the creation of a privacy-preserving decision-making system that capitalises on the intricate cryptographic building blocks, thereby ensuring a robust amalgamation of security and privacy. This endeavour not only draws inspiration from prior research but also expands the horizons of privacy-preserving protocols, decentralised systems, and cryptographic methodologies.

This chapter focuses on the intersection of blockchain and decision-making, particularly two branches: blockchain governance and blockchain voting. Blockchain governance explores how blockchain-based decentralised decision-making systems address democratic blockchain evolution issues. On the other hand, blockchain voting encompasses decentralised voting protocols for blockchain consensus and blockchain-based e-voting schemes for various applications. The aim of this chapter is to delve into the most representative blockchain-based decision-making systems and highlight the research gap.

3.2 Blockchain Governance

As the world becomes increasingly reliant on information systems, we are entering an era known as "Code is the law" [62]. In this paradigm, private actors, such as developers or project owners, embed their values and thoughts into the products through coding, which in turn regulates and governs activities and actions. This type of regulatory framework automates processes based on predefined rules and regulations, but it can also lead to challenges in terms of fairness and adherence to traditional regulations. One of the main issues arises when the preferences and needs of end users are not fully taken into account during the coding process. As a result, there may be a lack of fairness or deviation from established norms and regulations. For instance, when existing code fails to satisfy all stakeholders, any modifications to the code can potentially impact the interests of different parties.

A concrete example of this phenomenon can be observed in the context of blockchains and the problem of "hard forks" [63]. As new requirements and preferences emerge, blockchain platforms are forced to iterate and update, which can lead to divisions within the community, duplication of assets, fraudulent activities, and security risks. These challenges highlight the need for innovative solutions that can address the complexities of decentralised decision-making and ensure that the preferences and interests of all stakeholders are considered. This is where the research and contributions in this field become crucial, as there is a clear need to address the gap in current decentralised decision-making systems. By applying advanced cryptographic techniques, the aim is to enhance trust, privacy, and security in these systems, and provide innovative solutions that consider the preferences and interests of all stakeholders.

To address the challenges of forks and ensure the sustainable evolution of blockchains, extensive research has been conducted in the field of blockchain governance [64, 65, 66]. Blockchain governance can be categorised into two main approaches: off-chain governance and on-chain governance. In off-chain governance, decisions are typically made by developers and key decision-makers of the blockchain platform, resulting in a centralised decision-making process. However, this approach often lacks sufficient incentives for proposers, leading to a concentration of decision-making power among a minority. On the other hand, on-chain governance involves a decentralised on-chain voting mechanism, allowing community members of the blockchain platform to actively participate in the decision-making process regarding the platform's evolution. This approach aims to foster inclusivity and ensure a broader representation of stakeholders in the decision-making process.

To provide a comprehensive overview and comparison of these governance mechanisms, Table 3.2 presents a detailed analysis of various blockchain projects and their governance models. This comparison aims to shed light on the strengths, weaknesses, and challenges associated with each approach, paving the way for innovative solutions that enhance trust,

privacy, and security in blockchain governance systems.

Table 3.1: Comparison of Governance Mechanisms in Various Blockchain Projects

Project	**Governance Type**	**Decision-Making Process**	**Participants**	**Incentive**	**Challenges**
Bitcoin[12]	Off-chain	BIP submission community review	Community member	✗	Lack of formal voting potential for Sybil attacks
Ethereum[67]	Off-chain	EIP submission community feedback	Community member	✗	lack of identity verification Lack of formal voting potential for Sybil attacks
Compound[68]	On-chain	Stake COMP tokens vote on proposals	COMP token holders	✗	Pseudonymity risks ballot secrecy, potential protocol changes, centralisation concerns
Zcash[69, 70]	On-chain	ZIP submission committee voting	5 Committee members	✓	Small committee size high risk of corruption
Tezos[8, 9]	On-chain	Proposal submission adaptive phases	Delegates with tokens, public voting	✓	Continuous commitment required, public ballots
Polkadot[10]	On-chain	Adaptive majority thresholds batch approval	Token holders Council members	✓	Public ballots complexity in voting thresholds
Dash[71]	On-chain	Proposal submission masternode voting	Masternodes with 1000 DASH	✓	No explicit accountability for voting, low participation incentives
Cardano[11]	On-chain	Proposal submission, CA and vCA review	ADA holders CAs, vCAs	✓	Centralisation in vCA selection, lack of accountability

3.2.1 Off-chain governance

Bitcoin Improvement Proposals (BIPs, [12]) serve as a crucial mechanism for supporting updates and enhancements to the Bitcoin network. These proposals cover a wide range of changes, including updates to the network protocol, block and transaction validity rules, algorithm efficiency improvements, software compatibility enhancements, new features, process updates, and design issues related to Bitcoin. Anyone can submit their ideas for consideration as a BIP by posting them to an official mailing list for community review. The author of the initial idea or draft BIP takes on the responsibility of shepherding the discussions within the community and collecting feedback for further evaluation. The BIP editors, who are in charge of BIP draft editing and administration, play a critical role in the process. They ensure that BIP drafts are well-structured, complete, and conform to the standard BIP format. Additionally, they review the discussion history, assign BIP numbers, merge pull requests, monitor BIP changes, and handle the transfer of BIP ownership when needed.

Once a BIP has gone through community review, complied with the standard, and been reviewed by the BIP editors, it is submitted to the BIP's git repository as a pull request. For soft-forking changes that do not involve predefined flag timestamps or flag block heights, consensus is achieved with a threshold of 95% agreement, as seen in BIP 34, BIP 65, and BIP 66. However, for hard-forking changes, agreement must be reached by the entire community due to their more significant impact on the network.

The process of submitting, reviewing, and generating BIPs exhibits informal characteristics, which can lead to challenges in the decision-making process. These informality aspects manifest in several areas, including suffrage issues, eligibility criteria for participants, and the absence of specific rules guiding the voting process for arriving at final decisions. As a result, the decision-making process may suffer from unfairness, as proposers play a significant role in encouraging community consensus. However, there is no complete incentive structure, identity management, or accountability mechanism in place for proposers and community participants, which can lead to irresponsible behaviours and hinder the involvement and sustainability of BIPs. Furthermore, the lack of an identity management mechanism leaves the process vulnerable to Sybil attacks, where individuals can create multiple fake identities to influence decision outcomes. Moreover, the straight verifiability of the honesty in feedback submissions by proposers is not guaranteed unless BIP editors meticulously trace every piece of feedback back to the forums. This lack of transparency can lead to doubts about the integrity of the decision-making process.

The current informal aspects of the decision-making process in BIPs, where the minority holds strong decision-making powers, create a lack of verifiability regarding the final decisions reached. This lack of transparency can lead to concerns about the fairness and legitimacy of the decision-making outcomes. Additionally, the positional power of BIP editors can make them more easily trusted and influential within the community, potentially affecting the objectivity and inclusivity of the decision-making process.

Furthermore, the current decision-making phase exposes community members' opinions publicly on forums and git repositories, leading to a lack of privacy for participants' viewpoints. This lack of privacy can deter some individuals from freely expressing their opinions, potentially leading to skewed decision outcomes.

The adoption of forums, email, and manual administration in the decision-making process introduces risks of single-point failures, completeness failures, and accuracy failures. These risks are contrary to the promise and spirit of blockchain, which aims to achieve decentralised and trustless governance systems.

The widely adopted enterprise blockchain platform, Ethereum, utilises Ethereum Improvement Proposals (EIPs, [67]) as the primary mechanism for introducing new features, gathering community feedback, and making decisions regarding the evolution of the Ethereum ecosystem. EIPs share similarities with BIPs, and as a result, they inherit the main drawbacks discussed earlier with respect to off-chain governance.

As a response to these challenges and limitations in the current decision-making process, this thesis seeks to explore and propose privacy-preserving solutions based on cryptographic protocols. By introducing formal and privacy-preserving mechanisms, the proposed solutions aim to enhance the trust, involvement, and efficiency of decentralised governance systems, while safeguarding the privacy of participants' opinions and ensuring the integrity of the decision-making process. The goal is to address the issues related to identity management, verifiability, fairness, and privacy, ultimately improving the overall effectiveness and credibility of blockchain governance.

3.2.2 On-chain governance

In the Ethereum ecosystem, projects like Compound [68] have been actively exploring and implementing decentralised solutions for facilitating the exchange of time value among various Ethereum assets, such as Ether, ERC-20 stablecoins, and utility tokens. The main objective of these solutions is to tackle challenges like mispricing in borrowing mechanisms and negative yields resulting from storage costs and exchange risks. By leveraging the power of smart contracts and blockchain technology, these decentralised exchanges offer transparent, efficient, and secure mechanisms for asset exchange, thereby contributing to a more inclusive and resilient financial ecosystem not only on Ethereum but also across other blockchain platforms.

During its early stages, Compound's governance was centralised, where decisions like the interest rate model for computing borrowing and supplier's earned interest rates were made by the Compound platform itself. This centralisation raised concerns about transparency, fairness, and inclusivity in the decision-making process. To establish a decentralised administration model, Compound has introduced the COMP token, which grants holders a one-to-one voting power in Compound governance. This governance framework enables COMP token holders to participate in crucial decisions, such as the addition of new markets,

adjustments to collateral factors, and updates to interest rate models. Proposers can submit proposals on-chain by staking a minimum of $60,000$ COMP tokens. These proposals undergo a voting process, where COMP holders engage in approval voting during a three-day voting period. Once a proposal garners a majority of votes, it can be implemented after a minimum time lock of two days, ensuring security considerations are met.

However, the current Compound governance system employs pseudonymous addresses for voting and delegation, without any encryption. This setup poses a challenge to the confidentiality of ballots in Compound. To address this limitation, we aim to enhance the privacy and confidentiality of the voting process, while maintaining the decentralised nature and efficiency of Compound governance. By introducing formal privacy-preserving mechanisms, the proposed protocols can enhance the trust, privacy, and integrity of Compound governance, thereby fostering a more secure and inclusive decision-making environment.

Zcash [69, 70] is a privacy-focused cryptocurrency that allocates 20% of mining rewards to its own development, driven by an on-chain governance system. To propose changes and improvements, anyone can submit Zcash Improvement Proposals (ZIPs) to the project's git repository. The proposers are required to engage with the community through forums and document feedback to build consensus.

The decision-making process involves a voting mechanism where ZIPs are voted on by a committee of five Zcash Community Grants Committee members. These committee members are selected through a voting process from all applicants and receive fixed payments per calendar month. However, the small size of the committee (only five members) limits the scalability of decision-making. Additionally, this governance structure may lead to potential issues with minority decision-makers, including higher risks of corruption and coercion.

Despite its on-chain governance system, this approach may not fully realise the promise of decentralised and inclusive decision-making due to its limitations. In light of these challenges, we aim to address the shortcomings of existing governance models by enhancing the privacy, scalability, and fairness of decision-making processes. Through the introduction of formal privacy-preserving mechanisms, the proposed protocols aim to promote more inclusive and secure decentralised governance systems.

Decred's governance system [72], based on Proof of Stake voting, plays a crucial role in making decisions related to the Decred treasury and action proposals, such as software upgrades and policy changes. To prevent proposal spamming, proposers are required to pay a small fee when submitting proposals. Decred holders participate in the governance process by locking their Decred (DCR) to gain voting power and submit ballots.

However, some aspects of Decred's governance raise concerns. While Decred allocates 30% of the block reward to incentivise voting in each block, the specific reward and incentives for voting behaviours are not clearly defined. Additionally, DCR holders can withdraw their locked stakes 250 blocks after voting, and the final tally results are revealed only after 8064 blocks. This design allows dishonest DCR holders to evade punishment for their malicious

votes by withdrawing their locked stakes before the tally is revealed.

To address these challenges and enhance the integrity and fairness of Decred's governance, cryptographic protocols that incorporate privacy-preserving mechanisms and formal verifiability should be tackled. The primary goal is to promote active and honest participation in the governance process while safeguarding the privacy of individual stakeholders' voting decisions. By introducing transparent and effective incentive mechanisms, the proposed protocols aim to create a more secure and efficient governance system for Decred and other blockchain platforms. These protocols will be designed to encourage responsible and informed decision-making, ensuring that all stakeholders can participate with confidence, knowing that their voting choices remain confidential.

The Maker protocol [73], operating under MakerDAO's Multi-Collateral Dai (MCD) System powered by MKR and Dai tokens, employs a collateral assets system on the Ethereum platform. Proposals related to the Maker protocol can be submitted by anyone, but only MKR token holders are eligible to participate in the decision-making process for managing the protocol. The scope of decisions within the Maker protocol includes matters such as introducing new collateral asset types, adjusting risk parameters, setting the Dai savings rate, determining oracle feeds and emergency oracles, as well as making decisions regarding protocol interruptions and upgrades. Additionally, the protocol allows for assigning permissions to external actors, including oracles and global settlers.

Similar to the setting in Compound, winning proposals in the Maker protocol undergo a delay of up to 24 hours before becoming active, a measure taken for security reasons. The entire decision-making process in the Maker protocol is conducted publicly, with no explicit incentives provided to participants.

Given the decentralised nature of the governance system, there is a need to explore privacy-preserving and incentive mechanisms that can enhance the integrity and efficiency of decision-making within the Maker protocol. By incorporating cryptographic techniques and introducing appropriate incentives, we aim to strengthen the governance process and promote responsible and active participation among MKR token holders. Ultimately, the goal is to contribute to the development of a more robust and inclusive governance model for the Maker protocol and other similar blockchain platforms.

Tezos [8, 9] is a blockchain platform that promotes self-amendment, user-governed, and user-centric principles through its on-chain governance system. The decision-making process in Tezos is divided into five periods, each comprising approximately 102400 blocks. These periods encompass proposal submission, voting, testing, and final adoption. Delegates, who possess at least 8000 tokens, can submit and vote on proposals. During voting, delegates can also delegate their tokens to other delegates. The decision-making mechanism utilises approval voting, where proposals obtaining a super-majority of votes can be processed in the subsequent period, provided the participation reaches the current participation threshold.

A study by Khan et al.[74] analysed blockchain governance, including Tezos, from the perspective of Information Technology governance. They evaluated decision-making through

voting on proposals submitted in Tezos. If a majority is reached, the corresponding proposal is implemented and deployed on the main blockchain. The researchers computed payoff matrices for different blockchain governance models and predicted optimal governance strategies using Nash equilibrium.

Despite its innovative approach, Tezos faces some challenges in its governance system. One major concern is the lack of privacy and secrecy for delegates, as their delegated ballots are made public on-chain. Additionally, the system lacks sufficient incentives for participants, and if the participation threshold is not reached, the decision-making process may be rendered invalid and terminated prematurely. This raises questions about the overall sustainability and efficiency of the Tezos system. Furthermore, participants are required to commit continuously over 102400 blocks (approximately 35.5 days), which may expose them to additional risks, such as manipulation or system failure, that cannot be ignored.

To address the challenges posed by the long continuous commitment required in Tezos' governance system, we propose to explore novel cryptographic protocols that enable stakeholders to participate securely and efficiently without the need for continuous commitment throughout the entire decision-making process.

One approach is to develop threshold cryptographic techniques that allow stakeholders to collectively participate in decision-making without having to maintain continuous commitment. Threshold cryptography enables distributed and decentralised decision-making by dividing the private keys and responsibilities among multiple participants, known as threshold participants. These participants collaborate to create cryptographic signatures or cast votes without any single participant having complete control over the process. By distributing the commitment and decision-making responsibilities, the burden of continuous commitment is alleviated, and stakeholders can contribute efficiently without being tied to the system for extended periods.

Another potential solution is the use of privacy-preserving protocols that allow stakeholders to cast their votes anonymously and securely. Privacy-preserving voting schemes, such as mixnets or zero-knowledge proofs, can be utilised to ensure that votes are cast and tallied in a way that preserves the privacy of individual stakeholders. This ensures that participants can contribute their votes to the decision-making process without revealing their identity or continuously committing their tokens.

Furthermore, we can introduce appropriate incentive mechanisms to encourage active and meaningful participation in Tezos' governance. For example, stakeholders could be rewarded for voting or participating in specific governance decisions. These incentives can be designed to align the interests of stakeholders with the long-term success and sustainability of the platform, encouraging them to engage in the decision-making process without the need for continuous commitment.

Overall, by combining threshold cryptography, privacy-preserving protocols, and effective incentive mechanisms, we aim to enhance the efficiency and fairness of Tezos' governance system while addressing the challenges posed by the long continuous

commitment requirement. This will foster a more inclusive and robust decision-making process that promotes active participation from stakeholders without compromising their privacy or imposing excessive burdens.

Polkadot [10] employs an on-chain governance system to ensure the alignment of stakeholders' interests during the evolution of the platform. The governance process utilises adaptive super-majority thresholds and batch approval voting, which varies depending on the types of proposers involved, including stake holders, the council, and the technical committee. The three types of decision-making processes are known as Positive Turnout Bias (Super-Majority Approve), Negative Turnout Bias (Super-Majority Against), and Simple Majority.

In Polkadot's governance system, proposers, who can be stake holders or council members, have the option to lock their stakes to submit proposals, with a maximum of 100 proposals allowed. All active token holders and the council are eligible to participate in the network upgrade decision-making process, which spans a duration of 28 days. The weight of each participant's vote is determined by both their voting choices and the length of time their stake has been locked, with a time frame ranging from 2^1 to 2^8 days.

The Council in the Polkadot governance system is composed of 13 members, and its selection is based on the Sequential Phragmén Method [75]. To become a Council member, one must lock 5 DOT tokens, which represents the passive stakeholders in the network. However, it has been noted by [74] that this selection method requires a considerable amount of time, affecting the constant block generation time in the process.

The role of the Council includes approving treasury proposals and allocations, proposing sensible referendums, vetoing uncontroversially dangerous or malicious referenda, and selecting the Technical Committee (TC). During voting, some councillors may choose to abstain, which can impact decision-making efficiency. To address this, Polkadot selects a prime member from the councillors using Borda Count voting. This ensures that all councillors make explicit votes or that abstained votes are replaced with the prime member's votes.

TC is determined through a simple majority voting process conducted by the Council. The TC is responsible for proposing sensible referendums using the Democracy pallet, cancelling proposals with unanimous agreement (with the deposit of cancelled proposals being burnt), fixing bugs, reversing faulty runtime updates, adding new features, and submitting emergency proposals. However, the fairness and honesty of the selection process for the TC cannot be guaranteed, as it is directly chosen by the Council.

However, one limitation of the current system is that all votes are made public on the blockchain, which compromises the privacy of the ballots. To address this issue, it is crucial to explore and develop privacy-preserving protocols that enable stakeholders to cast their votes securely and anonymously. By introducing privacy measures, stakeholders can participate in the decision-making process with confidence, knowing that their voting choices remain confidential and cannot be traced back to their identities.

Additionally, the use of adaptive super-majority thresholds and batch approval voting can lead to more efficient and effective decision-making processes. These mechanisms can help strike a balance between the need for consensus and the timeliness of decision-making, ensuring that the governance process remains responsive to the evolving needs of the platform and its community.

By studying and implementing privacy-preserving protocols and refining the decision-making mechanisms, we aim to enhance the integrity, inclusivity, and efficiency of Polkadot's governance system. The main goal is to promote active participation from stakeholders while safeguarding the privacy and confidentiality of their voting choices, leading to a more robust and sustainable governance process for the platform.

To ensure the sustainability of Dash [71], the platform adopts an on-chain governance system that encourages collaborative decision-making among all masternodes. The distribution of block rewards allocates 90% for mining and masternode rewards, while 10% is dedicated to the decentralised governance monthly budget, supporting Dash's development.

In the Dash governance model, any participant can submit a proposal by paying a small fee of 5 Dash, which is burned upon submission. Proposers are expected to actively lead discussions within the Dash community forum, providing supplemental information to assist masternodes in making informed decisions. Masternodes, possessing a minimum of 1000 DASH, can cast their votes with options of yes, no, or abstain, using an approval voting scheme. Additionally, masternodes can publicly delegate their voting rights to other masternodes within approximately one month. Proposals garnering a net approval of at least 10% of the masternode network can secure funding.

However, the current Dash governance lacks explicit mechanisms for accountability and rewarding. Malicious or random voting behaviour by masternodes does not require them to bear any consequences. Moreover, the system does not actively encourage masternode participation.

To enhance the accountability and incentives of masternodes in the governance process, the thesis seeks to develop cryptographic protocols and incentive structures. By introducing formal accountability mechanisms, we aim to motivate masternodes to actively participate in the decision-making process, leading to a more responsible and efficient governance system for Dash.

Cardano [11] employs an on-chain governance system, employing fuzzy threshold voting, to allocate its treasury funds. Within the Cardano project, any participant can submit proposals, which are subsequently reviewed and filtered by Community Advisors (CAs) and Veteran Community Advisors (vCAs).

The voting process in Cardano allows voters with a minimum of 500 ADA to cast yes/no/abstain votes for any proposals, without the need to lock stakes. The distribution of treasury funds includes fixed percentages allocated to reward voters, CAs, and vCAs. Notably, CAs have the potential to be promoted to vCAs based on their outstanding contributions during the proposal submission phase.

Despite its merits, Cardano exhibits a centralisation concern, particularly in the selection process of vCAs from CAs. As CAs can be individuals from various backgrounds within the Cardano community, the centralisation introduced in promoting CAs to vCAs does not guarantee the honesty and fairness of the system. Moreover, there is no explicit penalty or mechanism for holding CAs and vCAs accountable for any malicious behaviour. This lack of accountability may allow vCAs with ulterior motives to filter out promising proposals, preventing them from gaining support.

To address these issues and improve the overall integrity of Cardano's governance, cryptographic protocols should be developed to introduce more formal accountability mechanisms. By incorporating such mechanisms, the proposed solutions aim to foster a more transparent and fair governance system, encouraging active participation and responsible decision-making among all stakeholders in Cardano.

3.3 Blockchain Voting

In the context of blockchain voting, two primary categories can be identified: blockchain-based e-voting systems and blockchain voting systems for achieving consensus among miners regarding data and transactions. Blockchain-based e-voting systems [76], addresses challenges encountered in traditional e-voting or voting systems. These challenges may include ensuring ballot secrecy, data integrity, reliability, and other issues related to the transparency and security of the voting process. By leveraging the capabilities of blockchain technology, these systems aim to overcome these challenges and provide a more trustworthy and efficient voting experience.

In contrast, the second category involves voting systems utilised within the blockchain platform itself. These systems play a crucial role in achieving consensus among miners regarding the agreement on data and transactions. Consensus mechanisms are essential for maintaining the integrity and consistency of the blockchain, ensuring that all nodes agree on the validity of new data and transactions added to the chain.

However, this section will primarily focus on the current state of blockchain-based e-voting systems. We will explore the innovative approaches and potential challenges faced in implementing such systems using blockchain technology. By analysing the existing e-voting solutions and their limitations, we aim to propose novel cryptographic protocols and privacy-preserving mechanisms to enhance the security, privacy, and efficiency of blockchain-based e-voting systems. By categorising and focusing on blockchain-based e-voting systems, the section can provide a comprehensive analysis of the challenges faced in the field and offer novel solutions to improve the voting experience and safeguard the integrity of the democratic process.

E-voting has emerged as a promising alternative to traditional paper-based voting systems by leveraging electronic systems for ballots submission, vote recording, and tally calculation. Extensive research has demonstrated that e-voting systems offer numerous advantages, including cost reduction, increased engagement, higher accuracy, improved voter turnout, enhanced effectiveness and efficiency of democracy, as well as enhanced security and privacy in the voting process [77, 78, 79].

Several countries, such as France, Spain, the United Kingdom, Estonia, and the United States, have already adopted e-voting systems for national elections [80]. A typical e-voting system consists of three main phases: voters' registration, ballot casting, and automatic tally result calculation. In order to ensure the fundamental principles of fairness [81, 82], verifiability [83], and privacy [84], e-voting systems encrypt ballots and compute tally results in a verifiable manner. Moreover, all voting data must be immutable and retained to ensure the integrity and transparency of the voting process [85].

Keeping these characteristics in consideration, e-voting systems have the potential to revolutionise the democratic process, making it more accessible, secure, and reliable for citizens. However, the adoption of e-voting also introduces unique challenges, such

as ensuring the authenticity of voters, protecting against cyber threats, and maintaining voter privacy. To address these challenges, innovative cryptographic protocols and privacy-preserving mechanisms are essential, enabling e-voting systems to fully realise their potential and enhance democratic practices on a global scale.

Following this discussion, a comparative analysis of various e-voting systems is presented in Table 3.2,3.3, and 3.4, which elucidates their respective advantages and challenges. This table synthesizers the operational features and cryptographic mechanisms that each system employs, alongside the specific vulnerabilities and limitations they might encounter. As illustrated in the table, while some systems excel in ensuring voter privacy and ballot secrecy through advanced cryptographic methods such as zero-knowledge proofs, others focus on enhancing the transparency and verifiability of the voting process, albeit sometimes at the expense of complexity or scalability.

Table 3.2: Comparison between Blockchain-Based E-Voting Systems and Traditional E-Voting Systems

Aspect	**Blockchain-Based E-Voting Systems**	**Traditional E-Voting Systems**
System Type	Decentralised, operates on blockchain technology.	Centralised, operated by authorised bodies.
Security	High security through cryptographic methods like hashing and encryption. Resistant to DDoS attacks.	Vulnerable to tampering and DDoS attacks due to centralised architecture.
Privacy	Enhanced privacy using methods like zero-knowledge proofs, anonymisation techniques.	Privacy risks due to potential internal misuse or data breaches.
Integrity and Transparency	High data integrity and transparency ensured by the immutable nature of blockchain. Publicly verifiable transactions.	Integrity depends on the security of central servers; transparency is limited by the organisation's disclosure.
Voter Anonymity	Advanced cryptographic techniques support strong anonymity while ensuring verifiability.	Anonymity can be compromised if the central system is breached.
Accessibility and Ease of Use	Potentially complex for average users due to the need for digital literacy in interacting with blockchain systems.	Generally user-friendly with familiar interfaces, but reliant on proper function of centralised systems.
Cost Efficiency	Potentially lower operational costs in the long run due to reduced physical infrastructure needs.	Higher operational and maintenance costs due to physical and IT infrastructure.
Scalability	Challenged by the need for consensus mechanisms, which can limit transaction throughput as the network grows.	Easier to scale within controlled environments but at the risk of increasing central point of failure vulnerabilities.
Auditability	High; every transaction is logged and verifiable, making audits straightforward and transparent.	Dependent on the integrity of the central authority; audits can be challenging if data manipulation occurs.
Innovations and Challenges	Innovations in privacy-preserving mechanisms, but challenges include ensuring universal verifiability and dealing with the scalability of consensus protocols.	Advances in user verification and system efficiency, but challenges remain in data security and system integrity assurance.
Adoption Examples	Used in pilot projects and smaller scale elections (*e.g.*, Estonia's i-voting).	Widely used in national elections in various countries, including the US and Estonia.
Reliability	High reliability as long as the network is operational; resilience to node failures.	Reliability depends heavily on the central server; susceptible to single points of failure.

Table 3.3: Protocols in Blockchain-Based E-Voting Systems 1

Protocol	**Key Features**	**Privacy Mechanisms**	**Integrity & Transparency**	**Challenges and Limitations**
[86]	Allows voters to change their ballots during the voting period.	None specified, lacks privacy.	Lacks ballot consistency and auditability.	Ensuring privacy and maintaining ballot consistency.
[87]	Utilises zero-knowledge proofs for tally validation without decrypting ciphertexts.	Zero-knowledge proofs ensure voter privacy and tally secrecy.	High transparency with verifiable tally computation.	Requires robust implementation of zero-knowledge proofs to avoid vulnerabilities.
[88]	Employs Shamir secret sharing and circle shuffle techniques for fair voting.	Relies on trusted authority for shuffling ballots.	Dependence on a central shuffling authority may compromise integrity if the authority is compromised.	Trust dependency on shuffling authority, privacy issues if authority is compromised.
[84]	Uses Hyperledger as a private and permission-based network for end-to-end privacy.	High privacy due to permissioned blockchain structure.	Lacks transparency and fairness; not fully verifiable.	Does not support verifiability or fairness adequately, limiting its appeal for public elections.
[89]	Proposes an always-on-voting framework that supports changing decisions between elections.	No specific privacy mechanisms detailed.	Supports only one winner, limiting complex election types.	Limited to simple election models; scalability and complexity management in larger settings.

Table 3.4: Protocols in Blockchain-Based E-Voting Systems 2

Protocol	**Key Features**	**Privacy Mechanisms**	**Integrity & Transparency**	**Challenges and Limitations**
[90]	Introduces self-tallying protocols that support score voting with abstentions considered.	Complex computations required for abstentions.	Ensures integrity and transparency through self-tallying capabilities.	Additional computational overhead, practicality in large-scale elections.
[15]	Aims to address the drawbacks of traditional elections with a blockchain approach but does not detail specific cryptographic methods.	Anonymity and privacy not sufficiently addressed.	Visibility of cast votes may compromise ballot secrecy.	Lack of detailed cryptographic measures and potential visibility of choices during voting.
[91]	Leverages blockchain technology, homomorphic encryption, and zero-knowledge proofs using Paillier encryption.	Homomorphic encryption and zero-knowledge proofs.	High level of privacy and security, with verifiable results.	Complexity in implementation and potential need for high computational resources.
[92]	Operates on Ethereum 2.0, using smart contracts to ensure voter anonymity, privacy, verifiability, and fairness.	Smart contracts enforce privacy but lack proof for encrypted ballots.	Enhanced security with smart contracts, but concerns over vote tampering.	Needs more robust proofs for encrypted ballots to ensure no tampering occurs.
[93]	Focuses on anonymous and mutable voting mechanisms.	Provides voter anonymity but lacks detailed mechanisms for vote encryption.	High integrity due to blockchain, but verification of encrypted ballots is unclear.	Issues with encrypted ballot proofs, potentially affecting voter confidence in the security of their vote.

Despite the advancements in privacy-preserving e-voting protocols, traditional e-voting systems still face significant challenges related to central authority control [80]. Malicious election authorities can manipulate tally results by altering the e-voting system's code, and data generated during the voting process may be shared with third parties without the voters' knowledge or consent [94, 95]. Moreover, centralised e-voting systems are vulnerable to single-point-of-failure, compromising the integrity, confidence, and correctness of tally results if the system malfunctions [96]. For instance, the Estonian I-voting system was found to be susceptible to denial-of-service attacks, leading to the failure of accepting new votes [96]. Additionally, vulnerabilities in trapdoor commitment schemes used in some e-voting systems allowed adversaries to alter ballots, leading to discontinuation by governments [97]. The lack of transparency in the voting process further hinders the trust and confidence stakeholders place in traditional e-voting systems [98]. To address these issues, the adoption of decentralised and privacy-preserving mechanisms, as seen in blockchain-based e-voting systems, can significantly improve the integrity, transparency, and trustworthiness of the voting process.

The decentralised nature of blockchain-based e-voting systems offers inherent resistance to distributed denial-of-service (DDoS) attacks [99, 76]. Even if individual nodes are compromised, the overall system can continue to function independently, and when the affected node rejoins the network, it can synchronise with a consistent view of the system. Storing voting data on the blockchain ensures the integrity and verifiability of the voting data, preventing malicious tampering [100, 76] due to the transparency and immutability properties provided by the blockchain. The adoption of blockchain-based e-voting systems has been considered as a means to increase public confidence and enhance voting reliability, leading to their implementation in government elections in countries such as the Netherlands, Estonia, and South Korea [101]. By leveraging the features of blockchain technology, these systems aim to offer a more secure, transparent, and trustworthy voting process for citizens and stakeholders.

Hardwick *et al.*[86] proposed a blockchain-based e-voting system that allows voters to alter their ballots during the voting period. However, their protocol lacks guarantees of privacy, ballot consistency, and auditability. To address the privacy concerns, Zhao and Chan [87] utilised zero-knowledge proofs to ensure tally validation without the need to decrypt ciphertexts. Bartolucci *et al.*[88] employed Shamir secret sharing and circle shuffle techniques to achieve fair voting on the blockchain. However, their protocols rely on a trusted authority for shuffling, and if this authority is compromised, the link between voters and their submissions can be revealed.

Yu *et al.*[102] introduced a platform-independent electronic voting system leveraging short linkable ring signatures, which effectively eliminates constraints on the number of voters and candidates. Similarly, Li *et al.*[103] proposed the Lat-voting protocol, which utilises a prefix-based linkable and trackable anonymous authentication scheme to ensure both anonymity and public traceability. However, a common limitation of these protocols is

their reliance on a central authority for calculating the voting results, which may compromise the decentralised nature of blockchain-based systems.

An alternative approach proposed by Zhang *et al*.[84] uses Hyperledger [104, 102] as a private and permission-based network for end-to-end privacy in on-chain voting. However, their solution lacks support for verifiability and fairness. Venugopalan *et al*.[89] introduce an always-on-voting framework that enables voters to change their decisions between two main elections. However, their protocol can only support one candidate as the final winner, limiting its applicability to more complex voting scenarios.

Addressing the need for self-tallying capabilities, Yang *et al*.[90] proposed a self-tallying election system that supports score voting. This innovative protocol considered scenarios where voters choose to abstain from voting, offering a comprehensive solution for a broader range of voter behaviours. However, the proposed solution required additional computations for those who have voted, introducing potential complexities and reducing practicality.

The treasury system proposed in *et al*.[13, 16] represents one of the few provably secure on-chain decision-making systems. It aims to manage funding and distribute power to empower the community of individuals, serving as an enabling force for change and progress in the blockchain platform. However, their system [13, 16] has several drawbacks.

Firstly, it uses the Distributed Key Generation (DKG) protocol by Gennaro *et al*.[38] for each voting event. The overall communication complexity of this DKG protocol is $\mathcal{O}(n^2)$ per key, where n is the number of parties involved. Consequently, the committee size cannot scale efficiently, posing limitations on the system's ability to accommodate a large number of participants. Secondly, in their voting protocol, voters and/or experts are required to make decisions on every submitted proposal. This approach demands significant voting effort, particularly when the number of proposed proposals becomes large. Such a design might not be practical and efficient in real-world scenarios. Thirdly, voting committee members are required to hold their key shares throughout the entire voting period, which could take 1-3 months in practice. This design introduces a prolonged exposure to risk, as the system becomes more vulnerable to potential failures and disruptions. Moreover, the proposed treasury system does not explicitly support participatory budgeting (PB) [105, 106], which could limit the inclusiveness and democratic nature of the decision-making process. Lastly, despite the authors' claim for end-to-end privacy, their protocol fails to provide verifiability of tally results by either the public or participants, undermining the transparency and trustworthiness of the system. Overall, while the treasury system proposed in [13, 16] represents an important step towards on-chain decision-making, it still faces significant challenges and limitations that need to be addressed for broader adoption and real-world applicability.

Yang *et al.* [14] introduced a blinding factor in the ballots, which can be cancelled out during the tally computation phase. The blinding factor is used to obfuscate the actual vote cast by each voter, ensuring that the voting choices remain private and cannot be directly linked to individual voters. While their protocol successfully provides privacy for the ballots,

it is worth noting that it requires the revelation of all private keys during the tally computation phase. This requirement may raise concerns regarding the security and confidentiality of sensitive information, as exposing private keys can potentially lead to vulnerabilities and compromise the integrity of the system.

Khan *et al.*[15] propose a blockchain-based approach to address the drawbacks of traditional elections. However, their system suffers from limitations such as visibility of casted votes during the voting process and the absence of voter anonymity. The e-voting platform, zVote [91], offers a solution to the security and privacy challenges in remote e-voting by leveraging blockchain technology, homomorphic encryption, and zero-knowledge proofs using Paillier encryption. To ensure privacy, their protocol necessitates the use of anonymous communications to prevent voter identification.

In the blockchain-based voting system called DVTChain [92], which operates on Ethereum 2.0 and employs smart contracts, measures are taken to ensure voter anonymity, privacy, verifiability, and fairness while enhancing security and minimising associated costs. However, it is worth noting that the system lacks adequate proof for the encrypted ballots, which may give rise to concerns regarding potential vote tampering and could lead to a loss of voter confidence. The issue can also be observed in Li *et al.*'s work, known as AMVchain [93].

The existing endeavors in this domain underscore the continuous strides towards crafting blockchain-based electronic voting systems that encompass diverse attributes and inherent trade-offs. Nonetheless, there persist challenges that necessitate concerted attention to engender a holistic and privacy-centric solution for the secure and streamlined conduct of voting on the blockchain. Notably, while several of the aforementioned initiatives have made noteworthy contributions, a gap remains in the comprehensive security analysis of these protocols within the universal composability framework, as established by Canetti [20]. This aspect is particularly vital, especially in the context of blockchain-based implementations, as it validates the efficacy of the proposed protocols when combined with other cryptographic mechanisms. Consequently, the veracity and robustness of these works come under scrutiny if they are to coexist harmoniously within the broader cryptographic ecosystem inherent to blockchain technologies.

3.4 Summary

This chapter delved into the landscape of existing research and implementations in the field of decentralised decision-making systems on blockchain. This comprehensive review highlights various approaches, methodologies, and technologies employed to address the challenges and opportunities posed by collaborative decision-making in a decentralised context.

The chapter begun by outlining the importance of decentralised decision-making in diverse domains, such as blockchain development funding, oracle systems, prediction markets, and blockchain governance. These applications showcase the significance of efficient and transparent decision-making mechanisms for ensuring the sustainability and growth of decentralised ecosystems.

The review encompassed various methodologies for achieving decentralised decision-making, including voting-based systems, consensus mechanisms, and liquid democracy models. The advantages and limitations of each approach are examined in terms of scalability, privacy, security, and usability. Notable protocols such as the "treasury system" are discussed in depth, with a focus on their cryptographic underpinnings, privacy guarantees, and decision-making mechanisms.

The exploration of related work also reveals the challenges faced by existing approaches, such as scalability bottlenecks, lack of flexibility in participation, and privacy concerns. While these systems have made significant strides in enhancing decentralised decision-making, there is still a need for novel solutions that can overcome these challenges and provide a more holistic and practical framework for collaborative decision-making on blockchain.

In the forthcoming chapter, we will embark on a comprehensive exploration of the system design underpinning the proposed decision-making system. This intricate design aims to bridge the gaps elucidated in the preceding chapter, where the limitations of existing decentralised decision-making systems were scrutinised. Through a meticulous and thoughtful approach, our system design endeavours to overcome these identified research gaps and introduce innovative solutions that elevate the landscape of collaborative decision-making on the blockchain.

Chapter 4

System's Design

> Science is the century-old endeavour to bring together by means of systematic thought the perceptible phenomena of this world into as thorough-going an association as possible.
>
> Albert Einstein

4.1 Overview

This chapter presents the system's design of the proposed privacy-preserving decision-making system built on the blockchain (Figure 4.1). The primary goal of this system is to maximise the influence of decision-makers' values, preferences, and beliefs in achieving consensus decision-making [107]. It supports collaboration, cooperation, egalitarianism, inclusion, and participatory budgeting [105] (as discussed in Section 4.2). The system also promotes delegative democracy [108], empowering individuals to participate in the democratic decision-making process [109]. Voters can either vote directly or delegate their voting power to other experts. At the core of the system is a two-stage voting scheme inspired from choice architecture [110], which divides the entire voting process into two stages. This approach saves voters and experts' voting efforts and encourages thoughtful voting.

The system involves four types of actors (as described in Section 4.3, inspired from [11]): project owners who generate proposals, experts who review and vote on the proposals, voters who vote on the proposals, and voting committee members who compute and reveal the final results. The security model and design goals (Section 4.4) are analysed, encompassing privacy, fairness, flexibility, efficiency, and end-to-end verifiability [111, 112, 113, 114, 115].

The proposed system operates in iterative periods, each consisting of three epochs:

- Pre-Voting Epoch (Section 4.5): This phase handles proposal submission, innovation management, and registration of voters and experts.
- Voting Epoch (Section 4.6): In this stage, ballots are submitted using the Preferential Voting (Section 4.6.1) and Threshold Voting (Section 4.6.2) mechanisms, and the results are revealed.
- Post-Voting Epoch (Section 4.7): This epoch manages proposal execution, penalties, and rewards.

Each epoch consists of multiple rounds, and each round may take several blocks. Considering the potential length of the voting process, the system supports an Evolving Committee (Section 4.8), allowing voting committee members to be replaced while ensuring overall security.

The design of our decision-making model follows established practices within the blockchain governance research, improved for better system responsiveness and security. Our system architecture is heavily inspired by the Cardano [11] and Ethereum governance frameworks [67]. The phased approach-dividing the governance process into Pre-Voting, Voting, and Post-Voting epochs-is modelled after Cardano's structured governance process. Similarly, the voting mechanisms is informed by the community-centric voting systems observed in Ethereum's EIPs. These established models provide a proven blueprint for effective decentralised decision-making, offering a robust foundation for further innovation.

However, distinguishing our model from these precursors, we introduce the concept of an Evolving Committee (4.8), which allows for adaptive changes to the committee composition in response to evolving security needs and community dynamics. This feature addresses potential risks associated with fixed governance bodies and enhances the long-term sustainability of the system. Furthermore, the incorporation of multiple voting rounds within each epoch allows for a more granular and consensus-oriented approach to decision-making, which is crucial for aligning diverse stakeholder interests.

Moreover, the seamless integration of proposal execution and reward management in the Post-Voting Epoch reflects an advancement over traditional models by embedding accountability directly into the workflow, thereby incentivising high-quality contributions and ensuring that successful initiatives are promptly and effectively implemented.

By integrating insights from Cardano's and Ethereum's governance structures, this proposed model leverages the strengths of these platforms while addressing their limitations through innovative adaptations. The enhancements made in our system are designed to foster a resilient, adaptive, and transparent governance mechanism, ensuring it remains responsive to the needs and security concerns of the blockchain community. These refinements and innovations are pivotal in defining the next generation of blockchain governance, promising a more dynamic and robust framework for decentralised decision-making.

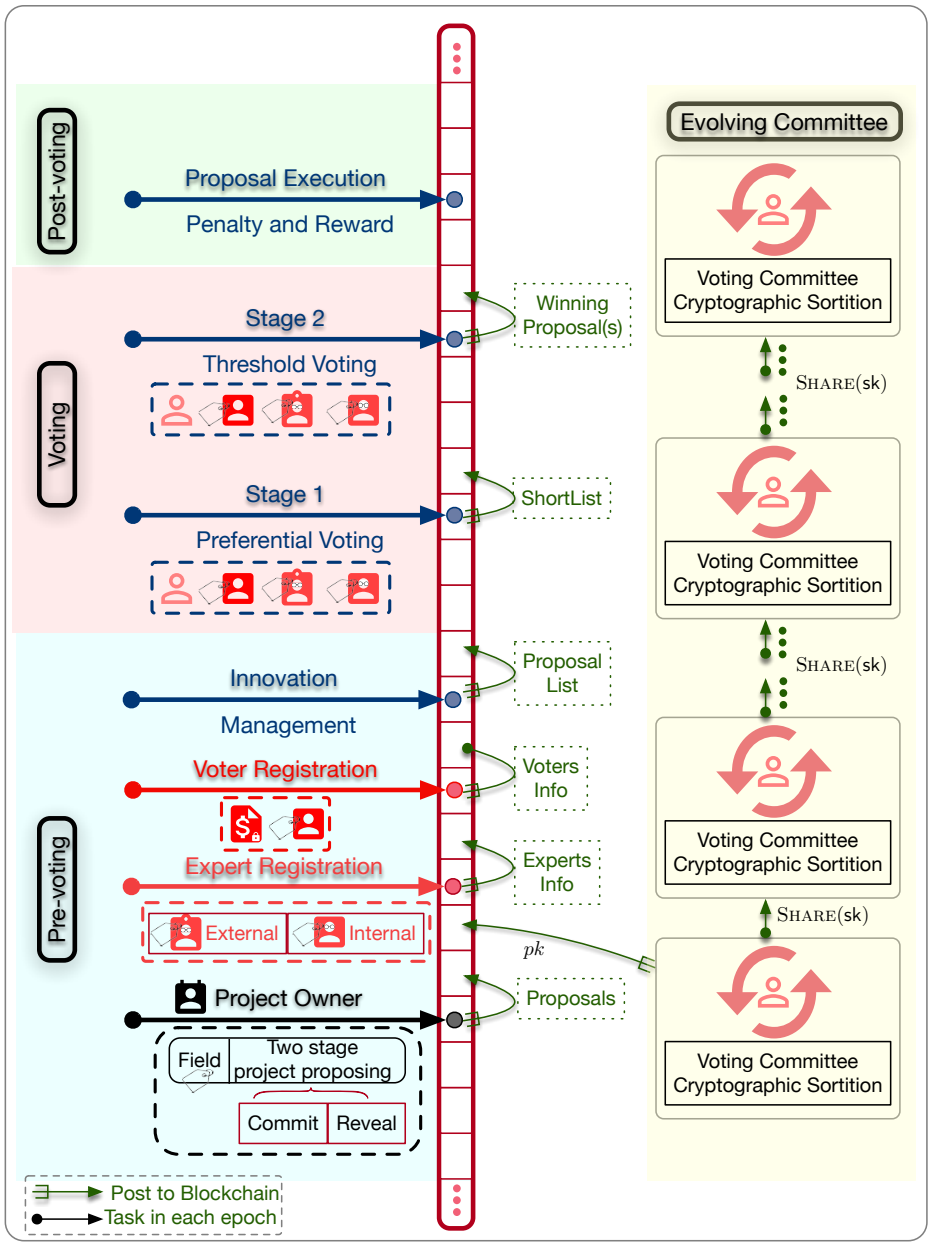

Figure 4.1: Systematic Design.

4.2 Participatory Budgeting

Participatory Budgeting is a fundamental concept within the proposed decision-making system, addressing how a limited budget should be allocated. The system supports participatory budgeting by categorising proposals, voters, and experts into different fields, each with its own budget allocation. In this thesis, we use blockchain development funding decision-making [11] as an example to explain the system, and the fields involved include:

- **Technology**: Covering the cost to build a developer ecosystem and support novel research, fostering a positive developer experience to incentivise adoption, productivity, and creativity.
- **Operation**: Covering the cost to establish open standards, interoperability, and cross-chain collaboration with projects based on other blockchains like Ethereum.
- **Products**: Covering the cost for quality Dapps and integration to increase utility.
- **Design**: Covering the cost for diversity and human creativity in the arts.
- **Market Expansion**: Covering the cost to explore research and scaling implementations to increase scalability.
- **Business**: Covering the cost to investigate Business-to-Business (B2B) and Business-to-Consumer (B2C) solutions to meet real business needs.
- **Foreign Growth**: Covering the cost to promote increased interaction with projects in foreign countries.
- **Customer Service**: Covering the cost to build an open-source development ecosystem, making core functionalities and community-owned open-source solutions available to everyone.
- **Finance**: Covering the cost for cryptocurrency-related tax policies and regulations.
- **Community**: Covering the cost for community-driven projects to increase meaningful participation among the community, including organising regular events, improving community advisor programs, and recruiting new members.
- **Miscellaneous Challenge**: Covering the cost for other ideas that focus on problems not covered by the aforementioned aspects.

These categories enable a more focused and efficient allocation of resources within the blockchain development funding decision-making process, fostering a more inclusive and impactful development of the blockchain platform.

4.3 Actors

Let v, e, c be integers in $\mathbb{Z}_q$. In field fld, one stake holder in each decision-making period might play one or more the following roles in each epoch i. These roles are inspired by existing governance models in blockchain systems, particularly drawing from the Cardano on-chain governance framework[11].

- **Proposal Owners**. Registered users who submit proposals seeking support are referred to as Proposal Owners. The winning proposal's owner receives a bonus reputation score for providing valuable and innovative proposals that meet essential field requirements. In the context of the blockchain development funding decision-making system, winning proposals receive funds at the end of the post-voting epoch. Similar to Cardano, where any participant can submit proposals, our system also encourages wide participation. However, unlike Cardano, which does not require proposal submitters to lock stakes, our system adds a layer of stake commitment to ensure that only serious proposals are put forward.

- **Voters**. A subset of stakeholders can register as voters by locking a certain amount of stakes on the blockchain, denoted by $\mathcal{V}_{\mathsf{fld}}^{[v]}$. Each voter, $\mathsf{V}_{\mathsf{fld}}^{(i)} \in \mathcal{V}_{\mathsf{fld}}^{[v]}$, can participate in voting within the field fld. The voting power of $\mathsf{V}_{\mathsf{fld}}^{(i)}$ is determined by the amount of stakes deposited during the registration phase. The consistency between a voter's decision and the final winning list, as well as their honesty during ballots submission, contribute to their reputation score and reward. This is a shift from Cardano's model where a minimum of 500 ADA allows voting without stake locking, thereby tailoring the voting power more significantly to stake size, which we posit increases voting seriousness and alignment with long-term project viability.

- **Experts**. Participants with high reputation and expertise can join the system to contribute their knowledge within field fld. There are two types of experts: Internal Experts and External Experts. Internal Experts are voters whose reputation scores exceed a predefined threshold, $\mathsf{T}_{\mathsf{fld}}$, while External Experts are highly recognised individuals invited to participate with a predefined initial reputation score. The responsibilities of experts include reviewing proposals and generating a proposal list during the pre-voting epoch, and voting on this list during the voting epoch. Both types of experts can only acquire voting power through delegation, meaning their ballots are only valid if some voters delegate their voting power to them. The consistency between an expert's decision and the final winning list, as well as their honesty during ballots submission, contribute to their reputation score and reward. Internal Experts are similar to Cardano's Community Advisors (CAs) but are distinguished by a mandatory reputation threshold. External Experts, akin to Veteran Community Advisors (vCAs) in Cardano, are recognised individuals invited based on a predefined reputation score.

Both types of experts review proposals and generate a proposal list during the pre-voting epoch, adding a structured layer of expertise review not explicitly defined in the Cardano system.

- **Voting Committee**. A subset of stakeholders selected by cryptographic sortition can join as Voting Committee members, denoted by $\mathcal{C}^{[c]}$. The probability of being selected is proportional to their locked stakes. Voting Committee members, denoted as $\mathsf{C}^{(t)}{}_{t\in[c]}$, collaboratively generate global key pairs used in voting and announce the final voting result. Honest voting committee members receive rewards at the end of each epoch. Voting Committee mirrors the delegated trust model seen in Cardano but incorporates explicit accountability mechanisms to address centralisation concerns noted in Cardano's selection of vCAs from CAs.

These roles ensure a diverse and inclusive participation in the decision-making process, promoting collaboration and expertise in the overall system.

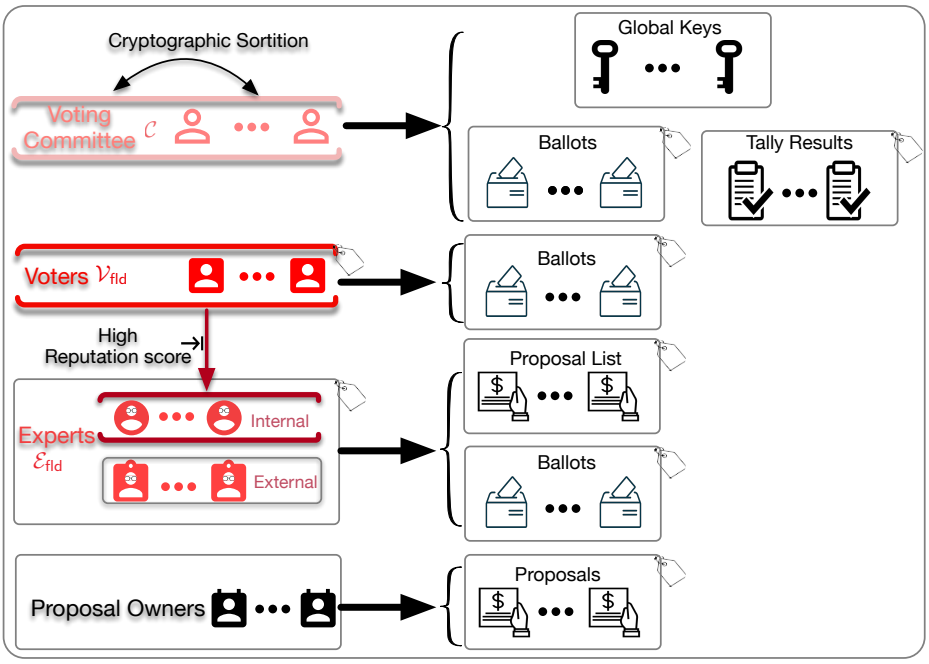

Figure 4.2: Actors.

4.4 Security Model and Design Goals

The proposed decision-making system involves four types of parties: project proposers, voters, experts, and voting committee members. Among them, proposers, voters, and experts

are not fully trusted, and their behaviours need to be verified within the system. Additionally, the voting committee members can have at most $1/2 \cdot c - 1$ compromised members, where c is the total number of voting committee members. The security analysis considers static corruption in the voting protocol, where the adversary decides which parties to corrupt before executing the protocol. It is important to note that the underlying blockchain infrastructure is assumed to be trusted.

The primary goal of this work is to design a privacy-preserving decision-making system over the blockchain. Specifically, the system aims to meet the following requirements, which have been discussed in previous literature ([111, 112, 113, 114, 115]):

- **Privacy.**

  The proposed system ensures the privacy of voter's voting preference, delegation choice, and expert's voting preference. Unless they choose to reveal it proactively, their preferences remain secret. The system prevents adversaries from deducing a voter or expert's preference unless they manipulate the voter or collude with other voters. The only exception to this is when all the voters come to a unanimous decision, in which case only the final tally results are revealed. In all other scenarios, individual preferences are kept confidential and cannot be inferred without active cooperation from the voters or experts themselves.

- **Fairness.**

  The proposed system ensures equal treatment for all participants regarding the receipt of information and involvement in a fair and neutral voting/proposing process. No proposers, voters, or experts have an advantage in revising their decisions based on peers' outputs or published results. The system maintains a level playing field where all participants have the same access to information and are involved in the decision-making process without any biases or preferences that could lead to unfair advantages. Transparency and openness are maintained throughout the entire process, promoting trust and confidence in the integrity of the decision-making system.

- **Flexibility and Efficiency.**

  To address the time constraints and improve efficiency, the proposed system allows voting committee members to work flexibly. The voting process is divided into multiple epochs and rounds, allowing committee members to participate and contribute during periods that are convenient for them. This flexibility ensures that committee members can effectively perform their roles without unnecessary time pressure. Furthermore, the system aims to minimise communication costs and enhance overall efficiency.

  By leveraging the blockchain infrastructure, communication and information sharing can be streamlined and optimised. The use of cryptographic techniques and privacy-preserving mechanisms helps reduce the amount of data that needs to be communicated

while ensuring the security and integrity of the voting process. In addition, the system employs cryptographic sortition for the selection of voting committee members, ensuring a fair and efficient process for forming the committee. This approach reduces the communication overhead associated with traditional voting committee selection methods. Overall, the combination of flexible working arrangements for committee members and efficient communication strategies helps to achieve a highly efficient and effective decision-making process within the proposed system.

- **End-to-end Verifiability.**

  The proposed system ensures end-to-end verifiability, which includes individual verifiability, universal verifiability, and eligibility verifiability.

  - **Individual Verifiability.**

    Individual verifiability allows voters and experts to verify the correctness of their own ballots. They can independently check whether their ballots have been published correctly and included in the final tally result. This feature enhances transparency and ensures that each participant's voting decisions are accurately reflected in the final outcome.

  - **Universal Verifiability.**

    Universal verifiability enables anyone to verify the fairness of the voting process and the correctness of the final tally result. This means that any external observer can independently validate that the voting process was conducted fairly and that the final result is accurate, without needing to trust the system or its operators.

  - **Eligibility Verifiability.**

    Eligibility verifiability ensures that only registered and valid parties can submit ballots or proposals. Each voter and expert can only submit one valid ballot. This verification process allows anyone to confirm that the final tally result is based on valid ballots from eligible voters and experts, preventing any manipulation or tampering.

  By providing end-to-end verifiability, the proposed system ensures transparency, integrity, and trustworthiness in the decision-making process, fostering confidence among all participants and external stakeholders.

4.5 Pre-Voting Epoch

The Pre-Voting epoch marks the beginning of each decision-making period and encompasses four main tasks: Proposal Registration, Expert Registration, Voter Registration, and Innovation Management.

- **Proposal Registration**.

  During the Proposal Registration phase, a series of proposal templates are released on the blockchain, each corresponding to a specific proposal field. These templates outline the scope and requirements for proposals in each field. Registered Proposal owners have the opportunity to submit their proposals for consideration in the decision-making process based on the chosen template, so that they can present their ideas, projects, or initiatives that require funding or support from the community. Each proposal is accompanied by relevant details, such as the scope, goals, and budget requirements, giving the voters and experts an opportunity to review and assess the proposal's potential.

  To ensure fairness and prevent potential issues like copying or imitating published proposals for ulterior motives, the system introduces a **Two Stage Project Proposing** procedure. This procedure involves the following steps:

  – Stage 1: **Commit.**

    During the Proposal Registration phase, proposal owners submit the commitment of their proposals by the given deadline. This commitment is then used to create the Proposal Commitment Transaction in the following form:

    $$\mathsf{Tx}(\{\mathsf{In}_i\}_{i=1}^{n}; \{\mathsf{Out}_j\}_{j=1}^{m}; (\mathsf{C}(P; r), \mathsf{addr})),$$

    Here, $\{\mathsf{In}_i\}_{i=1}^{n}$ and $\mathsf{Out}_j{}_{j=1}^{m}$ represent the input and output coins of the proposal owner, respectively, with the difference between them being the transaction fee. The Payload field of the transaction contains the commitment of the proposal, denoted as $\mathsf{C}(P; r)$, where P represents the proposal itself and r is a random value used for security purposes.

    The addr parameter in the transaction is the address used during the proposal execution phase. For instance, in the context of the blockchain development funding decision-making system, this address could be used for receiving funding if the proposal is approved.

    Simultaneously with the submission, a Preliminary Criteria-Checking process is conducted to rigorously analyse the metadata of the commitment transactions [116]. This scrutiny extends to examining behavioural patterns and origin details such as IP addresses, submission timestamps, and frequency of submissions from each sender's address. This early verification system leverages metadata analysis to identify and filter out potentially malicious or spam submissions before they can proceed. By setting a threshold for acceptable submission rates, the system effectively mitigates the risk of DDoS attacks and blocks actors attempting to flood the system with redundant proposals. Additional submissions exceeding this threshold are temporarily blocked or flagged for manual review.

By using this Proposal Commitment Transaction mechanism, proposal owners can securely and verifiably commit to their proposals, ensuring the integrity and validity of the decision-making process. This procedure is designed to promote transparency and fairness, as well as provide a clear and auditable record of proposal submissions.

– Stage 2: **Reveal**.

After the deadline, proposal owners proceed to open their commitments and reveal their proposals on the blockchain. This is done by creating the Proposal Reveal Transaction in the following form:

$$\mathsf{Tx}(\mathsf{In}_{i}{}_{i=1}^{n}; \mathsf{Out}_{j}{}_{j=1}^{m}; r).$$

Here, $\mathsf{In}_{i}{}_{i=1}^{n}$ and $\mathsf{Out}_{j}{}_{j=1}^{m}$ represent the input and output coins of the proposal owner, respectively, and r is the random value that was used in the commitment phase.

Following the initial metadata screening, once the proposals are disclosed, a second round of automated criteria-checking is initiated, which is now crucial as it focuses on the now fully visible content of the proposals. This phase employs advanced algorithms to detect anomalies in submission patterns that deviate from established norms, such as unusually high frequency or irregular timing, which may indicate a DDoS attack. The system dynamically adjusts these thresholds in real-time, based on observed network traffic and typical submission patterns, enhancing its ability to combat spam and repeated submissions. Through this rigorous verification process, the system ensures that only valid proposals are accepted for consideration in the subsequent voting period. Proposals found to be invalid, duplicates, or failing to meet the required standards are promptly rejected, thus preserving the integrity and enhancing the efficiency of the decision-making process.

After the proposals are disclosed, a second round of automated criteria-checking is initiated on the blockchain. This phase is crucial as it focuses on the content of the proposals, which is now fully visible. This rigorous process is designed to combat spam and prevent repeated submissions, as well as mitigate any potential DDOS attacks that target the revealed content. Through this verification, the system ensures that only valid proposals are accepted for consideration in the subsequent voting period. Proposals that are found to be invalid, duplicates, or fail to meet the required standards are promptly rejected, thus preserving the integrity and enhancing the efficiency of the decision-making process.

By using the Proposal Reveal Transaction and the subsequent automated criteria-checking, the system can ensure that only genuine and eligible proposals move

forward to the voting phase, promoting a fair and robust decision-making mechanism.

The Two Stage Project Proposing procedure ensures a fair and transparent opportunity for proposal owners to present their ideas, fostering collaboration and the generation of high-quality proposals that align with the decision-making process's goals. This procedure, combined with the Proposal Reveal Transaction and automated criteria-checking, enhances the overall integrity and efficiency of the decision-making process. It creates an inclusive and participatory environment, where stakeholders actively contribute their expertise, leading to well-informed and comprehensive decisions. The system's emphasis on transparency, collaboration, and fairness aims to establish a successful and sustainable governance model on the blockchain platform.

- **Voter Registration**.

  During the Voter Registration phase, stakeholders interested in voting can participate by locking some stakes on the blockchain to register as voters. They create a Voter Registration Transaction in the form of

$$\mathsf{Tx}(\mathsf{In}_i{}_{i=1}^{n}; \mathsf{Out}_j{}_{j=1}^{m}; (\mathsf{stk}_{k=1}^{t}, \mathsf{add}, \mathsf{fld})),$$

  where $\mathsf{stk}_{k=1}^{t}$ in the Payload field represents the locked stakes, and fld is the field tag. The voting power of a voter is determined by the amount of stakes they have locked, making it proportional to their locked stake in the system.

- **Expert Registration**.

  During the Expert Registration phase, well-known, highly regarded, and reputable individuals can be invited to register as external experts and are awarded an initial reputation score. They need to create an account on the blockchain to receive rewards and choose field tags to indicate their specific expertise. External experts create the External Expert Registration Transaction in the form of

$$\mathsf{Tx}(\{\mathsf{In}_i\}_{i=1}^{n}; \{\mathsf{Out}_j\}_{j=1}^{m}; (\{\mathsf{fld}, \ldots, \mathsf{fld}_n\}, \mathsf{add}, \mathsf{fld})).$$

  Note that an expert account can be assigned with more than one field tag to represent cross-disciplinary/domain expertise. However, the reputation score, penalty, and reward are updated separately in each field for participatory budgeting.

  In addition to external experts, voters whose reputation scores exceed a predefined threshold in a certain field can also register as internal experts during the pre-voting epoch. Internal experts create the Internal Expert Registration Transaction in the form of

$$\mathsf{Tx}(\{\mathsf{In}_i\}_{i=1}^{n}; \{\mathsf{Out}_j\}_{j=1}^{m}; (\{\mathsf{fld}, \ldots, \mathsf{fld}_n\}, \mathsf{Rep}_{\mathsf{fld}}, \mathsf{add}, \mathsf{fld})).$$

The voting power of both external and internal experts only comes from the voting power delegated to them by other voters. This is referred to as "Delegation Power". Being an expert allows a user to build social credit and reputation, giving them significant influence within the community even if they don't necessarily have a large stake.

- **Innovation management**.

  In the Innovation Management phase, experts review and scrutinise all the proposals that passed the pre-filtering stage. This phase serves as a soft filter to finalise the proposal list that will be eligible for voting in the voting epoch. Experts have the opportunity to discuss the proposals either online or offline, ensuring thorough evaluation.

  To ensure integrity, authentication, and system reliability, the proposal list is only considered valid if it is signed by a number of experts whose total reputation scores are more than 50% of all experts' reputation scores. This mechanism ensures that the proposal list has the support and approval of a significant portion of the expert community.

  Furthermore, experts can provide valuable feedback to project owners during this phase, improving their experience and helping them refine their proposals. Metadata is attached to every proposal by experts, which can be used to rank and order the proposals during the voting epoch.

  By involving experts in the innovation management process, the system benefits from their expertise and insights, creating a collaborative and knowledgeable community of decision-makers. This encourages experts to take their role seriously and provides proper incentives for their active and meaningful participation in the decision-making process.

4.6 Voting Epoch

The introduction of Two Stage Voting (TSV) in the proposed system is inspired by the concept of choice architecture, as discussed in [117]. TSV is designed to optimise the decision-making process by reducing the overall voting effort and encouraging thoughtful voting. In situations where the system has limited resources, such as funds, choice architecture can be applied to create a list of priorities in various fields. This helps to prioritise proposals and streamline the voting process, making it more efficient and effective.

On the other hand, without choice architecture, voters and experts would be required to vote on all proposals, which can be a time-consuming and challenging task. In such cases, it is more likely for voters and experts to fall into voting patterns that may not result in well-considered decisions. For example, some may vote for all proposals without careful

evaluation, while others may only vote for proposals they personally care about, neglecting the majority of the proposals. Moreover, there is a risk of random voting behaviour, where some participants vote for proposals at the top of the list without examining the later proposals.

By implementing Two Stage Voting, the proposed system encourages thoughtful voting and ensures that voters and experts have sufficient time and information to make informed decisions.

- **Preferential Voting**.

  In the first stage, known as the Preferential Voting stage, experts and voters review and evaluate proposals, creating a shortlist of priorities according to the predefined condition such as the total budget in each field and fund required by each proposal in the blockchain development funding decision-making system. This stage allows participants to rank proposals based on their preferences and priorities.

- **Threshold Voting**.

  In the second stage, known as the Threshold Voting stage, voters and experts only need to vote on this prioritised list of proposals. This reduces their voting effort and streamlines the decision-making process. Instead of voting on all proposals, participants can focus on the most important ones, which leads to more meaningful and well-informed voting decisions. A final winning list is generated in this stage according specific systematic requirements.

Figure 4.3 provides an example of plain-text ballots in the two stage voting scheme, assuming the honesty of all voters and experts. During the Preferential Voting stage, the ballot is represented as a list of three ranked unit vectors, signifying voters'/experts' preferences for the proposals. Each vector for the voters has a size equal to the number of proposals plus the number of experts, and only one element in the vector is set to 1 to represent the voter's choice, while the rest are set to 0. Similarly, each vector for the experts has a size equal to the number of proposals. These vectors are used to indicate the experts' preferences for the proposals. In this stage, voters can also choose to delegate their voting power to specific experts by setting the element corresponding to that expert's position in their vector to 1, while setting all other elements to 0.

For example, Voter $\mathsf{V}_f^{(1)}$ ranks Proposal $P_f^{(1)}$ as the first choice, followed by Proposal $P_f^{(3)}$ as the second choice, and Proposal $P_f^{(5)}$ as the third choice. They cannot select Proposal $P_f^{(1)}$ as both the first and second choice. Voter $\mathsf{V}_f^{(2)}$ delegates its voting power to expert $\mathsf{E}_f^{(1)}$, so $\mathsf{E}_f^{(1)}$ will have the same voting power as $\mathsf{V}_f^{(2)}$ and make choices on their behalf. Expert $\mathsf{E}_f^{(2)}$ ranks Proposal $P_f^{(1)}$ as the first choice, followed by Proposal $P_f^{(5)}$ as the second choice, and Proposal $P_f^{(3)}$ as the third choice. However, since no voters delegate their voting power to $\mathsf{E}_f^{(2)}$, $\mathsf{E}_f^{(2)}$'s ballots will not be considered in the tally computation.

After collecting all the ballots, the system will go through the ranked proposals and generate the final shortlist based on the available funds. The shortlist will include proposals with the highest ranks until the available funds are fully allocated. In the example provided, at the end of this stage, Proposal $P_f^{(3)}$ and Proposal $P_f^{(2)}$ will be shortlisted, assuming that the current funds can only support these two proposals.

During the Threshold Voting stage, voters and experts vote either YES, NO, or ABSTAIN for Proposal $P_f^{(3)}$ and Proposal $P_f^{(2)}$. Each ballot consists of two unit vectors, with each vector in the experts' ballots having a size of 3, and each vector in the voters' ballots having a size of 3 plus the number of experts. Voters can delegate their votes for each proposal to different experts. For example, Voter $\mathsf{V}_f^{(2)}$ delegates the voting decision for Proposal $P_f^{(1)}$ to $\mathsf{E}_f^{(2)}$, and the voting decision for Proposal $P_f^{(2)}$ to $\mathsf{E}_f^{(1)}$. In this example, the final tally result is computed based on the votes, and it is evident that Proposal $P_f^{(3)}$ receives the most YES votes.

By allowing voters to delegate their votes to experts and combining the votes from both voters and experts, the system ensures that each proposal is evaluated thoroughly and fairly. This approach leverages the expertise of the experts while preserving the collective decision-making power of the voters. The final tally result represents the consensus of the community, reflecting the preferences and priorities of the participants in the decision-making process.

The Two Stage Voting mechanism helps to streamline the decision-making process and encourage more thoughtful voting. By allowing voters and experts to rank proposals in the Preferential Voting stage and only vote on the shortlisted proposals in the Threshold Voting stage, the system promotes more meaningful participation and ensures that the most important proposals are given priority based on the available resources. By using Two Stage Voting, the system enhances the overall quality and fairness of the decision-making process. It allows participants to dedicate more attention and scrutiny to the proposals that matter the most, and it encourages active engagement in the decision-making process. This mechanism not only increases the efficiency of the decision-making process but also empowers participants to have a more significant impact on the final outcomes.

Furthermore, the implementation of Two Stage Voting fosters a collaborative environment where stakeholders can actively contribute their expertise and preferences, resulting in better-informed and comprehensive decisions. It promotes inclusivity and participation, as voters and experts are more likely to engage when they have a more manageable and meaningful voting task.

In summary, Two Stage Voting is a valuable addition to the decision-making system, enhancing its effectiveness and supporting a more democratic and participatory decision-making process over the blockchain platform. The Preferential Voting stage allows for a curated list of priorities, while the Threshold Voting stage streamlines the final voting process, resulting in a more efficient and well-informed decision-making process.

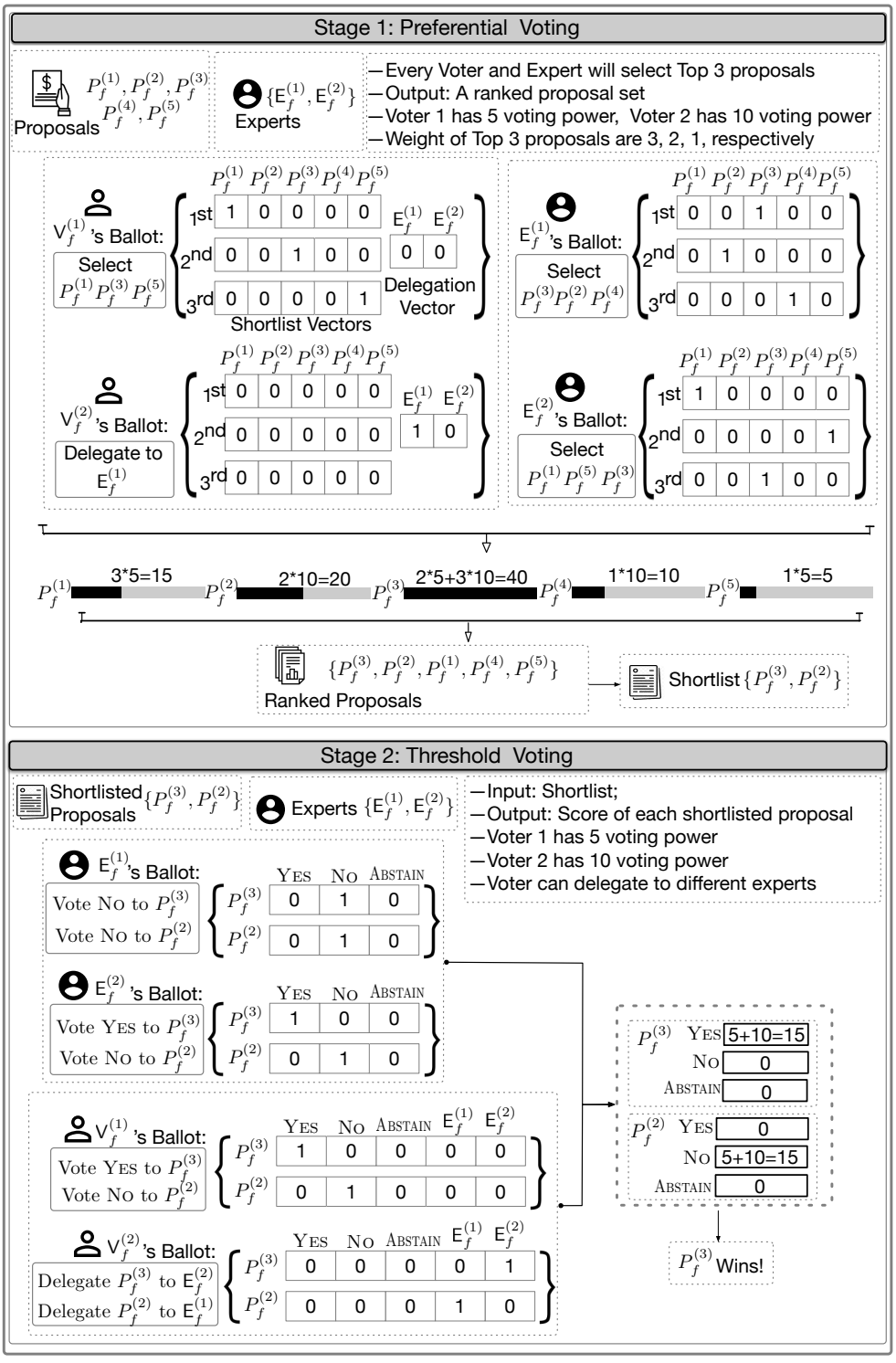

Figure 4.3: Plain-text Ballots example in two stage voting scheme (Assume all voters and experts are honest).

4.6.1 Preferential Voting

In the preferential voting stage, voters and experts cast their votes on the proposal list generated during the innovation management phase. The voting mechanism used is Borda Count Voting, a widely recognised method in social choice theory ([118, 119]).

In Borda Count Voting, each voter and expert ranks the proposals in order of their preference. The top-ranked proposal receives a certain number of points, the second-ranked proposal receives slightly fewer points, and so on, until the last-ranked proposal receives only one point. The points are then summed for each proposal, and the proposal with the highest total points is considered the most preferred by the community.

Based on the results of the Borda Count Voting and other criteria such as the funds requested by each proposal and the total available funds in the corresponding field, a shortlist of proposals is generated. This shortlist includes the proposals that are most favored by the community and meet the specified conditions for funding. These shortlisted proposals proceed to the next stage of the decision-making process, the Threshold Voting stage, where voters and experts further vote to select the final winning proposals.

The Preferential Voting stage in the proposed decision-making system consists of the following tasks:

- **Voter Cast**.

  Voter can either select its own shortlist or delegate its voting power to **one** expert, then submit encrypted ballot to blockchain;

- **Expert Vote**.

  Expert select its own shortlist and submit encrypted ballots to blockchain. Note that expert's ballot is only valid if it has delegation power from voter(s);

- **Tally**.

  Voting committee members jointly compute and reveal the tally results; Proposals are ranked according to the scores. Taking the blockchain development funding decision-making system as an example, top ranked proposals are shortlisted until the total fund is exhausted.

In the Preferential Voting stage, the size of the proposal list is denoted by n, and a fixed parameter $s \leq n$ is predefined for each epoch. Both voters and experts are required to choose their top s ordered preference priorities from the proposal list. These priorities can be based on various factors, such as the funding asked by each proposal, the quality of the proposals, and other relevant criteria.

After selecting their preferences, voters and experts use the public key generated by the voting committee to encrypt their ballots. These encrypted ballots are then posted to the blockchain, ensuring the privacy and security of their voting decisions.

In support of the principles of delegative democracy, an operational framework is introduced wherein voters possess the prerogative to allocate their voting authority to recognised experts based on the experts' proclivities, professional backgrounds, and historical voting patterns. This mechanism of delegation affords voters the opportunity to vest their voting determinations in experts whom they hold in high regard, and who they anticipate will make judicious choices on their behalf.

Two primary rationales underlie the incentivisation for voters to avail themselves of the delegation of voting power. Primarily, the number of experts typically remains limited in contrast to the volume of proposals or projects, thereby rendering the delegation process more feasible relative to the arduous task of individually scrutinising and evaluating each proposal. Secondarily, the realm of experts tends to exhibit a lesser degree of turnover when compared to the domain of proposals. This inherent stability within the expert cohort facilitates voters' ease in delegating to familiar and dependable experts, thereby engendering a sense of dependability within the electoral procedure.

The envisaged system, through the accommodation of delegation and the cultivation of well-considered voting practices, aspires to streamline the decision-making paradigm and nurture a milieu characterised by heightened efficiency and inclusiveness for all participants. In this construct, voters are afforded the means to actively partake and wield substantive influence over ultimate determinations, even in instances where their temporal availability or specialised knowledge for individual proposal assessments is limited. Conversely, experts are bestowed with the capacity to channel their specialised domain acumen and insights, thereby ensuring a decision-making process imbued with enhanced sagacity and comprehensive evaluation.

As shown in Figure 4.3-(a), voters generate s shortlist vectors with size n. In each shortlist vector, the respective priority is set as 1 in the vector, the rest of the vector are set to 0. For example, $\mathsf{V}_f^{(1)}$'s priorities are $P_f^{(1)}, P_f^{(3)}, P_f^{(5)}$, the first shortlist vector is $\{1, 0, 0, 0, 0\}$ as a unit vector. Besides the shortlist vectors, voters need to generate a delegation vector with size e, which is the number of experts in this epoch. For example, $\mathsf{V}_f^{(1)}$ chooses to vote by itself, so its delegation vector is all zero. $\mathsf{V}_f^{(2)}$ delegates to $\mathsf{E}_f^{(1)}$, its delegation vector is $\{1, 0\}$, and shortlist vectors are all zero. Experts only need to generate shortlist vectors, which are similar to voters' shortlist vectors.

In the tally task, the voting committee plays a crucial role in jointly computing and revealing the ranked proposals. Each proposal in the ballot is assigned ranking points based on its position in the voter's or expert's preferences. The top proposal receives s ranking points, the second gets $s - 1$ ranking points, and so on until the s-th proposal, which gets 1 ranking point. The score of each proposal is then computed by multiplying the ranking points by the corresponding voting power or delegation power of the voter or expert.

Based on the funding asked by each proposal and the total available funding in the field, a set of shortlisted proposals is automatically generated. These shortlisted proposals will proceed to the Threshold Voting stage for final decision-making. The shortlist not only helps

the decision-makers to know the highest priorities and allocate the funds effectively but also provides valuable guidance to the community, steering the use of funds towards the most critical and impactful projects.

Moreover, within the framework under consideration, the balloting choices of experts are openly disclosed on the blockchain. This overt transparency affords fellow voters the opportunity to peruse and authenticate the historical trajectory of experts' voting preferences. This provision equips voters with a more comprehensive basis upon which to judiciously delegate their voting authority to these experts.

The public revelation of experts' historical voting patterns serves to fortify the principles of responsibility and engenders a prevailing sentiment of reliance within the community. This transparency acts as a mechanism for holding experts accountable for their decisions, and in tandem, encourages a heightened degree of trust among the members of the community.

4.6.2 Threshold Voting

During this particular phase, both the electorate and the experts partake in casting their votes with the options of YES, NO, and ABSTAIN in relation to each proposition featured within the shortlist. Upon the culmination of this stage, the definitive assortment of proposals that attain eligibility for funding is ascertained through an automated process.

Threshold Voting encompasses the following tasks:

- **Voter Cast**.

  For each proposal in the shortlist, voter can vote directly, or delegate its voting power to an expert [1], who makes decision about this proposal, then every voter submits encrypted ballot to blockchain;

- **Expert Vote**.

  Expert votes on the shortlist, its ballot only be valid if it has delegation power from voter(s). Experts' ballots will be made publicly at the end of voting epoch;

- **Tally**.

  In the tally task, the voting committee members work together to perform several critical tasks. First, they compute the delegation power of each expert. Delegation power represents the total voting power delegated to each expert by the voters who chose to delegate their voting authority. This computation takes into account the voting power of each delegating voter and the preferences of experts they delegated to.

  Next, the voting committee computes the tally results for each proposal in the shortlist. The tally results include the number of YES votes, NO votes, and ABSTAIN votes for

[1] In Threshold Voting, a voter can delegate each proposal to different experts; in Preferential Voting, voter can only delegate to one expert. Delegation ballots can be found in Figure 4.3 made by voter $\mathsf{V}_f^{(2)}$.

each proposal. These results are computed based on the encrypted ballots submitted by voters and experts during the voting phase.

Once the tally results are computed, the voting committee reveals the results to the public. Transparency in the tally results ensures that the decision-making process is fair and accountable. The public can verify the accuracy of the tally results and ensure that the decisions are made in the best interest of the community.

After revealing the tally results, the proposals are ranked according to their scores. Proposals that received at least 10% of the positive difference between YES and NO votes are considered for funding. The top-ranked proposals, based on their scores, are the ones that will be funded.

Denote the size of shortlist in this stage by s, as shown in Figure 4.3-(b), experts need to generate s vectors with size 3, standing for YES, NO, and ABSTAIN options. For each position, 1 represents to choose this option, and 0 otherwise. For example, $\mathsf{E}_f^{(1)}$ chooses to vote "NO to $P_f^{(3)}$, then the vector for $P_f^{(3)}$ is $\{0, 1, 0\}$. Voters need to generate s vectors with size $3 + e$, standing for YES, NO, and ABSTAIN options and delegation to experts. For example, $\mathsf{V}_f^{(2)}$ delegate the choice for $P_f^{(3)}$ to $\mathsf{E}_f^{(2)}$, this vector is $\{0, 0, 0, 0, 1\}$.

This ranking and funding process ensures that the most supported and promising proposals, as determined by the community through their votes, are prioritised for funding. It allows the decision-making process to be more democratic and representative, as proposals with substantial support are more likely to receive funding. Overall, this approach promotes fairness, efficiency, and effectiveness in the allocation of resources and decision-making within the proposed system.

4.7 Post-Voting Epoch

In the post-voting epoch, the winning proposal(s) that received sufficient support in the Threshold Voting stage are eligible for funding. The allocated funds are then distributed to the respective project owners to execute their proposed initiatives or projects.

At the same time, a certain proportion of the fund is used to reward participants who made correct and honest decisions during the decision-making process. This includes both voters and experts who provided valuable inputs and voted in alignment with the final winning proposals. By rewarding participants for their constructive contributions, the system incentivises active engagement and thoughtful decision-making.

Honest voting committee members, who carried out their duties diligently and honestly during the voting process, also receive a fixed amount of reward for their efforts. This ensures that the voting committee members are motivated to act in the best interest of the community and uphold the integrity of the decision-making process.

In addition to the rewards, participants' reputation scores are updated based on their behaviours throughout the decision-making process. This reputation system serves as a mechanism to assess the trustworthiness and reliability of participants. Participants who consistently make correct decisions and contribute positively to the community will experience an increase in their reputation scores, granting them more influence and voting power in future decision-making events. This reputation system serves as an effective incentive for participants to actively engage in the decision-making process, fostering a culture of responsibility and accountability within the community.

Conversely, participants who fail to submit valid ballots or engage in malicious behaviours may face penalties, such as the loss of deposited stakes, as a deterrent against dishonest and irresponsible actions. This approach ensures that the decision-making system remains fair, transparent, and driven by participants who demonstrate a genuine commitment to the community's interests.

Overall, the post-voting epoch plays a crucial role in not only funding the winning proposals but also rewarding and updating the reputation of participants. It helps maintain the integrity of the decision-making system, encourage active participation, and ensure that decisions are made in a fair, transparent, and accountable manner.

4.8 Evolving Committee

As previously discussed, the proposed system addresses the challenge of long voting epochs by introducing the concept of an evolving committee to periodically replace the voting committee. This mechanism ensures that the voting committee members do not need to be online and hold global key pairs for extended periods, which could be cumbersome and impractical.

In each round, stakeholders interested in joining the voting committee demonstrate their honesty and commitment by locking a certain amount of stakes on the blockchain. Subsequently, these stakeholders privately perform a cryptographic sortition process to determine the composition of the voting committee for that round. The probability of being selected to be part of the committee is directly proportional to the number of stakes they have locked. After successfully winning the cryptographic sortition, the selected stakeholders publicly post the sortition proof on the blockchain, confirming their identity as members of the voting committee for that round.

The process begins with the genesis voting committee in the first round, which collectively runs a Distributed Key Generation (DKG) protocol to generate global key pairs. These global key pairs consist of a global public key and a global secret key, which play a crucial role in the secure voting process. In subsequent rounds, the voting committee re-shares the global secret key to the next voting committee, ensuring a seamless transition of responsibilities and maintaining the security and continuity of the decision-making process.

At the conclusion of the voting epoch, the voting committee undertakes critical tasks, such as Delegation Calculation and Tally Calculation, with the utmost precision and efficiency to compute the final tally results. These tasks are essential for determining the outcome of the decision-making process and ensuring the integrity and fairness of the final results. The voting committee's meticulous execution of these calculations is vital for the successful conclusion of the decision-making process over the blockchain platform. These tasks are pivotal in determining the final results, maintaining the integrity of the process, and ensuring fairness and transparency. The voting committee's diligence in carrying out these computations is fundamental to the overall effectiveness and reliability of the decision-making system.

As previously outlined in Section 4.3, the voting committee is selected through cryptographic sortition, drawing inspiration from the proposer selection methods detailed in works such as [19, 120]. This approach is similarly employed to select nodes in related research [121, 122, 123, 124, 125], ensuring the process is publicly verifiable and yields a randomly and independently selected voting committee in each round. Committee members are required to lock stakes to participate in this self-election process, with the understanding that dishonest behavior will lead to the forfeiture of their stakes at the end of the decision-making period.

To ensure robust security measures and effectively manage potential adversarial threats within the committee, a comprehensive analysis of the security implications is conducted. This includes assessments of adversarial capabilities and the integrity of the voting process. Detailed numerical examples and committee configurations under various adversarial conditions are extensively explored in Section 7.2.3 of this thesis. These analyses are crucial for determining optimal committee sizes and decision-making thresholds necessary to mitigate risks from potential adversarial actions.

The security of the decision-making system, remains highly secure against adversarial control. Given the overwhelming majority of voting committee members are projected to be honest in each round, both the privacy of ballots and the overall protocol's integrity are effectively safeguarded.

Furthermore, to ensure the long-term integrity and accountability of the voting committee, members found to be cheating face severe penalties, including the forfeiture of all deposited stakes and a permanent ban from future participation in the decision-making system. These stringent measures are crucial for maintaining the trust and efficacy of the governance process, reinforcing that the system is underpinned not only by technological safeguards but also by a framework of enforceable accountability measures.

The introduction of the evolving committee not only alleviates the burden on voting committee members but also enhances the overall efficiency and security of the decision-making system. This approach enables the system to adapt to changing conditions and ensures a consistent and reliable governance model over the blockchain platform. By employing cryptographic techniques and incentivising stakeholder participation, the

proposed system fosters a robust and democratic decision-making process that can accommodate diverse stakeholders and address various challenges effectively.

4.9 Summary

In this chapter, we have provided an overview of the proposed privacy-preserving decision-making system over blockchain. The system is designed to support participatory budgeting and encourage collaboration and inclusivity in the decision-making process. We have introduced the key actors in the system, including proposal owners, voters, experts, and the voting committee. The system operates in iterative epochs, each comprising three stages: Proposal Registration, Voter Registration, Expert Registration, and Innovation Management in the Pre-Voting Epoch; Preferential Voting and Threshold Voting in the Voting Epoch; and Funding and Reward Distribution in the Post-Voting Epoch.

We have highlighted the importance of the Two Stage Voting Scheme in reducing voting effort and promoting thoughtful voting decisions. In the Preferential Voting stage, voters and experts rank proposals based on their preferences, creating a curated list of priorities. In the Threshold Voting stage, they only need to vote on this prioritised list, streamlining the decision-making process and focusing on the most important proposals.

Moreover, we have discussed the Evolving Committee functionality, allowing the replacement of the voting committee in each epoch, ensuring the flexibility and efficiency of the system. The voting committee is responsible for tasks such as Distributed Key Generation, Tally Calculation, and Delegation Calculation, which are essential for the accuracy and fairness of the decision-making process.

In the subsequent chapters, we will delve into the technical aspects of the proposed system. Chapter 5 will present the Distributed Batch Key Generation functionality and a low-complexity protocol to generate multiple keys simultaneously for the voting committee. Chapter 6 will describe the Two Stage Voting Scheme in detail, including voting functionalities and protocols in the voting epoch. Chapter 7 will focus on the Evolving Protocol, which enables the seamless replacement of the voting committee. Lastly, Chapter 8 will discuss the incentives, such as reputation management, to motivate active participation and cooperation among proposal owners, voters, and experts in the decision-making process. Together, these building blocks form a comprehensive and robust privacy-preserving decision-making system over the blockchain platform.

Chapter 5

Building Block: Distributed Key Generation

> It used to be expensive to make things public and cheap to make them private. Now it's expensive to make things private and cheap to make them public.
>
> Clay Shirky

5.1 Overview

As Internet brings us together and frees us from the geographic fetters, governments, organisations and individuals become victims of malicious behaviours over the Internet. As a consequence, these malicious behaviours threaten confidentiality, security, and trust in online interactions. Issues such as data breaches and scams have become prevalent due to the allure of economic gains and other hidden motives behind malicious activities. Despite efforts to control these behaviours, they continue to proliferate, posing significant challenges to ensuring the integrity and reliability of online systems.

In response to these challenges, one approach might be to delegate confidential computations to a trusted third party, relying on this entity to compute and provide the final results. However, constructing an absolutely trusted authority over the internet is not a straightforward task. Even if such an authority were established, there would always be the risk of manipulation or compromise, leading to a single point of failure and potential collapse of the entire system.

Instead of relying on a trusted third party, threshold cryptosystem ([126]) guarantees the security of the whole system by fault tolerance. Distributed Key Generation (DKG) enables

participation nodes to jointly generate secrets (global public keys and global secret keys) in a private, secure and distributed way. In DKG, the secret is generated by all participants, and there is no participant in sole possession of it. In the Sharing phase, every participant uses secret sharing to generate a secret, the global secret is reconstructed by adding all these shares. Any sufficient large set of the participants can reconstruct the secret, while smaller subset can not derive any additional information except the public information.

DKG protocol plays a fundamental and versatile role in numerous cryptographic protocols and applications, serving as the foundation of threshold cryptosystems, enabling secure and distributed operations in different scenarios. One significant application of DKG is in identity-based encryption schemes [127, 128]. It is used to generate distributed private keys for clients, addressing the key escrow issue and enhancing the security of the encryption process. In public key cryptosystems, DKG enables threshold public key decryption [129]. In this context, each participant holds a partial secret key, and a subset of participants can collaboratively decrypt cipher-texts encrypted with the global public key, enhancing the resilience and privacy of the decryption process.

Moreover, DKG supports threshold signatures [130, 131, 132, 133], a crucial feature in public key cryptosystems. In threshold signature schemes, each participant holds a partial signing key, and a subset of participants can collectively sign messages by combining their partial signatures. This ensures that no single participant can produce a valid signature, enhancing security and enabling secure and distributed signing.

In addition to these applications, DKG has been employed in randomness beacons [134], Byzantine consensus protocols [135, 136, 137], time-stamping services [138, 139], data archive systems [140, 141], and distributed coin tossing protocols [142, 143, 144]. In each of these applications, DKG enables secure and distributed operations, ensuring the integrity and trustworthiness of critical cryptographic operations.

In summary, DKG serves as a fundamental building block for various cryptographic protocols, providing a reliable and secure mechanism for distributed and threshold operations. Its versatility and wide range of applications make it a crucial component in the design and implementation of secure and trustless systems across different domains.

Since Pedersen's pioneering work on DKG protocol [145], researchers have explored and extended DKG in various directions. Several improvements have been proposed to enhance the security and efficiency of DKG protocols.

Gennaro et al. [146, 38] addressed the issue of adversaries altering the distribution of generated keys in variants of Pedersen's DKG protocol. Canetti et al. [147] introduced techniques to achieve adaptive security in DKG. Further advancements were made by Kate et al. [148], who proposed asynchronous DKG protocols based on bivariate polynomials. The communication cost of DKG has also been a subject of investigation, and techniques to reduce it have been proposed by Canny and Shrimpton [149] and Neji et al. [150].

Despite these breakthroughs in DKG protocols, the communication overload of DKG remains a challenge, particularly in the context of large-scale key generation. In Table

5.1, we compare different DKG protocols in terms of network model, communication channel, computation complexity, communication complexity, and threshold. Most DKG protocols, such as [147, 151, 38, 150, 152, 153, 154], require participants to broadcast $\mathcal{O}(n)$-sized messages to generate a global key pair, resulting in $\mathcal{O}(n^2)$ communication complexity. Zhang et al. [155] achieved DKG with $\mathcal{O}(n)$ communication complexity based on ciphertext-policy attribute-based encryption (CP-ABE). Gurkan et al. [156] achieved a communication complexity of $\mathcal{O}(n \log n)$ with Byzantine nodes of $\log n$. In asynchronous and semi-synchronous DKG, current protocols ([157, 133, 158, 159, 160]) can tolerate $1/3$ faulty nodes with communication complexity between $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$.

To address the challenge of communication complexity in DKG, we propose a novel *Distributed Batch Key Generation (DBKG)* method in this chapter.In the DBKG protocol $\Pi^{n,\mu,m}_{\text{DBKG}}$ when generating $\mathbf{N}$ keys, where $\mathbf{N} \subset [n]$, n is the number of participants, every party needs to compute and send $\mathcal{O}(n)$ cipher-texts and commitments. The Correct Sharing NIZK has $\mathcal{O}(n)$ computation and communication cost for proving and verification. Therefore, the overall communication cost for DBKG protocol is amortised $\mathcal{O}(n)$.

In the initialisation and commitment generation phase, each participant selects $t + 1$ coefficients to construct a polynomial and generates corresponding commitments for each coefficient. Given that t is less than half of n, the computational cost for this step is $\mathcal{O}(t)$, which is typically simplified to $\mathcal{O}(n)$ when t is proportional to n.

During the distribution and encryption stage, each participant is responsible for encrypting a share for every other participant. This entails performing n encryption operations, where each operation is assumed to have a constant time complexity, thus aggregating to a total complexity of $\mathcal{O}(n)$. The next crucial step involves the generation of NIZK proof to assert the correctness of the encrypted shares. The creation of these proofs, involving polynomial operations and cryptographic proof generation, maintains a complexity of $\mathcal{O}(n)$.

Verification and complaint handling are essential components of the protocol. Each participant verifies NIZK proofs submitted by all others, with this verification process typically scaling linearly with the number of participants, leading to a complexity of $\mathcal{O}(n)$. Should there be any complaints, given that they are expected from up to t dishonest participants, and $t < \frac{n}{2}$, the complaint handling by each honest participant would also follow a complexity of $\mathcal{O}(t)$, which is representable as $\mathcal{O}(n)$ but is inherently less burdensome than initially anticipated.

The final key compilation phase requires each participant to combine contributions from all others to compute the final public key, involving n multiplicative operations and hence resulting in a complexity of $\mathcal{O}(n)$.

Therefore the HIM-based DBKG Protocol efficiently manages an $\mathcal{O}(n)$ computational complexity per participant across all major operational phases. This system's structured approach not only emphasises the protocol's robustness but also highlights its suitability for practical implementation in large-scale environments demanding high security and efficient

decentralised key generation processes.

Additionally, in our analysis of the HIM-based DBKG Protocol, the Hyper-Invertible Matrix (HIM) is utilised as a predefined and static component within the cryptographic framework. Since the HIM is established prior to the execution of the protocol and remains unchanged across its applications, its computation or generation does not recurrently impact the operational complexity of each protocol instance. Instead, the HIM serves as a constant cryptographic tool that participants access and use for specific operations within the protocol, such as facilitating transformations or other matrix-based computations essential for the protocol's functionality. As a result, the computational overhead associated with generating the HIM is considered a one-time setup cost, external to the routine execution costs of the protocol. This setup is assumed to be completed before the protocol's active phases begin, thereby excluding it from the per-execution computational complexity analysis.

Our approach differs from [155] as we use Elgamal encryption in the protocol and analyse its security under the UC framework. Leveraging the additively homomorphic property of Elgamal encryption, our proposed DKG improves the efficiency of applying distributed keys. In Chapter 6, we explain how to efficiently encrypt ballots and reveal the final tally results in the voting scheme based on the keys generated in DBKG.

The remaining sections of this chapter are structured as follows:

- Section 5.2 provides a comprehensive review of the state-of-the-art in DKG protocols, covering both theoretical and practical aspects.
- In Section 5.3, we introduce a new Distributed Batch Key Generation (DBKG) ideal functionality under the Universal Composability (UC) framework.
- Section 5.4 presents our proposed DBKG protocol, which *UC-realises* DBKG functionality from Section 5.3. The protocol is based on Hyper-Invertible Matrix (HIM) [25] and achieves a computation complexity of $\mathcal{O}(n)$ and a communication complexity of $\mathcal{O}(n)$.
- In Section 5.5, we discuss the Zero-Knowledge proofs utilised in the DBKG protocol.
- Section 5.6 analyses the security of the HIM-based DBKG and establishes its security properties within the UC framework.

Table 5.1: Comparison of DKG Protocols

Protocol	**Network**	**Comp.**	**Comm.**	**Threshold**
[155]	Sync.	$\mathcal{O}(n)$	$\mathcal{O}(n)$	1/2
[156]	Sync.	$\mathcal{O}(n\log^2 n)$	$\mathcal{O}(n\log n)$	$\log n$
[154]	Sync.	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	1/2
[157]	Async.	$\mathcal{O}(n^2)$	$\mathcal{O}(n^3)$	1/3
[133]	Async.	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	1/3
[158]	Async.	$\mathcal{O}(n^2)$	$\mathcal{O}(n^3\log n)$	1/3
[159]	Async.	$\mathcal{O}(n^2)$	$\mathcal{O}(n^3)$	1/3
[153]	Sync.	$\mathcal{O}(n\log n)$	$\mathcal{O}(n^2)$	1/2
[152]	Sync.	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	1/2
[150]	Partial Sync.	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	1/2
[160]	Partial Sync.	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	1/3
[38]	Sync.	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	1/2
[151]	Sync.	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	1/2
[147]	Sync.	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	1/2
HIM DBKG (Section 5.4)	Sync.	$\mathcal{O}(n)$	amortised $\mathcal{O}(n)$	1/2

- n is the number of participants in DKG;
- **Network** is either asynchronous or synchronous in the DKG protocols. In the asynchronous network, transmission data can be delayed by adversary, but all the messages sent between honest participants must be delivered eventually. Ideal functionalities of asynchronous communication and synchronous communication can be found in [20];
- **Comp.** and **Comm.** are the computation complexity and communication complexity of the DKG protocols for all participants;
- **Threshold** is the tolerance of corruption of the DKG protocols.

5.2 State of the Art

Pedersen proposed the JF-DKG protocol, which is the first Distributed Key Generation (DKG) protocol in the threshold cryptosystem. It is based on N parallel invocations of Feldman's Verifiable Secret Sharing [161]. The JF-DKG protocol has been utilised in various cryptographic protocols, including the digital signature standard [162], over the following decades.

However, subsequent research conducted by Gennaro *et al*.in their works [146, 38] uncovered a notable flaw in the key generation process of the JF-DKG (Joint-Feldman Distributed Key Generation) protocol. They revealed that the generated keys within this protocol did not conform to a uniformly random distribution. Their findings demonstrated the potential for malicious actors to exploit this vulnerability by manipulating two participants. Through this manipulation, they were able to introduce bias into the final bit of the publicly generated key, thereby affording adversaries a significant advantage.

To address and rectify these identified vulnerabilities, they introduced an alternative DKG protocol. This novel approach was grounded in the principles of Pedersen's Verifiable Secret Sharing (VSS) and Feldman's VSS, both of which contribute to enhancing the security and randomness properties of the protocol. Under this newly proposed protocol, participants are obligated to construct a consistent set of qualified participants, referred to as the "qualified participants set", during the Sharing phase. This set is standardised across all honest participants. Subsequently, during the Reconstruction phase, the collective public key is computed utilising this established set.

The novel DKG protocol, featuring these improvements, has been effectively integrated into various emerging threshold cryptosystems, as evidenced in the works by Duan et al. [163] and Herranz et al. [164]. These contributions collectively underscore the progressive refinement of cryptographic systems, with heightened emphasis on security, robustness, and the rectification of vulnerabilities.

Nevertheless, Gennaro *et al*.'s DKG protocol requires a private channel to send shares while keeping them hidden from dishonest participants. This approach comes with three drawbacks:

- Identifying corrupted users becomes challenging when a private channel is used. Both the sender and receiver of each exchanging message can be dishonest about the authenticity of the message. As a result, determining whether a dishonest sender sends an invalid share or a dishonest receiver falsely claims to have received an invalid share becomes difficult.

- If the consistency check of a share fails, an additional round is required to address the complaint. During this extra round, participants need to reveal their secret shares to resolve the complaints, leading to increased complexity.

- In practice, establishing and maintaining a private channel is often a challenging task, adding further practical difficulties to the protocol's implementation.

The drawbacks associated with Gennaro *et al.*'s DKG protocol significantly limit its practical usability. To address some of these issues, [147] proposed improvements that achieve adaptive security. Despite this improvement, similar to the works of [165, 166, 167], these methods still face inherent practical challenges as they rely on Gennaro *et al.*'s DKG protocol as a crucial component in their threshold cryptosystems.

To overcome the challenges associated with private channels, [151] proposed a one-round DKG protocol that utilises a public channel instead. In their approach, Publicly Verifiable Secret Sharing (PVSS) is employed to distribute partial secret keys, allowing everyone to verify the correctness of the shares received by the recipients. The protocol achieves this by utilising the Paillier cryptosystem to encrypt the secret shares, and each encryption is accompanied by a non-interactive zero-knowledge proof (NIZK) for public verification.

While this protocol offers advantages in terms of utilising a public channel and enabling verifiability, it introduces key management complexities. Participants must manage Paillier cryptosystem keys under various mathematical assumptions, such as the residuosity class problem, which adds to the overall intricacy of the system

[150] made enhancements to the JF-DKG protocol by adopting a public channel and ensuring a uniform distribution of keys. Their approach also eliminated the need for participants to reveal their shares in case of a complaint. However, a vulnerability was identified in the protocol related to the handling of complaints. The protocol resumes the dealing phase upon receiving an invalid share, which can leave it susceptible to denial of service (DoS) attacks initiated by corrupted participants submitting invalid shares during the complaint phase, unless the number of possible malicious behaviours has been pre-defined.

In [149], a sparse matrix-based DKG protocol was proposed, which achieved poly-logarithmic communication and computational costs per participant without the need for global broadcast. This scheme comes with the requirement of performing a permutation pre-processing step, which must be conducted by a trusted dealer before the protocol can be executed.

[148, 160] proposed the first semi-synchronous DKG based on Pedersen's DKG protocol, with a communication complexity of $\mathcal{O}(n^4)$. This protocol can only tolerate up to $1/3$ faulty participants and requires patching a Byzantine agreement protocol to DKG. Their scheme was further improved by [153] using authenticated multi-point evaluation trees (AMT) to reduce the computation complexity to $\mathcal{O}(n/\log n)$, with the trade-off of an increase in communication complexity.

Another approach was presented by [156], which proposed a non-interactive DKG protocol based on a new PVSS scheme and a broadcast channel. This protocol utilises a verifiable unpredictable function and aggregation over a gossip network in a continuous manner. In this protocol, the generated secret key is a group element rather than a field element, which limits its applicability in scenarios like threshold encryption and signature

schemes.

In addition to the mentioned works, DKG has been explored under various security assumptions, including Static Security ([146]), Adaptive Security ([168]), and Proactive Security ([160]). Under the static security assumption, the adversary needs to determine the corrupted set before the protocol begins, while under the adaptive secure assumption, the adversary can corrupt participants at any time during the protocol execution. For the proactive secure assumption, the adversary can corrupt and un-corrupt participants during the protocol execution.

In this thesis, we focus on static corruption as our primary security assumption. An interesting extension of the current work would be to investigate how to achieve adaptive security ([147, 169, 170, 168, 171]) and proactive security ([172, 160]) in our DKG protocol. Such extensions would enhance the robustness and flexibility of the protocol in the face of more challenging and dynamic adversary models.

In terms of communication channels, various types have been explored for DKG, including private channels ([38]), public channels ([150, 173]), and authenticated channels ([157]). In this thesis, we consider a public communication channel based on the blockchain, which provides a transparent and decentralised platform for communication among participants.

Different communication models have also been considered in DKG, including synchronous communications ([145, 48, 151, 149, 38, 150, 174, 154, 156]), semi-synchronous communications ([148]), and asynchronous communications ([175]). The synchronous communication model assumes that all participants have a known upper bound on the time taken to deliver messages to each other, while the semi-synchronous model allows for variable message delivery times but has a known upper bound on the time for message delivery. The asynchronous model does not impose any timing assumptions on message delivery.

In this thesis, we focus on the synchronous communication model, following the works of [126] and [152]. The synchronous model is practical in many scenarios and allows for efficient communication and coordination among participants, making it well-suited for our proposed DKG protocol over the blockchain platform.

In recent years, DKG has emerged as an essential component in blockchain systems, driven by the development of blockchain technology. For example, in the time-lock encryption protocol, [174] utilises DKG to provide the key pair setup for threshold encryption. DKG has also been employed by Honey Badger [176] to prevent single point failure in distributed systems. Trustworthy randomness beacon is achieved using DKG in [177], while [178] uses DKG to create a common coin in consensus protocols. Additionally, [173] leverages DKG to provide a multi-signature wallet based on threshold signature schemes.

However, in most of these cases, existing DKG protocols are utilised. For instance, [152] implements the JF-DKG protocol with symmetric encryption using a Deffie-Hellman Key Exchange protocol. In their work, Ethereum serves as the public communication channel,

and a smart contract is used to automatically verify complaints, similar to [155]. Furthermore, [179] develops a public key encryption scheme with keyword search based on the JF-DKG protocol.

The utilisation of DKG in various blockchain-based applications demonstrates its importance in achieving secure and decentralised operations. In this chapter, we propose a new Distributed Batch Key Generation (DBKG) method to address some of the limitations and inefficiencies observed in existing DKG protocols, enabling more efficient and scalable key generation in blockchain systems.

5.3 Distributed Batched Key Generation Functionality $\mathcal{F}_{\text{DBKG}}^{n,t,m}$

In this section, we present the design of a Distributed Batch Key Generation (DBKG) Functionality, denoted as $\mathcal{F}_{\text{DBKG}}^{n,t,m}$ in Fig. 5.1. The DBKG functionality is responsible for generating m global key pairs, which consist of global public keys and global secret keys. It interacts with a set of participants denoted as $\mathsf{P} := \{P_1, \ldots, P_n\}$, and an ideal adversary, denoted as S. The DBKG functionality is parameterised by the threshold value t and the number of generated global key pairs m.

In the DBKG functionality, we define two sets, namely $\mathsf{P_c}$ for the set of corrupted participants and $\mathsf{P_h}$ for the set of honest participants, such that $|\mathsf{P_c}| + |\mathsf{P_h}| = n$, and $|\mathsf{P_c}| \leq t-1$. Initially, the set $\mathbf{N}$ is set to be empty and is maintained throughout the process.

The goal of the DBKG functionality is to securely generate m global key pairs while ensuring that the keys remain secret and are correctly distributed among the honest participants. The DBKG functionality ensures that at least t honest participants must cooperate to reconstruct any of the generated global secret keys.

Each participant, denoted as P_i, initiates the key generation process by sending the message $(\mathsf{Gen}, \mathsf{sid}, P_i)$ to the DBKG functionality $\mathcal{F}_{\text{DBKG}}^{n,t,m}$. Upon receiving this message, $\mathcal{F}_{\text{DBKG}}^{n,t,m}$ adds P_i to the set $\mathbf{N}$ and notifies the ideal adversary S about P_i's request to start key generation using the message $(\textsc{KeyGenNotify}, \mathsf{sid}, P_i)$. The DBKG functionality then proceeds to the next step and repeats this process until all participants have sent the message $(\mathsf{Gen}, \mathsf{sid}, P_i)$, and $|\mathbf{N}| = n$.

The DBKG functionality ensures that each generated global key pair is independent and uniformly distributed. This is a crucial aspect in achieving security and fairness in the generation of multiple global key pairs for the proposed system. To ensure the uniform distribution of global public keys and secret keys, the DBKG functionality $\mathcal{F}_{\text{DBKG}}^{n,t,m}$ first selects random global secret keys, denoted as $\{\mathsf{gsk}_v\}_{v=1}^{m}$, where $v \in [m]$. Subsequently, $\mathcal{F}_{\text{DBKG}}^{n,t,m}$ computes the corresponding global public keys by:

$$\mathsf{gpk}_v := g^{\mathsf{gsk}_v}. \tag{5.1}$$

In order to guarantee adversary's power of controlling global secret keys shares (partial secret keys) of corrupted participants, S can send corrupted participants' partial secret keys to $\mathcal{F}_{\text{DBKG}}^{n,t,m}$ by $(\textsc{CorruptShares}, \mathsf{sid}, \{j, \{\mathsf{psk}_{j,v}\}_{v=1}^{m}\}_{P_j \in \mathsf{P_c}})$. $\mathcal{F}_{\text{DBKG}}^{n,t,m}$ then constructs m degree-$(t-1)$ polynomials based on m random global secret keys, $m \cdot |\mathsf{P_c}|$ corrupted partial secret keys and $m \cdot a$ random partial secret keys of honest participants: $\mathcal{F}_{\text{DBKG}}^{n,t,m}$ sets $a := t - |\mathsf{P_c}| - 1$, $\mathsf{P_h}' \subset \mathsf{P_h}$, $|\mathsf{P_h}'| = a$, selects random $\{\mathsf{psk}_{i,v}\}_{P_i \in \mathsf{P_h}', v \in [m]}$. For $v \in [m]$, $\mathcal{F}_{\text{DBKG}}^{n,t,m}$ constructs random polynomial, $F_v(x) := \sum_{b=0}^{t-1} a_b \cdot x^b$, under the restriction $F_v(j) = \mathsf{psk}_{j,v}$ for $P_j \in \{\mathsf{P_c} \cup \mathsf{P_h}'\}$, and $F_v(0) = \mathsf{gsk}_v$. Then honest participants' partial

secret keys and partial public keys can be computed by

$$\begin{aligned} \mathsf{psk}_{v,i} &:= F_v(i), \\ \mathsf{ppk}_{v,i} &:= g^{\mathsf{psk}_{v,i}}. \end{aligned} \tag{5.2}$$

Then $\mathcal{F}_{\text{DBKG}}^{n,t,m}$ sends partial secret key, $\{\mathsf{psk}_{v,i}\}_{v=1}^{m}$, to $P_i \in \mathsf{P}$.

Additionally, any party can request global public keys and partial public keys by $(\text{READPK}, \mathsf{sid})$, $\mathcal{F}_{\text{DBKG}}^{n,t,m}$ returns global public keys and the partial public keys of participants by $(\text{READPKRETURN}, \mathsf{sid}, \{\mathsf{gpk}_v\}_{v=1}^{m}, \{\mathsf{ppk}_{v,i}\}_{v=1,i=1}^{m,n})$.

DBKG Ideal Functionality $\mathcal{F}_{\text{DBKG}}^{n,t,m}$

$\mathcal{F}_{\text{DBKG}}^{n,t,m}$ interacts with participants, $\mathsf{P} := \{P_1, \ldots, P_n\}$, and ideal adversary, S. It's parameterised with threshold, t, and the number of generated global key pairs, m. Denote $\mathsf{P_c}$ as the set of corrupted participants, and $\mathsf{P_h}$ as the set of honest participants, $|\mathsf{P_c}| + |\mathsf{P_h}| = n$, and $|\mathsf{P_c}| \leq t-1$. $\mathcal{F}_{\text{DBKG}}^{n,t,m}$ maintains a set, $\mathbf{N}$ (initially set to $\emptyset$).

$\mathcal{F}_{\text{DBKG}}^{n,t,m}$ does the following:

- Upon receiving $(\mathsf{Gen}, \mathsf{sid}, P_i)$ from $P_i \in \mathsf{P}$:
  - Set $\mathbf{N} = \mathbf{N} \cup \{P_i\}$;
  - Send $(\text{KEYGENNOTIFY}, \mathsf{sid}, P_i)$ to S;
  - Continue to next step until $|\mathbf{N}| = n$.
- Upon receiving $(\text{CORRUPTSHARES}, \mathsf{sid}, \{j, \{\mathsf{psk}_{j,v}\}_{v=1}^{m}\}_{P_j \in \mathsf{P_c}})$ from S:
  - Pick $\{\mathsf{gsk}_v\}_{v=1}^{m} \leftarrow (\mathbb{Z}_q)^{[m]}$;
  - Compute $\mathsf{gpk}_v := g^{\mathsf{gsk}_v}$ for $v \in [m]$;
  - Set $a := t - |\mathsf{P_c}| - 1$, $\mathsf{P_h}' \subset \mathsf{P_h}$, $|\mathsf{P_h}'| = a$;
  - Select $\{\mathsf{psk}_{i,v}\}_{P_i \in \mathsf{P_h}', v \in [m]} \leftarrow (\mathbb{Z}_q)^{[m \cdot a]}$;
  - For $v \in [m]$, construct random polynomial $F_v(x) := \sum_{b=0}^{t-1} a_b \cdot x^b$ under the restriction $F_v(j) = \mathsf{psk}_{j,v}$ for $P_j \in \{\mathsf{P_c} \cup \mathsf{P_h}'\}$, and $F_v(0) = \mathsf{gsk}_v$;
  - Compute $\mathsf{psk}_{v,i} := F_v(i)$ and $\mathsf{ppk}_{v,i} := g^{\mathsf{psk}_{v,i}}$ for $P_i \in \mathsf{P_h}$, $v \in [m]$;
  - Send $(\mathsf{sid}, \{\mathsf{psk}_{v,i}\}_{v=1}^{m})$ to $P_i \in \mathsf{P}$;
- Upon receiving $(\text{READPK}, \mathsf{sid})$ from any party, return the corresponding party the message $(\text{READPKRETURN}, \mathsf{sid}, \{\mathsf{gpk}_v\}_{v=1}^{m}, \{\mathsf{ppk}_{v,i}\}_{v=1,i=1}^{m,n})$.

Figure 5.1: The ideal functionality $\mathcal{F}_{\text{DBKG}}^{n,t,m}$.

5.4 HIM based DBKG protocol $\Pi^{n,t,m}_{\text{DBKG}}$

Naively, one could generate multiple keys by executing multiple DKG protocols, such as the one proposed by Gennaro et al. [38], with linear complexity. However, a more efficient approach is to leverage Hyper-Invertible Matrix (HIM) based random secret sharing (as shown in Figure 2.3 and introduced in [25]), which allows sharing n randomness values with only n^2 sharings, resulting in an amortized communication cost of $\mathcal{O}(n)$.

Nevertheless, due to the feasible mapping from inputs to outputs in HIM (as described in Theorem 5 and Property 2), ensuring the uniformly distributed global key pairs requires generating only $n - t$ pairs, where t is the threshold (with $|\mathsf{P_c}| \leq t - 1$, $\mathsf{P_c}$ being the set of corrupted participants). In this section, we introduce the HIM based Distributed Batch Key Generation (DBKG) Protocol $\Pi^{n,t,m}_{\text{DBKG}}$, which accomplishes the generation of multiple global key pairs. Subsequently, we explain the zero-knowledge proofs used in the protocol in Section 5.5 and provide an analysis of the security of the HIM based DBKG in Section 5.6. This approach enables an efficient and secure generation of multiple global key pairs for the proposed system.

Figure 5.2 and Figure 5.3 present the protocol $\Pi^{n,t,m}_{\text{DBKG}}$ which *UC-realises* $\mathcal{F}^{n,t,m}_{\text{DBKG}}$ in $\{\mathcal{F}_{\text{BC}}\}$-*hybrid world* ($\mathcal{F}_{\text{BC}}$ is a blockchain ideal functionality taken from [47]).

Initially, every participant is assumed to hold a same predefined Hyper-Invertible Matrix, **HIM**, denoted by $\{\lambda_{v,w}\}^{n,n}_{v=1,w=1}$. Let η, p be the Elgamal encryption scheme parameters, the whole DKG process is triggered by the command, $(\mathsf{Gen}, \mathsf{sid}, P_i)$ from $\mathcal{Z}$ to P_i.

P_i selects a random value, x_i, from $\mathbb{Z}_q$, and polynomial parameters, $\{a_{i,k}\}^{t-1}_{k=1}$ from $(\mathbb{Z}_q)^{[t-1]}$. The polynomial can be constructed by

$$F_i(z) := \sum_{k=0}^{t-1} a_{i,k} \cdot z^k, \tag{5.3}$$

where $a_{i,0} := x_i$. Then P_i shares x_i to other participants with polynomial $F_i(\cdot)$ by encrypting shares with receiver's public key under Lifted Elgamal encryption:

$$\begin{aligned} &s_{i,j} := F_i(j) \\ &(d_{i,j,1}, \ldots, d_{i,j,\eta}) := \mathsf{LEG.ENCODE}(s_{i,j}) \text{ for } j \in [n] \\ &(A_{i,j,b}, B_{i,j,b}) := \mathsf{LEG.Enc}_{\mathsf{pk}_j}(d_{i,j,b}; r_{i,j,b}) \text{ for } j \in [n] \text{ and } b \in [\eta], \end{aligned} \tag{5.4}$$

where $\{r_{i,j,b}\}^{n,\eta}_{j=1,b=1}$ are randomly picked from $\mathbb{Z}_q$.

In addition, P_i needs to commit the polynomial for VSS,

$$C_{i,k} := \mathsf{Com}_{\mathsf{ck}}(a_{i,k}; a'_{i,k}) \text{ for } k \in [0, t-1], \tag{5.5}$$

where $\{a'_{i,k}\}^{t-1}_{k=0}$ are randomly picked from $(\mathbb{Z}_q)^{[t]}$. To show the cipher-texts sent by P_i are correctly computed from $F_i(\cdot)$, P_i should generate Correct Sharing NIZK (Figure 5.4), σ_i, to

show that each share is correctly computed from the polynomial based on the commitment and ciphertexts:

$$\sigma_i \leftarrow \mathsf{NIZK}\left\{\begin{array}{l}(g, \mathsf{ck}, \{\mathsf{pk}_j\}_{j=1}^n, \{A_{i,j,b}, B_{i,j,b}\}_{j=1,b=1}^{n,\eta}, \{C_{i,k}\}_{k=0}^{t-1}),\\ (\{s_{i,j}\}_{j=1}^n, \{r_{i,j,b}\}_{j=1,b=1}^{n,\eta}, \{a_{i,k}, a'_{i,k}\}_{k=0}^{t-1}) :\\ \{\prod_{b=1}^{\eta}(A_{i,j,b})^{p^b} = g^{\sum_{b=1}^{\eta} r_{i,j,b}\cdot p^b}\}_{j=1}^n\\ \wedge\{\prod_{b=1}^{\eta}(B_{i,j,b})^{p^b} = g^{s_{i,j}} \cdot \mathsf{pk}_j^{\sum_{b=1}^{\eta} r_{i,j,b}\cdot p^b}\}_{j=1}^n\\ \wedge\{\prod_{k=0}^{t-1}(C_{i,k})^{j^k} = g^{s_{i,j}} \cdot \mathsf{ck}^{\sum_{k=0}^{t-1}(a'_{i,k})^{j^k}}\}_{j=1}^n\end{array}\right\}.$$

P_i posts the ciphertexts together with commitment and σ_i to $\mathcal{F}_{\text{BC}}$.

In next round, P_i gets its shares $\{s_{j,i}\}_{j=1}^n$ from other participants by decryption the cipher-texts with its own secret key, sk_i:

$$\{s_{j,i} = \text{DECODE}(\{\mathsf{LEG.Dec}_{\mathsf{sk}_i}(A_{j,i,b}, B_{j,i,b})\}_{b=1}^{\eta})\}_{j=1}^n. \tag{5.6}$$

P_i can constructs a qualified participant set QUAL by verifying Correct Sharing NIZK proofs (Figure 5.4). P_i's m partial secret keys can be locally computed based on the shares from qualified participants and corresponding **HIM** parameters, with HIM based random secret sharing,

$$\mathsf{psk}_{v,i} := \sum_{j\in\mathsf{QUAL}} \lambda_{v,j} \cdot s_{j,i} \text{ for } v \in \mathbf{N}. \tag{5.7}$$

For threshold decryption (Section 2.9.2), P_i needs to commit its polynomial with randomness zero and posts the commitment to $\mathcal{F}_{\text{BC}}$,

$$D_{i,k} := \mathsf{Com}_{\mathsf{ck}}(a_{i,k}; 0) \text{ for } k \in [0, t-1]. \tag{5.8}$$

In next round, P_i gets other participants' commitments, $\{D_{j,k}\}_{j\in\mathsf{QUAL},k\in[0,t-1]}$ and checks if they are consistent with, $\{s_{j,i}\}_{j\in\mathsf{QUAL}}$,

$$\prod_{k=0}^{t-1}(D_{j,k})^{i^k} = g^{s_{j,i}}. \tag{5.9}$$

If verification about a participant, P_j, fails, P_i needs to reveal $s_{j,i}$ and clarify a valid complain about P_j, by generating a Correct Decryption NIZK σ'_i (Figure 5.5) to prove that it indeed gets $s_{j,i}$ from decrypting the ciphertexts encrypted by its public key sent by P_j:

$$\sigma_i \leftarrow \mathsf{NIZK}\left\{\begin{array}{l}(g, \mathsf{ck}, \{\mathsf{pk}_j\}_{j=1}^n, \{A_{i,j,b}, B_{i,j,b}\}_{j=1,b=1}^{n,\eta}, \{C_{i,k}\}_{k=0}^{t-1}),\\ (\{s_{i,j}\}_{j=1}^n, \{r_{i,j,b}\}_{j=1,b=1}^{n,\eta}, \{a_{i,k}, a'_{i,k}\}_{k=0}^{t-1}) :\\ \{\prod_{b=1}^{\eta}(A_{i,j,b})^{p^b} = g^{\sum_{b=1}^{\eta} r_{i,j,b}\cdot p^b}\}_{j=1}^n\\ \wedge\{\prod_{b=1}^{\eta}(B_{i,j,b})^{p^b} = g^{s_{i,j}} \cdot \mathsf{pk}_j^{\sum_{b=1}^{\eta} r_{i,j,b}\cdot p^b}\}_{j=1}^n\\ \wedge\{\prod_{k=0}^{t-1}(C_{i,k})^{j^k} = g^{s_{i,j}} \cdot \mathsf{ck}^{\sum_{k=0}^{t-1}(a'_{i,k})^{j^k}}\}_{j=1}^n\end{array}\right\}.$$

The rest participants can jointly interpolate P_j's polynomial if the complain is valid, then post correct $\{D_{j,k}\}_{k=0}^{t-1}$ to $\mathcal{F}_{\text{BC}}$. P_i can compute the global public keys by

$$\mathsf{gpk}_v := \prod_{j \in \mathsf{QUAL}} (D_{j,0})^{\lambda_{v,j}} \text{for } v \in \mathbf{N}, \tag{5.10}$$

where $\mathbf{N} \subset [n]$ is a predefined set and $|\mathbf{N}| = m$.

The environment, $\mathcal{Z}$, can send $(\text{READPK}, \mathsf{sid})$ to any participant, P. P computes

$$\mathsf{ppk}_{v,i} = \prod_{j \in \mathsf{QUAL}} (\prod_{k=0}^{t-1} (D_{j,k})^{i^k})^{\lambda_{v,j}} \text{ for } v \in \mathbf{N} \text{ and } i \in [n], \tag{5.11}$$

then returns $(\text{READPKRETURN}, \mathsf{sid}, (\{\mathsf{gpk}_v\}_{v \in \mathbf{N}}, \{\mathsf{ppk}_{v,i}\}_{v \in \mathbf{N}, i \in [n]}))$ to the environment $\mathcal{Z}$.

The HIM based DBKG Protocol $\Pi^{n,t,m}_{\text{DBKG}}$ (Part 1)

Denote participants set by $\mathsf{P} := \{P_1, \ldots, P_n\}$, each participant has a predefined HIM: $\mathbf{HIM} := \{\lambda_{v,w}\}^{n,n}_{v=1,w=1}$ and a predefined set $\mathbf{N} \subset [n]$, where $|\mathbf{N}| = m$. Denote the Elgamal encryption scheme parameters by η, p.

- Upon receiving $(\mathsf{Gen}, \mathsf{sid}, P_i)$ from $\mathcal{Z}$, $P_i \in \mathsf{P}$ does the following:
  - Select $x_i \leftarrow \mathbb{Z}_q$, $\{a_{i,k}\}^{t-1}_{k=1} \leftarrow (\mathbb{Z}_q)^{[t-1]}$, construct a polynomial $F_i(z) := \sum^{t-1}_{k=0} a_{i,k} \cdot z^k$, where $a_{i,0} := x_i$;
  - Select $\{a'_{i,k}\}^{t-1}_{k=0} \leftarrow (\mathbb{Z}_q)^{[t]}$;
  - Compute $C_{i,k} := \mathsf{Com}_{\mathsf{ck}}(a_{i,k}; a'_{i,k})$ for $k \in [0, t-1]$;
  - Compute $s_{i,j} := F_i(j)$, and $(d_{i,j,1}, \ldots, d_{i,j,\eta}) := \textsc{Encode}(s_{i,j})$ for $j \in [n]$;
  - Select $r_{i,j,b} \leftarrow \mathbb{Z}_q$;
  - Compute $(A_{i,j,b}, B_{i,j,b}) := \mathsf{LEG.Enc}_{\mathsf{pk}_j}(d_{i,j,b}; r_{i,j,b})$ for $j \in [n]$ and $b \in [\eta]$;
  - Generate Correct Sharing NIZK σ_i Cf. Figure 5.4:

$$\sigma_i \leftarrow \mathsf{NIZK}\left\{\begin{array}{l}(g, \mathsf{ck}, \{\mathsf{pk}_j\}^n_{j=1}, \{A_{i,j,b}, B_{i,j,b}\}^{n,\eta}_{j=1,b=1}, \{C_{i,k}\}^{t-1}_{k=0}),\\ (\{s_{i,j}\}^n_{j=1}, \{r_{i,j,b}\}^{n,\eta}_{j=1,b=1}, \{a_{i,k}, a'_{i,k}\}^{t-1}_{k=0}) :\\ \{\prod^{\eta}_{b=1}(A_{i,j,b})^{p^b} = g^{\sum^{\eta}_{b=1} r_{i,j,b} \cdot p^b}\}^n_{j=1}\\ \wedge\{\prod^{\eta}_{b=1}(B_{i,j,b})^{p^b} = g^{s_{i,j}} \cdot \mathsf{pk}_j^{\sum^{\eta}_{b=1} r_{i,j,b} \cdot p^b}\}^n_{j=1}\\ \wedge\{\prod^{t-1}_{k=0}(C_{i,k})^{j^k} = g^{s_{i,j}} \cdot \mathsf{ck}^{\sum^{t-1}_{k=0}(a'_{i,k})^{j^k}}\}^n_{j=1}\end{array}\right\}.$$

  - Send $(\mathsf{Write}, \mathsf{sid}, (\{A_{i,j,b}, B_{i,j,b}\}^{n,\eta}_{j=1,b=1}, \sigma_i, \{C_{i,k}\}^{t-1}_{k=0}))$ to $\mathcal{F}_{\text{BC}}$[a].
  - Send $(\mathsf{Read}, \mathsf{sid})$ to $\mathcal{F}_{\text{BC}}$ and get $(\{\{A_{j,i,b}, B_{j,i,b}\}^{\eta}_{b=1}, \sigma_j, \{C_{j,k}\}^{t-1}_{k=0}\}\}^n_{j=1})$;
  - Compute $\{s_{j,i} = \textsc{Decode}(\{\mathsf{LEG.Dec}_{\mathsf{sk}_i}(A_{i,j,b}, B_{j,i,b})\}^{\eta}_{b=1})\}^n_{j=1}$;
  - For $j \in [n]$, if $\mathsf{Verify}(\sigma_j, \{A_{i,j,b}, B_{i,j,b}\}^{n,\eta}_{j=1,b=1}, \{C_{i,k}\}^{t-1}_{k=0}) = 0$, set $\mathsf{QUAL} = [n] \setminus \{j\}$. Compute $D_{i,k} := \mathsf{Com}_{\mathsf{ck}}(a_{i,k}; 0)$ for $k \in [0, t-1]$;
  - Compute its own partial secret keys as $\mathsf{psk}_{v,i} := \sum_{j \in \mathsf{QUAL}} \lambda_{v,j} \cdot s_{j,i}$ for $v \in \mathbf{N}$;

[a] Sender like P_i does not need to post message for itself, we use $j \in [n]$ instead of $j \in \{[n] \setminus \{i\}\}$ to make protocol neat.

Figure 5.2: HIM based DBKG Protocol $\Pi^{n,t,m}_{\text{DBKG}}$ in $\{\mathcal{F}_{\text{BC}}\}$-*hybrid world* (Part 1).

The HIM based DBKG Protocol $\Pi^{n,t,m}_{\text{DBKG}}$ (Part 2)

- Upon receiving $(\mathsf{Gen}, \mathsf{sid}, P_i)$ from $\mathcal{Z}$, $P_i \in \mathsf{P}$ does the following after the processing in Figure 5.2:
  - Send $(\mathsf{Write}, \mathsf{sid}, (\{D_{i,k}\}_{k=0}^{t-1}))$ to $\mathcal{F}_{\text{BC}}$;
  - Send $(\mathsf{Read}, \mathsf{sid})$ to $\mathcal{F}_{\text{BC}}$, and get $(\{\{D_{j,k}\}_{k=0}^{t-1}\}_{j\in\mathsf{QUAL}\setminus\{i\}})$;
  - Check if $\prod_{k=0}^{t-1}(D_{j,k})^{i^k} = g^{s_{j,i}}$ holds. If verification fails for P_j, Send $(\mathsf{Write}, \mathsf{sid}, (\text{COMPLAINT}, s_{j,i}, \sigma'_i))$ to $\mathcal{F}_{\text{BC}}$, where σ'_i is a Correct Decryption NIZK, Cf. Figure 5.5:

$$\sigma'_i \leftarrow \mathsf{NIZK}\left\{\begin{array}{l}(g, \mathsf{pk}_i, \{A_{j,i,b}, B_{j,i,b}\}_{b=1}^{\eta}, s_{j,i}), (\mathsf{sk}_i) : \\ \{d_{j,i,b} = \mathsf{LEG.Dec}_{\mathsf{sk}_i}(A_{j,i,b}, B_{j,i,b})\}_{b=1}^{\eta} \\ \wedge s_{j,i} = \text{DECODE}(\{d_{j,i,b}\}_{b=1}^{\eta}) \ \wedge \mathsf{pk}_i = g^{\mathsf{sk}_i}\end{array}\right\}.$$

  - If any participant $\{P_w\}_{w\in\mathsf{QUAL}}$ posted $(\text{COMPLAINT}, s_{v,w}, \sigma'_w)$ about P_v with $\mathcal{F}_{\text{BC}}$, do the following if $\mathsf{Verify}(\sigma'_k, \{(A_{v,w,b}, B_{v,w,b})\}_{b=1}^{\eta}) = 1$:
    - Send $(\mathsf{Read}, \mathsf{sid})$ to $\mathcal{F}_{\text{BC}}$, and get $(\{s_{v,j}, \sigma'_j\}_{j\in\mathsf{QUAL}})$;
    - Select t values from $\{s_{v,j}\}_{j\in\mathsf{QUAL}}$, where $\prod_{k=0}^{t-1}(D_{v,k})^{j^k} = g^{s_{a,j}}$;
    - Interpolate $F_v(x)$, compute $\{D_{v,k}\}_{k=0}^{t-1}$. Post $(\{D_{v,k}\}_{k=0}^{t-1}, P_w)$ to $\mathcal{F}_{\text{BC}}$.
  - Compute global public keys as $\mathsf{gpk}_v := \prod_{j\in\mathsf{QUAL}}(D_{j,0})^{\lambda_{v,j}}$ for $v \in \mathbf{N}$;
  - Send $(\mathsf{Write}, \mathsf{sid}, (\{\mathsf{gpk}_v\}_{v\in\mathbf{N}}))$ to $\mathcal{F}_{\text{BC}}$.
- Upon receiving $(\text{READPK}, \mathsf{sid})$ from $\mathcal{Z}$, the participant P does the following:
  - Send $(\mathsf{Read}, \mathsf{sid})$ to $\mathcal{F}_{\text{BC}}$ and get $(\{\mathsf{gpk}_v\}_{v\in\mathbf{N}}, \{D_{j,k}\}_{k=0,j=1}^{t-1,n})$;
  - Compute $\mathsf{ppk}_{v,i} = \prod_{j\in\mathsf{QUAL}}(\prod_{k=0}^{t-1}(D_{j,k})^{i^k})^{\lambda_{v,j}}$ for $v \in \mathbf{N}$ and $i \in [n]$;
  - Return $(\text{READPKRETURN}, \mathsf{sid}, (\{\mathsf{gpk}_v\}_{v\in\mathbf{N}}, \{\mathsf{ppk}_{v,i}\}_{v\in\mathbf{N}, i\in[n]}))$ to $\mathcal{Z}$.

Figure 5.3: HIM based DBKG Protocol $\Pi^{n,t,m}_{\text{DBKG}}$ in $\{\mathcal{F}_{\text{BC}}\}$-*hybrid world* (Part 2).

5.5 Zero-Knowledge Proofs in $\Pi^{n,t,m}_{\text{DBKG}}$

5.5.1 Correct Sharing Proof

The goal of Correct Sharing Proof is to show a party, P_i, correctly shared $\{s_{i,j}\}_{j=1}^{n}$ to other parties in DBKG Protocol, $\Pi^{n,t,m}_{\text{DBKG}}$ (Figure 5.2). The idea behind this proof is to show the consistence of $\{s_{i,j}\}_{j=1}^{n}$ in commitments of polynomial, $\{C_{i,k}\}_{k=0}^{t-1}$, and ciphertexts, $\{A_{i,j,b}, B_{i,j,b}\}_{j=1,b=1}^{n,\eta}$.

Batch verification [180] and Schnorr protocol [181] are used to construct construct correct sharing proof in Figure 5.4. In this SHVZK proof, $\mathcal{P}$ wants to prove to $\mathcal{V}$ the knowledge of $\{r_{i,j,b}\}_{j=1,b=1}^{n,\eta}$, $\{s_{i,j}\}_{j=1}^{n}$, $\{a_{i,k}, a'_{i,k}\}_{k=0}^{t-1}$ in ciphertexts, $\{A_{i,j,b}, B_{i,j,b}\}_{j=1,b=1}^{n,\eta}$ and commitments of polynomial, $\{C_{i,k}\}_{k=0}^{t-1}$.

In the first round, $\mathcal{V}$ selects a random λ for batch verification, which can also be computed by hashing the statement for NIZK

$$\lambda \leftarrow \mathsf{hash}(\langle C_{i,k}\rangle_{k=0}^{t-1}, \langle A_{i,j,b}, B_{i,j,b}\rangle_{j=1,b=1}^{n,\eta}\rangle. \tag{5.12}$$

In the second round, $\mathcal{P}$ generates random commitments and ciphertexts, and sends to $\mathcal{V}$. Firstly, $\mathcal{P}$ chooses random $\{a_j, c_j\}_{j=1}^{n}$ and b. Then it computes

$$\begin{aligned} &a := \sum_{j=1}^{n} a_j \cdot \lambda^{j-1}, c := \sum_{j=1}^{n} c_j \cdot \lambda^{j-1}, \\ &A := g^a \cdot \mathsf{ck}^b, C := g^c, \\ &B_j := g^{a_j} \cdot \mathsf{pk}_j^{c_j} \text{ for } j \in [n]. \end{aligned} \tag{5.13}$$

$\mathcal{P}$ sends $A, \{B_j\}_{j=1}^{n}, C$ to $\mathcal{V}$. Then $\mathcal{V}$ responses by either selecting a random challenge, e, or hashing the messages sent by $\mathcal{P}$ in last round

$$e \leftarrow \mathsf{hash}(\langle A, \langle B_j\rangle_{j=1}^{n}, C\rangle) \tag{5.14}$$

In the last round, $\mathcal{P}$ masks secret shares, $\{s_{i,j}\}_{j=1}^{n}$, commitment randomnesses, $\{a'_{i,k}\}_{k=0}^{t-1}$, and encryption randomnesses, $\{r_{i,j,b}\}_{b=1,j=1}^{\eta,n}$ based on challenge, e and the randomnesses of random commitments and ciphertexts selected in the second round

$$\begin{aligned} &Z_{j,1} := s_{i,j} \cdot e + a_j \text{ for } j \in [n], \\ &Z_{j,3} := (\sum_{b=1}^{\eta} r_{i,j,b} \cdot p^b) \cdot e + c_j \text{ for } j \in [n], \\ &Z_2 := (\sum_{j=1}^{n} (\sum_{k=0}^{t-1} a'_{i,k} \cdot j^k) \cdot \lambda^{j-1}) \cdot e + b. \end{aligned} \tag{5.15}$$

Then $\mathcal{P}$ sends $\{\{Z_{j,1}, Z_{j,3}\}_{j=1}^{n}, Z_2\}$ to $\mathcal{V}$.

To decide whether to accept $\mathcal{P}$'s response, $\mathcal{V}$ computes the following

$$\begin{aligned}
&e \leftarrow \mathsf{hash}(\langle A, \langle B_j \rangle_{j=1}^{n}, C \rangle) \text{ for } \mathsf{NIZK},\\
&\lambda \leftarrow \mathsf{hash}(\langle C_{i,k} \rangle_{k=0}^{t-1}, \langle A_{i,j,b}, B_{i,j,b} \rangle_{j=1,b=1}^{n,\eta} \rangle \text{ for } \mathsf{NIZK},\\
&Z_1 := \sum_{j=1}^{n} Z_{j,1} \cdot \lambda^{j-1},\\
&Z_3 := \sum_{j=1}^{n} Z_{j,3} \cdot \lambda^{j-1},\\
&D := \prod_{j=1}^{n} (\prod_{k=0}^{t-1} ((C_{i,k})^{j^k}))^{\lambda^{j-1}},\\
&E_j := \prod_{b=1}^{\eta} (B_{i,j,b})^{p^b} \text{ for } j \in [n],\\
&F := \prod_{j=1}^{n} (\prod_{b=1}^{\eta} (A_{i,j,b})^{p^b})^{\lambda^{j-1}}.
\end{aligned} \tag{5.16}$$

Then $\mathcal{V}$ checks if the following hold:

$$\begin{aligned}
&(D)^e \cdot A = g^{Z_1} \cdot \mathsf{ck}^{Z_2}\\
&(E_j)^e \cdot B_j = g^{Z_{j,1}} \cdot \mathsf{pk}_j^{Z_{j,3}} \text{ for } j \in [n],\\
&(F)^e \cdot C = g^{Z_3}.
\end{aligned} \tag{5.17}$$

Theorem 8 presents the formal proof about correctness, soundness, and zero-knowledge of Correct Sharing ZK argument in $\Pi_{\mathrm{DBKG}}^{n,t,m}$.

Correct Sharing ZK argument

CRS: $\{g, \mathsf{ck}, \{\mathsf{pk}_j\}_{j=1}^n\} \in \mathbb{G} \setminus \{1\}$
Statement: $\{C_{i,k}\}_{k=0}^{t-1}$, $\{A_{i,j,b}, B_{i,j,b}\}_{j=1,b=1}^{n,\eta}$
Witness: $\{r_{i,j,b}\}_{j=1,b=1}^{n,\eta}, \{s_{i,j}\}_{j=1}^n, \{a_{i,k}, a'_{i,k}\}_{k=0}^{t-1}$
Protocol:

- $\mathcal{P} \leftarrow \mathcal{V} : \lambda \leftarrow \mathbb{Z}_q$;
  /*Set $\lambda \leftarrow \mathsf{hash}(\langle C_{i,k}\rangle_{k=0}^{t-1}, \langle A_{i,j,b}, B_{i,j,b}\rangle_{j=1,b=1}^{n,\eta}\rangle$ for NIZK*/
- $\mathcal{P}$ does the following:
  - Select $\{a_j, c_j\}_{j=1}^n \leftarrow (\mathbb{Z}_q)^{[2\cdot n]}$;
  - Select $b \leftarrow \mathbb{Z}_q$;
  - Compute $a := \sum_{j=1}^n a_j \cdot \lambda^{j-1}$;
  - Compute $c := \sum_{j=1}^n c_j \cdot \lambda^{j-1}$;
  - Compute $A := g^a \cdot \mathsf{ck}^b$, $C := g^c$;
  - Compute $B_j := g^{a_j} \cdot \mathsf{pk}_j^{c_j}$ for $j \in [n]$;
  - $\mathcal{V} \leftarrow \mathcal{P} : \langle A, \{B_j\}_{j=1}^n, C\rangle$.
- $\mathcal{P} \leftarrow \mathcal{V} : e \leftarrow \mathbb{Z}_q$;
  /*Set $e \leftarrow \mathsf{hash}(\langle A, \langle B_j\rangle_{j=1}^n, C\rangle)$ for NIZK*/
- $\mathcal{P}$ computes the following:
  - $Z_{j,1} := s_{i,j} \cdot e + a_j$ for $j \in [n]$;
  - $Z_{j,3} := (\sum_{b=1}^{\eta} r_{i,j,b} \cdot p^b) \cdot e + c_j$ for $j \in [n]$;
  - $Z_2 := (\sum_{j=1}^n (\sum_{k=0}^{t-1} a'_{i,k} \cdot j^k) \cdot \lambda^{j-1}) \cdot e + b$;
  - $\mathcal{V} \leftarrow \mathcal{P} : \langle \{Z_{j,1}, Z_{j,3}\}_{j=1}^n, Z_2\rangle$.

Verification:

- $\mathcal{V}$ computes the following:
  - $e \leftarrow \mathsf{hash}(\langle A, \langle B_j\rangle_{j=1}^n, C\rangle)$ for NIZK;
  - $\lambda \leftarrow \mathsf{hash}(\langle C_{i,k}\rangle_{k=0}^{t-1}, \langle A_{i,j,b}, B_{i,j,b}\rangle_{j=1,b=1}^{n,\eta}\rangle$ for NIZK;
  - $Z_1 := \sum_{j=1}^n Z_{j,1} \cdot \lambda^{j-1}$;
  - $Z_3 := \sum_{j=1}^n Z_{j,3} \cdot \lambda^{j-1}$;
  - $D := \prod_{j=1}^n (\prod_{k=0}^{t-1} ((C_{i,k})^{j^k}))^{\lambda^{j-1}}$;
  - $E_j := \prod_{b=1}^{\eta} (B_{i,j,b})^{p^b}$ for $j \in [n]$;
  - $F := \prod_{j=1}^n (\prod_{b=1}^{\eta} (A_{i,j,b})^{p^b})^{\lambda^{j-1}}$.
- $\mathcal{V}$ checks the following:
  - $(D)^e \cdot A = g^{Z_1} \cdot \mathsf{ck}^{Z_2}$;
  - $(E_j)^e \cdot B_j = g^{Z_{j,1}} \cdot \mathsf{pk}_j^{Z_{j,3}}$ for $j \in [n]$;
  - $(F)^e \cdot C = g^{Z_3}$.

Figure 5.4: Correct Sharing ZK argument in $\Pi_{\mathrm{DBKG}}^{n,t,m}$.

Theorem 8 (Correct Sharing)**.** *Assume the DDH problem is hard. Let* $F_i(j) = s_{i,j} = \sum_{b=1}^{\eta}(d_{i,j,b}) \cdot p^b$*, the protocol described in Figure 5.4 is an honest verifier zero-knowledge argument of knowledge of* $\{a_{i,k}, a'_{i,k}\}_{k=0}^{t-1}$ *and* $\{r_{i,j,b}\}_{b=1,j=1}^{\eta,n}$ *such that there exists a degree* $(t-1)$ *polynomial:* $F_i(z) = \sum_{k=0}^{t-1} a_{i,t} \cdot z^t$ *that:*

- $C_{i,k} = g^{a_{i,k}} \cdot \mathsf{ck}^{a'_{i,k}}$ *for* $k \in [0, t-1]$*;*
- $(A_{i,j,b}, B_{i,j,b}) = \mathsf{Enc}_{\mathsf{pk}_j}(d_{i,j,b}; r_{i,j,b})$ *for* $j \in [n]$ *and* $b \in [\eta]$.

Proof of Theorem 8.

- **Completeness.**
  Let us demonstrate the completeness of the proposed scheme. Firstly, we observe the following equations:

$$\begin{aligned} D &:= \prod_{j=1}^{n}(\prod_{k=0}^{t-1}((C_{i,k})^{j^k}))^{\lambda^{j-1}} = g^{\sum_{j=1}^{n} s_{i,j}\cdot\lambda^{j-1}} \cdot \mathsf{ck}^{\sum_{j=1}^{n}\sum_{k=0}^{t-1} a'_{i,k}\cdot j^k\cdot\lambda^{j-1}}, \\ A &:= g^a \cdot \mathsf{ck}^b, Z_{j,1} := s_{i,j} \cdot e + a_j, \\ Z_1 &:= \sum_{j=1}^{n} Z_{j,1} \cdot \lambda^{j-1} = (\sum_{j=1}^{n} s_{i,j} \cdot \lambda^{j-1}) \cdot e + a, \\ Z_2 &:= (\sum_{j=1}^{n}(\sum_{k=0}^{t-1} a'_{i,k}) \cdot j^k \cdot \lambda^{j-1}) \cdot e + b. \end{aligned} \tag{5.18}$$

  We can then prove the following equation:

$$(D)^e \cdot A = g^{(\sum_{j=1}^{n} s_{i,j}\cdot\lambda^{j-1})\cdot e+a} \cdot \mathsf{ck}^{(\sum_{j=1}^{n}\sum_{k=0}^{t-1} a'_{i,k}\cdot j^k\cdot\lambda^{j-1})\cdot e+b} = g^{Z_1} \cdot \mathsf{ck}^{Z_2}. \tag{5.19}$$

  In addition, for $j \in [n]$, the following hold

$$E_j := \prod_{b=1}^{\eta}(B_{i,j,b})^{p^b} = g^{s_{i,j}} \cdot \mathsf{pk}_j^{\sum_{b=1}^{\eta} r_{i,j,b}\cdot p^b}, \tag{5.20}$$

$$\begin{aligned} B_j &:= g^{a_j} \cdot \mathsf{pk}_j^{c_j}, Z_{j,1} := s_{i,j} \cdot e + a_j, \\ Z_{j,3} &:= (\sum_{b=1}^{\eta} r_{i,j,b} \cdot p^b) \cdot e + c_j. \end{aligned} \tag{5.21}$$

  It is easy to infer that the following verification holds:

$$(E_j)^e \cdot B_j = g^{(s_{i,j})\cdot e+a_j} \cdot \mathsf{pk}_j^{(\sum_{b=1}^{\eta} r_{i,j,b}\cdot p^b)\cdot e+c_j} = g^{Z_{j,1}} \cdot \mathsf{pk}_j^{Z_{j,3}}. \tag{5.22}$$

Lastly, the following can be deduced

$$\begin{aligned} C &:= g^c, \\ Z_3 &:= \sum_{j=1}^{n} Z_{j,3} \cdot \lambda^{j-1} = (\sum_{j=1}^{n}(\sum_{b=1}^{\eta} r_{i,j,b} \cdot p^b) \cdot \lambda^{j-1}) \cdot e + c, \\ F &:= \prod_{j=1}^{n}(\prod_{k=1}^{\eta}(A_{i,j,b})^{p^b})^{\lambda^{j-1}} = g^{\sum_{j=1}^{n}(\sum_{b=1}^{\eta} r_{i,j,b} \cdot p^b) \cdot \lambda^{j-1}}. \end{aligned} \tag{5.23}$$

Therefore, the followings hold:

$$(F)^e \cdot C = g^{(\sum_{j=1}^{n}(\sum_{b=1}^{\eta} r_{i,j,b} \cdot p^b) \cdot \lambda^{j-1}) \cdot e + c} = g^{Z_3}. \tag{5.24}$$

- **Soundness.**

  The soundness is proved by showing that the protocol is an argument of knowledge (AoK), and it has a witness-extended emulator. At first, since $\lambda \in \mathbb{Z}_q^*$ is randomly chosen by $\mathcal{V}$, based on Schwartz-Zippel lemma, prover $\mathcal{P}$ has negligible probability of convincing $\mathcal{V}$ unless all λ^{j-1} related variables match on each side of the equality for all $j \in [n]$.

  It is assumed that there exists a PPT witness-extended extractor $\mathcal{E}$ runs $\langle \mathcal{P}^*, \mathcal{V} \rangle$ to get transcripts. In addition, if $\mathcal{P}$ is able to make an acceptable argument, then $\mathcal{E}$ can also succeed with the same probability. $\mathcal{E}$ rewinds the protocol to the first challenge phase (λ) and runs it with fresh challenges until it has n acceptable arguments. More specifically, each time $\alpha \in [n]$, $\mathcal{E}$ first gives new challenge (λ_α) and a challenge $e_{\alpha,1}$, then $\mathcal{E}$ can get:

$$\begin{aligned} Z_{j,1}^{(\alpha,1)} &:= s_{i,j} \cdot e_{\alpha,1} + a_j \ \text{ for } j \in [n], \\ Z_2^{(\alpha,1)} &:= (\sum_{j=1}^{n}(\sum_{k=0}^{t-1} a'_{i,k}) \cdot j^k \cdot \lambda^{j-1}) \cdot e_{\alpha,1} + b, \\ Z_{j,3}^{(\alpha,1)} &:= (\sum_{b=1}^{\eta} r_{i,j,b} \cdot p^b) \cdot e_{\alpha,1} + c_j \ \text{ for } j \in [n]. \end{aligned} \tag{5.25}$$

  Then, $\mathcal{E}$ rewinds the protocol to the second challenge phase feeding new challenge $e_{\alpha,2}$ and gets:

$$\begin{aligned} Z_{j,1}^{(\alpha,2)} &:= s_{i,j} \cdot e_{\alpha,2} + a_j \ \text{ for } j \in [n], \\ Z_2^{(\alpha,2)} &:= (\sum_{j=1}^{n}(\sum_{k=0}^{t-1} a'_{i,k}) \cdot j^k \cdot \lambda^{j-1}) \cdot e_{\alpha,2} + b, \\ Z_{j,3}^{(\alpha,2)} &:= (\sum_{b=1}^{\eta} r_{i,j,b} \cdot p^b) \cdot e_{\alpha,2} + c_j \ \text{ for } j \in [n]. \end{aligned} \tag{5.26}$$

For $j \in [n]$, by computing $(Z_{j,1}^{(\alpha,1)} - Z_{j,1}^{(\alpha,2)})/(e_{\alpha,1} - e_{\alpha,2})$ and $(Z_{j,3}^{(\alpha,1)} - Z_{j,3}^{(\alpha,2)})/(e_{\alpha,1} - e_{\alpha,2})$, $\mathcal{E}$ can get $s_{i,j}$ and $\sum_{b=1}^{\eta} r_{i,j,b} \cdot p^b$. Taking any t elements from $\{s_{i,j}\}_{j=1}^{n}$, $\mathcal{E}$ can reconstruct the polynomial $F_i(z)$, thus it can extract witness $\{a_{i,t}\}_{j=1,t=0}^{n,t-1}$. In addition, witness $\{r_{i,j,b}\}_{b=1,j=1}^{\eta,n}$ can be computed by decoding $\sum_{b=1}^{\eta} r_{i,j,b} \cdot p^b$.

To ease understanding of the proof and simplify the equations, the following is defined:

$$\begin{aligned} J_{i,j} &:= \sum_{k=0}^{t-1} a'_{i,k} \cdot j^k \text{ for } j \in [n], \\ \mathcal{Y} &:= \begin{bmatrix} J_{i,j} & \dots & J_{i,n} \end{bmatrix}. \end{aligned} \tag{5.27}$$

By computing $(Z_2^{(\alpha,1)} - Z_2^{(\alpha,2)})/(e_{\alpha,1} - e_{\alpha,2})$, $\mathcal{E}$ will get

$$\Gamma_\alpha = \mathcal{Y} \cdot \begin{bmatrix} 1 \\ \lambda_\alpha \\ \dots \\ (\lambda_\alpha)^{n-1} \end{bmatrix}.$$

There is overwhelming probability that transcripts with n different challenges, can be generated, and these challenges give a $(n) \times (n)$ invertible transposed Vandermonde polynomial matrix [182]:

$$\Lambda = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \lambda_1 & \lambda_2 & \dots & \lambda_n \\ \vdots & \vdots & \ddots & \vdots \\ (\lambda_1)^{n-1} & (\lambda_2)^{n-1} & \dots & (\lambda_n)^{n-1} \end{pmatrix}.$$

Denote $\Omega := [\Gamma_1, \dots, \Gamma_n]$. Given $\mathcal{Y} \cdot \Lambda = \Omega$, $\mathcal{E}$ can get $J_{i,j} = \sum_{k=0}^{t-1} a'_{i,k} \cdot j^k$ which can be considered as polynomial function $G(x)$ with coefficients $\{a'_{i,k}\}_{k=0}^{t-1}$ evaluated on nodes j. Hence witness $\{a'_{i,k}\}_{k=0}^{t-1}$ can be easily extracted by Lagrange Interpolation.

- **Zero Knowledge.**

  In terms of special honest verifier zero-knowledge, a simulator, S, is constructed to the challenges $\{\lambda, e\} \leftarrow (\mathbb{Z}_q)^{[2]}$ and statement $\{C_{i,k}\}_{k=0}^{t-1}$, $\{A_{i,j,b}, B_{i,j,b}\}_{j=1,b=1}^{n,\eta}$ as inputs, it should output a simulated transcript the distribution of which is indistinguishable from the real one.

  In detail, S firstly picks Z_2 from $\mathbb{Z}_q$, and $\{Z_{j,1}, Z_{j,3}\}_{j=1}^{n} \leftarrow (\mathbb{Z}_q)^{[2 \cdot n]}$, then computes $Z_1, Z_2, D, \{E_j\}_{j=1}^{n}, F$ according to the protocol description, and computes the following:

$$\begin{aligned} A &= g^{Z_1} \cdot \mathsf{ck}^{Z_2} \cdot (D)^{-e}, \\ B_j &= g^{Z_{j,1}} \cdot \mathsf{pk}_j^{Z_{j,3}} \cdot (E_j)^{-e} \text{ for } j \in [n], \\ C &= g^{Z_3} \cdot (F)^{-e}. \end{aligned} \tag{5.28}$$

After that, S outputs the simulated transcripts as follows:

$$\langle A, \{B_j\}_{j=1}^n, C, \{Z_{j,1}\}_{j=1}^n, Z_2, \{Z_{j,3}\}_{j=1}^n \rangle.$$

Since $\{a_j\}_{j=1}^n, b, \{c_j\}_{j=1}^n, \omega, \upsilon$ are uniformly random, the distribution of simulated $\{Z_{j,1}\}_{j=1}^n$, Z_2, $\{Z_{j,3}\}_{j=1}^n$ should also be uniformly random, hence simulated $\{Z_{j,1}\}_{j=1}^n, Z_2, \{Z_{j,3}\}_{j=1}^n$ are identical to the distribution of them in the argument. In addition, due to that $A, \{B_j\}_{j=1}^n, C$ follow the same distribution in real argument, as they are uniquely determined for fixed elements from group $\mathbb{G}$. Therefore, to conclude, simulated transcripts has the same distribution as real transcripts in a real argument.

□

5.5.2 Correct Decryption Proof

In this proof, $\mathcal{P}$ needs prove to $\mathcal{V}$ that $s_{j,i}$ is indeed computed from decrypting ciphertexts $\{A_{j,i,b}, B_{j,i,b}\}_{b=1}^{\eta}$. Given $\mathsf{pk}_s = g^{\mathsf{sk}_i}$, $D_1 := \prod_{d=1}^{\eta}(A_{j,i,b})^{p^b}$ and $D_2 := \prod_{d=1}^{\eta}(B_{j,i,b})^{p^b}$, it's easy to infer that $D_2/(g^{s_{j,i}}) = (D_1)^{\mathsf{sk}_i}$. Therefore, Chaum-Pedersen protocol[49] can be used to prove discrete-logarithm equality, by showing that $\log_g(\mathsf{pk}_i) = \log_{D_1}(D_2/(g^{s_{j,i}}))$. The ZK argument is given in Figure 5.5 and its proof is presented in Theorem 9.

In the first round, $\mathcal{P}$ selects a random w, and computes

$$D_1 := \prod_{d=1}^{\eta}(A_{j,i,b})^{p^b}, D_2 := \prod_{d=1}^{\eta}(B_{j,i,b})^{p^b}, A := g^w, B := D_1^w. \tag{5.29}$$

Then $\mathcal{P}$ sends $\{A, B\}$ to $\mathcal{V}$.

In the second round, $\mathcal{V}$ responses by a challenge e, which can be randomly selected, or computed by hashing $\mathcal{P}$'s messages,

$$e \leftarrow \mathsf{hash}(\langle A, B \rangle). \tag{5.30}$$

In the third round, $\mathcal{P}$ computes and sends Z to $\mathcal{V}$,

$$Z := w + e \cdot \mathsf{sk}_i. \tag{5.31}$$

$\mathcal{V}$ computes the challenge e, and decide if it can accept $\mathcal{P}$'s messages by check if the followings hold:

$$\begin{aligned} g^Z &= A \cdot (\mathsf{pk}_i)^e, \\ {D_1}^Z &= B \cdot (D_2/(g^{s_{j,i}}))^e. \end{aligned} \tag{5.32}$$

Correct Decryption ZK argument

CRS: $\{g, \mathsf{pk}_i\} \in \mathbb{G} \setminus \{1\}$
Statement: $\{A_{j,i,b}, B_{j,i,b}\}_{b=1}^{\eta}, s_{j,i}$
Witness: sk_i
Protocol:

- $\mathcal{P}$ does the following:
  - Compute $D_1 := \prod_{d=1}^{\eta}(A_{j,i,b})^{p^b}$;
  - Compute $D_2 := \prod_{d=1}^{\eta}(B_{j,i,b})^{p^b}$;
  - Select $w \leftarrow \mathbb{Z}_q$;
  - Compute $A := g^w$, $B := D_1^w$;
  - $\mathcal{P} \leftarrow \mathcal{V} : \langle A, B \rangle$.
- $\mathcal{P} \leftarrow \mathcal{V} : e \leftarrow \mathbb{Z}_q$;
  /* For NIZK, set $e \leftarrow \mathsf{hash}(\langle A, B \rangle)$ */
- The prover $\mathcal{P}$ does the following:
  - Compute $Z := w + e \cdot \mathsf{sk}_i$;
  - $\mathcal{V} \leftarrow \mathcal{P} : \langle Z \rangle$.

Verification:

- $\mathcal{V}$ sets $e \leftarrow \mathsf{hash}(\langle A, B \rangle)$ for NIZK;
- $\mathcal{V}$ checks the following:
  - $g^Z = A \cdot (\mathsf{pk}_i)^e$;
  - ${D_1}^Z = B \cdot (D_2/(g^{s_{j,i}}))^e$.

Figure 5.5: Correct Decryption ZK Argument in $\Pi_{\text{DBKG}}^{n,t,m}$.

Theorem 9 (Correct Decryption)**.** *Assume the DDH problem is hard. Let* $D_1 := \prod_{d=1}^{\eta}(A_{j,i,b})^{p^b}$ *and* $D_2 := \prod_{d=1}^{\eta}(B_{j,i,b})^{p^b}$*, the protocol described in Figure 5.5 is an honest verifier zero-knowledge argument of knowledge of* sk_i *such that* $D_2/(g^{s_{j,i}}) = (D_1)^{\mathsf{sk}_i}$.

Proof of Theorem 9.

- **Completeness.**

Set $R := \sum_{d=1}^{\eta}(r_{j,i,b})^{p^b}$, it is easy to compute that

$$\begin{aligned} D_1 &= g^R, \\ D_2 &= g^{s_{j,i}} \cdot (\mathsf{pk}_i)^R. \end{aligned} \tag{5.33}$$

Given that $A := g^w$, $B := D_1^w$, $Z := w + e \cdot \mathsf{sk}_i$, $\mathsf{pk}_i = g^{\mathsf{sk}_i}$, it is easy to infer the following:

$$\begin{aligned} g^Z &= g^{w+e\cdot\mathsf{sk}_i} = A \cdot (\mathsf{pk}_i)^e, \\ {D_1}^Z &= {D_1}^w \cdot g^{(R\cdot e\cdot\mathsf{sk}_i)} = B \cdot (D_2/(g^{s_{j,i}}))^e. \end{aligned} \tag{5.34}$$

- **Soundness.**

  The soundness can be proven by extraction of sk_i by rewinding with different challenges as following:

$$\begin{aligned} Z_1 &:= w + e_1 \cdot \mathsf{sk}_i, \\ Z_2 &:= w + e_2 \cdot \mathsf{sk}_i. \end{aligned} \tag{5.35}$$

  $\mathcal{E}$ can get witness sk_i by computing $(Z_1 - Z_2)/(e_1 - e_2)$.

- **Zero Knowledge.**

  Given challenge $e \leftarrow \mathbb{Z}_q$ and statement $\{A_{j,i,b}, B_{j,i,b}\}_{b=1}^{\eta}, s_{j,i}$, simulator S selects a random Z from $\mathbb{Z}_q$, and outputs $A := g^Z \cdot (\mathsf{pk}_i)^{-e}$ and $B = {D_1}^Z \cdot (D_2/(g^{s_{j,i}}))^{-e}$. As w in the real argument is random, Z in both argument should be random. In addition, A and B follow the same distribution as real transcripts in a real argument.

□

5.6 Security Analysis of HIM based DBKG

Under the UC framework ([20]), the security of protocol is proved by the indistinguishability of ideal world with ideal functionality and real world with real protocol. We give the Theorem 10 for the security analysis of HIM based DBKG.

Theorem 10 (Distributed Batch Key Generation)**.** *Assume* Com *is perfect hiding and computational binding with adversary advantage of* $\mathsf{Adv}^{\mathsf{Binding}}_{\mathsf{Com}}(1^\kappa, \mathcal{A})$. *Assume Correct Sharing* NIZK *and Correct Decryption* NIZK *are perfect complete, perfect special honest verifier zero knowledge, and computational sound with adversary advantage of* $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Sharing}(1^\kappa, \mathcal{A})$ *and* $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Dec}(1^\kappa, \mathcal{A})$. *Assume* Enc *is IND-CPA secure with adversary advantage of* $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A})$. *The protocol* $\Pi^{n,t,m}_{\mathrm{DBKG}}$ *in Figure 5.2 and Figure 5.3 UC-realise* $\mathcal{F}^{n,t,m}_{\mathrm{DBKG}}$ *in Figure 5.1 in* $\{\mathcal{F}_{\mathrm{BC}}\}$*-hybrid world against static corruption up to* $t-1$ *parties with distinguishing advantage upper bounded by*

$$(t-1)\cdot(\mathsf{Adv}^{\mathsf{Binding}}_{\mathsf{Com}}(1^\kappa,\mathcal{A}) + \mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Sharing}(1^\kappa,\mathcal{A}) + \mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Dec}(1^\kappa,\mathcal{A})) + (n-1)\cdot\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa,\mathcal{A}).$$

Proof of Theorem 10.
To prove theorem 10, the first step is to construct a simulator, S, such that no nonuniform PPT environment, $\mathcal{Z}$, can distinguish between ideal world and real world: in ideal world, the ideal execution is $\mathsf{EXEC}_{\mathcal{F}^{n,t,m}_{\mathrm{DBKG}},\mathsf{S},\mathcal{Z}}$ where the parties interact with functionality, $\mathcal{F}^{n,t,m}_{\mathrm{DBKG}}$, in the ideal world and corrupted parties are controlled by the simulator, S; in the real world, the real execution is $\mathsf{EXEC}^{\mathcal{F}_{\mathrm{BC}}}_{\Pi^{n,t,m}_{\mathrm{DBKG}},\mathcal{A},\mathcal{Z}}$, where the parties, $\mathsf{P} = \{P_1,\ldots,P_n\}$, run protocol, $\Pi^{n,t,m}_{\mathrm{DBKG}}$, in the $\{\mathcal{F}_{\mathrm{BC}}\}$-$hybrid\ world$ and the corrupted parties are controlled by a dummy adversary, $\mathcal{A}$, who simply forwards messages from/to $\mathcal{Z}$.

Simulator. The simulator, S, internally runs $\mathcal{A}$, forwarding messages to/from the environment, $\mathcal{Z}$. The simulator, S, simulates the following interactions with $\mathcal{A}$:

- Upon receiving $(\textsc{KeyGenNotify}, \mathsf{sid}, P_i)$ from the ideal functionality $\mathcal{F}^{n,t,m}_{\mathrm{DBKG}}$ about an honest participant P_i, the simulator S does the following:
  - Simulate Correct Sharing NIZK, σ_i;
  - Select $x_i \leftarrow \mathbb{Z}_p$, $\{a_{i,k}, a'_{i,k}\}_{k=0}^{t-1} \leftarrow (\mathbb{Z}_p)^{[2\cdot t]}$ and $\{r_{i,j,b}\}_{j=1,b=1}^{n,\eta} \leftarrow (\mathbb{Z}_p)^{[n\cdot\eta]}$, construct a random degree $t-1$ polynomial

$$F_i(z) := \sum_{k=0}^{t-1} a_{i,k}\cdot z^k; \tag{5.36}$$

  - For $k \in [0, t-1]$, compute

$$C_{i,k} := g^{a_{i,k}} \cdot \mathsf{ck}^{a'_{i,k}}; \tag{5.37}$$

- For $j \in [n]$, compute

$$s_{i,j} := F_i(j), (d_{i,j,1}, \ldots, d_{i,j,\eta}) := \textsc{Encode}(s_{i,j}); \tag{5.38}$$

- For $j \in [n]$, $b \in [\eta]$, compute

$$(A_{i,j,b}, B_{i,j,b}) := \mathsf{Enc}_{\mathsf{pk}_j}(d_{i,j,b}; r_{i,j,b}); \tag{5.39}$$

- Post $\{\{A_{i,j,b}, B_{i,j,b}\}_{j=1,b=1}^{n,\eta}, \sigma_i, \{C_{i,k}\}_{k=0}^{t-1})\}$ to $\mathcal{F}_{\text{BC}}$.

- Once the simulated $\mathcal{F}_{\text{BC}}$ receives $(\{A_{j,i,b}, B_{j,i,b}\}_{i=1,b=1}^{n,\eta}, \sigma_j, \{C_{j,k}\}_{k=0}^{t-1})$ from a corrupted participant, $P_j \in \mathsf{P_c}$, the simulator, S, does the following:
  - If σ_j is invalid, set $\mathsf{QUAL} := [n] \setminus \{j\}$, ignore this message. Otherwise continue to next step;
  - Send $(\mathsf{Gen}, \mathsf{sid}, P_j)$ to ideal functionality, $\mathcal{F}_{\text{DBKG}}^{n,t,m}$;
  - For $P_i \in \mathsf{P_h}$, decrypt $\{A_{j,i,b}, B_{j,i,b}\}_{b\in[\eta]}$ to get $\{d_{j,i,b}\}_{b\in[\eta]}$, and compute

$$s_{j,i} := \textsc{Decode}(d_{j,i,1}, \ldots, d_{j,i,\eta}). \tag{5.40}$$

    If any decryption fails, S aborts;
  - Construct a set $\mathsf{R} \in \mathsf{P_h}$, where $|\mathsf{R}| = t$. Reconstruct $F_j(z)$ by

$$F_j(z) = \sum_{P_f \in \mathsf{R}} \lambda_{j,f} \cdot s_{j,f}, \tag{5.41}$$

    where $\{\lambda_{j,f}\}_{P_f\in\mathsf{R}}$ are Lagrange Coefficients, denote $F_j(z)$ by $F_j(z) = \sum_{k=0}^{t-1} a_{j,k} \cdot z^k$. If $\{s_{j,f}\}_{P_f\in\mathsf{R}}$ fail to lie on the same polynomial, S aborts.

- Once $\mathcal{F}_{\text{BC}}$ gets $\{(\{A_{j,i,b}, B_{j,i,b}\}_{i=1,b=1}^{n,\eta}, \sigma_j, \{C_{j,k}\}_{k=0}^{t-1})\}_{P_j\in\mathsf{P_c}}$ from all corrupted users, the simulator, S does the following:
  - $P_j \in \mathsf{P_c}$ computes

$$\begin{aligned} &D_{j,k} := g^{a_{j,k}} \text{ for } k \in [0, t-1], \\ &s_{i,j} := F_i(j) \text{ for } i \in \mathsf{QUAL}, \\ &\mathsf{psk}_{v,i} := \sum_{i\in\mathsf{QUAL}} \lambda_{v,i} s_{i,j} \text{ for } v \in \mathsf{N}. \end{aligned} \tag{5.42}$$

  - $P_j \in \mathsf{P_c}$ sends $(\textsc{CorruptShares}, \mathsf{sid}, \{j, \{\mathsf{psk}_{v,i}\}_{v\in\mathbf{N}}\}_{P_j\in\mathsf{P_c}})$ to $\mathcal{F}_{\text{DBKG}}^{n,t,m}$;

- Once $\mathcal{F}_{\text{BC}}$ gets $(\{D_{v,k}\}_{k\in[0,t-1]}, P_w)$ about an honest participant P_w against P_v, S simulates Correct Decryption NIZK σ'_w for P_w, and post $(\textsc{Complaint}, s_{v,w}, \sigma'_w)$ to $\mathcal{F}_{\text{BC}}$;

- Once $\mathcal{F}_{\text{BC}}$ gets complaint $(\text{COMPLAINT}, s_{a,k}, \sigma'_k)$ from P_k against P_a, S interpolates P_a's polynomial based on t shares sent to honest participants from P_a. If these shares are not in the same polynomial, S aborts;
- Upon receiving $(\text{READPKRETURN}, \mathsf{sid}, \{\mathsf{gpk}_v\}_{v=1}^{m}, \{\mathsf{ppk}'_{v,i}\}_{v=1,i=1}^{m,n_{r+1}})$from $\mathcal{F}_{\text{DBKG}}^{n,t,m}$, the simulator, S, does the following:
  - Select an honest participant from $\mathsf{P_h}$, $P_f \in \mathsf{P_h}$, and select $D_{f,0} \leftarrow \mathbb{G}$. Note that $\{\mathsf{gpk}_v\}_{v\in\mathbf{V}}$ can be computed by the exponential product of $\mathbf{HIM}$ and $\{D_{j,0}\}_{j=1}^{n}$. Based on the second property of Hyper Invertible Matrix (Cf. Property 2) , given $n-t$ values of $\{\mathsf{gpk}_v\}_{v\in\mathbf{N}}$, $t-1$ values of $\{D_{i,0}\}_{P_i\in\mathsf{P_c}}$, and an additional random value $D_{f,0}$, it's possible to compute $D_{j,0}$ for $P_j \in \mathsf{P_h}$;
  - For $P_i \in \mathsf{P_h}$, denote a polynomial over $\mathbb{G}$ by $G_i(z) := \prod_{k=0}^{t-1}(D_{i,k})^{z^k}$, of which the evaluation on points $\{j\}_{P_j\in\mathsf{P_c}}$ are equal to $\{g^{s_{i,j}}\}_{P_j\in\mathsf{P_c}}$. Based on the value of $D_{j,0}$ and $t-1$ values of $\{g^{s_{i,j}}\}_{P_j\in\mathsf{P_c}}$, S can reconstruct $G_i(\cdot) = \prod_{P_j\in\mathsf{P_c}}(g^{s_{i,j}})^{\lambda_{i,j}} \cdot (D_{i,0})^{\lambda_{i,0}}$. Therefore, S is able to compute $\{D_{i,k}\}_{k=0}^{t-1}$;
  - Post $(\{D_{i,k}\}_{k=0}^{t-1})$ for $P_i \in \mathsf{P_h}$ to $\mathcal{F}_{\text{BC}}$.

Indistinguishability.

The indistinguishability of ideal execution, $\mathsf{EXEC}_{\mathcal{F}_{\text{DBKG}}^{n,t,m},\mathsf{S},\mathcal{Z}}$, and real execution, $\mathsf{EXEC}^{\mathcal{F}_{\text{BC}}}_{\Pi_{\text{DBKG}}^{n,t,m},\mathcal{A},\mathcal{Z}}$, is proven through a series of hybrid worlds $\mathcal{H}_0, \ldots, \mathcal{H}_7$.

Hybrid $\mathcal{H}_0$: It is the real protocol execution $\mathsf{EXEC}^{\mathcal{F}_{\text{BC}}}_{\Pi_{\text{DBKG}}^{n,t,m},\mathcal{A},\mathcal{Z}}$.

Hybrid $\mathcal{H}_1$: $\mathcal{H}_1$ is the same as $\mathcal{H}_0$ except that in $\mathcal{H}_1$, Correct Sharing NIZK, σ_i, sent by a honest party P_i is replaced by simulated proof.
Claim: $\mathcal{H}_1$ and $\mathcal{H}_0$ are perfectly indistinguishable.
Proof: Since NIZK is perfect complete and perfect special honest verifier zero knowledge, if any adversary, $\mathcal{A}$, can distinguish $\mathcal{H}_1$ from $\mathcal{H}_0$, then an adversary, $\mathcal{B}$, can be constructed, who can break the ZK property of NIZK σ_i. ■

Hybrid $\mathcal{H}_2$: $\mathcal{H}_2$ is the same as $\mathcal{H}_1$ except that in $\mathcal{H}_2$, the message, $(\{\{A_{i,j,b}, B_{i,j,b}\}_{j=1,b=1}^{n,\eta})$, sent by the honest user, P_i, is replaced with $(\{\{A'_{i,j,b}, B'_{i,j,b}\}_{j=1,b=1}^{n,\eta})$, where $(A'_{i,j,b}, B'_{i,j,b}) = \mathsf{Enc}_{pk_j}(d'_{i,j,b}; r_{i,j,b})$.
Claim: If the lifted ElGamal encryption scheme is IND-CPA secure with adversarial advantage, $\mathsf{Adv}_{\mathsf{Enc}}^{\mathsf{IND\text{-}CPA}}(1^\kappa, \mathcal{A})$, then $\mathcal{H}_2$ and $\mathcal{H}_1$ are indistinguishable with distinguishing advantage at most $(n-1)\cdot \mathsf{Adv}_{\mathsf{Enc}}^{\mathsf{IND\text{-}CPA}}(1^\kappa, \mathcal{A})$.
Proof: $n-1$ ciphertexts have been changed which encrypted random strings, therefore, if any adversary, $\mathcal{A}$, can distinguish $\mathcal{H}_2$ from $\mathcal{H}_1$, then an adversary, $\mathcal{B}$, can be constructed,

who can break IND-CPA game of Lifted Elgamal encryption scheme. The overall adversary advantage in $\mathcal{H}_2$ is $(n-1) \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A})$. ■

Hybrid $\mathcal{H}_3$: $\mathcal{H}_3$ is the same as $\mathcal{H}_2$ except that in $\mathcal{H}_3$, the message, $(\{C_{i,k}\}_{k=0}^{t-1})$, sent by the honest user, P_i, is replaced with $(\{C'_{i,k}\}_{k=0}^{t-1})\})$, where $C'_{i,k} = \mathsf{Com}_{\mathsf{ck}}(\bar{a}_{i,k}, a''_{i,k})$, and $\{\bar{a}_{i,k}, a''_{i,k}\}$ are randomly from $\mathbb{Z}_q$.
Claim: $\mathcal{H}_3$ and $\mathcal{H}_2$ are perfectly indistinguishable.
Proof: $(\{C'_{i,k}\}_{k=0}^{t-1})\})$ are computed by Pedersen Commitment, due to its perfect hiding property, no adversary can differentiate $(\{C_{i,k}\}_{k=0}^{t-1})$ and $(\{C'_{i,k}\}_{k=0}^{t-1})\})$. ■

Hybrid $\mathcal{H}_4$: $\mathcal{H}_4$ is the same as $\mathcal{H}_3$ except that in $\mathcal{H}_4$, S aborts if the shares sent to honest parties by P_j fail to lie on the same polynomial.
Claim: If NIZK is computational sound with adversary advantage of $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Sharing}(1^\kappa, \mathcal{A})$, $\mathcal{H}_4$ and $\mathcal{H}_3$ are indistinguishable.
Proof: If shares sent by corrupted P_j can pass NIZK but fail to lie on the same polynomial, it means adversary compromise the soundness property of NIZK, it can open NIZK to different witness. In this case, S aborts. Therefore, $\mathcal{H}_4$ and $\mathcal{H}_3$ are indistinguishable with adversary advantage of $(t-1) \cdot \mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Sharing}(1^\kappa, \mathcal{A})$. ■

Hybrid $\mathcal{H}_5$: $\mathcal{H}_5$ is the same as $\mathcal{H}_4$ except that in $\mathcal{H}_5$, the messages $\{D_{i,k}\}_{k=0}^{t-1}$ sent by honest user P_i are backwards calculated from $\{\mathsf{gpk}_v\}_{v \in \mathbf{V}}$ and $\{g^{s_{i,j}}\}_{P_j \in \mathsf{P_c}}$ received from $\mathcal{F}^{n,t,m}_{\mathrm{DBKG}}$.
Claim: If Com is computational binding with adversary advantage of $\mathsf{Adv}^{\mathsf{Binding}}_{\mathsf{Com}}(1^\kappa, \mathcal{A})$, $\mathcal{H}_5$ and $\mathcal{H}_4$ are indistinguishable.
Proof: Firstly, if $D_{i,0}$ posted by corrupted party, P_i, is not the same as the one used to compute $\{\mathsf{gpk}_v\}_{v \in \mathbf{V}}$, then an adversary, $\mathcal{A}$, can be constructed to break the computationally binding property of Pedersen Commitment. Secondly, the distribution of $\{D_{i,k}\}_{k=0}^{t-1}$ in $\mathcal{H}_5$ have identical distribution to $\{D_{i,k}\}_{k=0}^{t-1}$ in $\mathcal{H}_4$ if $\{D_{i,k}\}_{k=0}^{t-1}$ are fixed. Due to the $t-1$ commitments from corrupted parties in $\{\mathsf{gpk}_v\}_{v \in \mathbf{V}}$, the adversary advantage in $\mathcal{H}_5$ is $(t-1) \cdot \mathsf{Adv}^{\mathsf{Binding}}_{\mathsf{Com}}(1^\kappa, \mathcal{A})$. ■

Hybrid $\mathcal{H}_6$: $\mathcal{H}_6$ is the same as $\mathcal{H}_5$ except that in $\mathcal{H}_6$, Correct Decryption NIZK, σ'_w, sent by honest user, P_w, is replaced by a simulated proof when there is a complaint.
Claim: $\mathcal{H}_6$ and $\mathcal{H}_5$ are perfectly indistinguishable.
Proof: Since NIZK is perfect complete and perfect special honest verifier zero knowledge, if any adversary, $\mathcal{A}$, can distinguish $\mathcal{H}_6$ from $\mathcal{H}_5$, then an adversary, $\mathcal{B}$, can be constructed, who can break the ZK property of NIZK σ'_w. ■

Hybrid $\mathcal{H}_7$: $\mathcal{H}_7$ is the same as $\mathcal{H}_6$ except that in $\mathcal{H}_7$, S aborts if it finds shares sent to honest parties from P_a are not in the same polynomial about a complaint, $(\textsc{Complaint}, s_{a,k}, \sigma'_k)$.

Claim: If Correct Decryption NIZK is computational sound with adversary advantage of $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Dec}(1^\kappa, \mathcal{A})$, $\mathcal{H}_7$ and $\mathcal{H}_6$ are indistinguishable.
Proof: If shares sent by corrupted P_a can pass NIZK but fail to lie on the same polynomial, it means adversary compromise the soundness property of NIZK, it can open NIZK to different witness. In this case, S aborts. Therefore, $\mathcal{H}_7$ and $\mathcal{H}_6$ are indistinguishable with adversary advantage of $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Dec}(1^\kappa, \mathcal{A})$. ■

The adversary's view of $\mathcal{H}_7$ is identical to the simulated view $\mathsf{EXEC}_{\mathcal{F}^{n,t,m}_{\mathrm{DBKG}},\mathsf{S},\mathcal{Z}}$. Therefore, no PPT$\mathcal{Z}$ can distinguish the view of the ideal execution from the view of the real execution with more than advantage upper bounded by

$$(t-1)\cdot(\mathsf{Adv}^{\mathsf{Binding}}_{\mathsf{Com}}(1^\kappa, \mathcal{A}) + \mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Sharing}(1^\kappa, \mathcal{A}) + \mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Dec}(1^\kappa, \mathcal{A})) + (n-1)\cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A}).$$

This concluded the proof of Theorem 10. □

5.7 Summary

In this chapter, we presented an overview of Distributed Key Generation (DKG) protocols and reviewed the existing DKG protocols based on their communication and computation complexity. We discussed prior research that explored DKG protocols under different security and communication assumptions. From this analysis, we identified the need for a Batched Distributed Key Generation (DBKG) protocol to generate multiple keys simultaneously, which served as the motivation for our proposed work.

We introduced the new Distributed Batch Key Generation ideal functionality and provided the details of our proposed DBKG protocol. The DBKG protocol utilises two non-interactive zero-knowledge proofs to verify the sharing and decryption steps, employing Σ protocols. To ensure the security of DBKG, we analysed its security under the Universal Composability (UC) framework, comparing the indistinguishability between the ideal world and the real world.

The proposed batch DKG protocol is designed to be highly flexible, capable of supporting a large number of participants. The protocol's effectiveness in accommodating various participant numbers is influenced by factors such as network conditions and the computational capabilities of individual participants. Importantly, while it can efficiently manage groups ranging from hundreds to thousands, the efficiency is maintained through advanced cryptographic algorithms and optimised message handling processes. These enhancements ensure that the workload and message complexity for each participant remain manageable.

Transitioning to the concept of scalability in our protocol, it is important to note that our definition diverges from traditional system research perspectives. Typically, scalability in systems research is associated with the capability to handle an increased workload by adding more resources, such as servers in a horizontally scalable architecture. However, in the context of our cryptographic protocol, scalability pertains to the capacity to generate a greater number of cryptographic keys without a proportional increase in message exchanges. This aspect of scalability is vital for cryptographic efficiency and practical applicability in large-scale environments.

To achieve this, the scalability of our batch DKG protocol is realised by optimising the protocol to ensure that the increase in the number of keys generated does not linearly increase the communication costs. We leverage advanced cryptographic techniques that facilitate batch processing of operations and aggregation of proofs. This significantly reduces the per-key overhead in both computation and communication. Such an approach ensures that our system can efficiently scale to support the generation and management of multiple cryptographic keys simultaneously, making it particularly suitable for applications that require rapid and concurrent generation and management of cryptographic keys.

In the next chapter, we will demonstrate how the DBKG protocol plays a crucial role in a two-stage voting scheme, where the ballots are encrypted using public keys generated by

the DBKG protocol.

Chapter 6

Building Block: Two Stage Voting Scheme

The enemy knows the system.

Claude Shannon, Kerckhoffs's principle

6.1 Overview

Electronic voting (E-voting, [183, 186, 187, 188, 189, 190, 191, 192, 193, 184, 185]) enables voters to cast their ballots remotely through the Internet. In comparison to traditional paper-based voting, E-voting eliminates the need for paper printing and in-person ballot submission, leading to faster ballot counting and promoting ecological sustainability. The convenience of E-voting has been shown to positively impact voter turnout ([194]). As a powerful technology, E-voting is considered the foundation of E-government for elections in various countries, such as Estonia, Brazil, and the US.

An E-voting scheme should satisfy the following essential requirements ([111, 195, 196, 191, 83, 193]):

- **Privacy:** In a secure E-voting scheme, a voter's preference should remain confidential and not be deducible unless the adversary manipulates the voter or colludes with other voters. The only exception is when all voters unanimously agree on a decision, in which case only the final tally result is revealed.
- **Integrity:**It should be practically impossible for any participant to modify the ballots without detection.
- **Correctness:** The final tally result only include the valid and unique ballots.

- **End-to-end verifiability**
  - **Individual Verifiability:** Voters can verify if their ballots are included in the final tally result;
  - **Universal Verifiability:** Anyone can verify the final tally result corresponds to the ballots;
  - **Eligibility Verifiability:** Only registered and valid voters can submit ballots, each voter can only submit one ballot. Anyone can verify that if the finally tally result comes from the valid ballots from valid voters.
- **Fault tolerance:** The voting system is resilient to up to a number of malicious participants while guaranteeing the liveness and safeness of the whole system.

To fulfill the mentioned requirements, various cryptographic protocols have been employed in E-voting systems. Current E-voting schemes can be classified into four categories:

- **Mix-Net-based E-voting.**

  Mix-Net ([197]) is a routing protocol. In mix-net-based E-voting, mix-nets are utilised to shuffle and anonymise the encrypted ballots, which enables anonymous hard-to-trace communication and breaks the link between voters and their ballots. Each mix-net server takes encrypted ballots as input and produces permuted encrypted ballots as output. This process ensures voter privacy by making it difficult to trace the original ballot to a specific voter.

  The pioneering E-voting scheme was introduced by Chaum [197] and was based on a decryption mix-net protocol using RSA encryption. In this protocol, the sender encrypts the message using onion encryption, where multiple layers of encryption are applied to the message. The first mixer in the network then decrypts the outer layer of the ciphertext, shuffles the ciphertext, and forwards it to the next mixer. This process is repeated by each mixer until all mixers have processed the ciphertext.

  Chaum's E-voting scheme provides a level of privacy as long as at least one mixer in the network remains honest. This means that even if some mixers collude to compromise voter privacy, the presence of at least one honest mixer will prevent them from linking individual votes to specific voters. The use of RSA encryption ensures that the mixers can manipulate the ciphertext without gaining access to the actual vote choices, maintaining the secrecy of the ballots.

  The concept of onion encryption, where each mixer only decrypts one layer of encryption, adds an additional layer of security to the E-voting scheme. This design prevents any individual mixer from learning the complete voting information and reinforces the privacy of the voting process.

[198] proposed a re-encryption mix-net protocol based on Elgamal encryption, where ciphertexts from the senders are re-encrypted by mixers. This approach offers significant advantages over Chaum's scheme, including increased efficiency, robustness, and flexibility. Unlike Chaum's onion encryption, re-encryption mix-nets require senders to encrypt their messages only once, simplifying the process and making it more lightweight.

Further improvements to mix-net based E-voting were made by [199], aiming at achieve voter verifiability in addition to privacy. However, their protocol relies on an anonymous channel, which may introduce challenges in practical implementations. To address this, [200] introduced confirmation numbers in a revised-simplified verifiable re-encryption mix-net to enhance the overall security.

In another direction, [201] optimised Helios 1.0 and proposed a mix-net protocol with end-to-end verifiability. This improvement streamlines the integrity proof for mix-nets and accelerates the computation involved in the mixing process. [202] made a significant contribution to the field of mix-net-based E-voting by implementing a proof of correct shuffle using the Coq proof assistant. This implementation represents a crucial milestone as it is the first to be machine-checked for cryptographic correctness, ensuring the integrity of the shuffle process.

Despite these advancements, mix-net-based E-voting schemes have certain drawbacks. They require significant computational power and complexity in the shuffle and decryption phases, making them less suitable for large-scale voting scenarios. Additionally, mix-net-based schemes are vulnerable to DDoS attacks. If any mixer fails during the voting process, it can disrupt the entire election, posing a considerable risk to the integrity of the voting system.

- **Blind Signature based E-voting:**

  In blind signature-based E-voting schemes, blind signature schemes are employed to ensure voter privacy. This approach enables voters' ballots to be blindly signed by the election authority, preserving the anonymity of their choices. Blind signatures, initially introduced by Chaum [203] and further developed by Camenisch [204], allow an authority to sign a disguised message from a sender without knowledge of the unblinded message. The resulting blind signature provides public verifiability against the original message. In a blind signature-based E-voting system, voters can blind their ballots, while the voting authority can still validate the votes without access to the actual content of the ballots. This cryptographic technique ensures confidentiality and privacy in the E-voting process.

  Several blind signature-based E-voting schemes have been proposed in the literature. [81] was pioneer in implementing such a scheme, but their protocol required voters to participate until the tally phase terminates. Kumar *et al.*[205] introduced a blind

signature-based E-voting scheme with identity-based signatures, leveraging voters' unique identification. On the other hand, [206] integrated blind signature-based E-voting with a Kerberos authentication method, but this approach was found to be susceptible to DoS attacks. These blind signature-based E-voting schemes offer confidentiality and anonymity for voters, but some of them exhibit limitations in terms of participation requirements and vulnerability to specific attacks.

While blind signature-based E-voting schemes offer anonymity without using zero-knowledge proofs, many existing approaches involve high computational costs to manage certificates. For instance, [207] proposed a multi-authority and coercion-resistant scheme based on fake credentials for coercion exit, mix-nets, and blind signatures. However, this scheme requires significant computational resources, as it involves computing mix-nets, tokens, and RSA keys for each voter.

Overall, blind signature-based E-voting schemes provide a promising avenue for ensuring voter privacy, but their widespread adoption may be constrained by the computational overhead associated with certificate management in some existing protocols.

- **Homomorphic E-voting:**

  Homomorphic encryption is employed in homomorphic E-voting schemes to facilitate the aggregation of encrypted ballots without the need for decryption. The homomorphic properties of certain encryption schemes, such as Elgamal encryption [35], Paillier encryption [208], and RSA encryption [209], allow specific computations to be performed directly on ciphertexts without revealing their underlying plaintext values. This characteristic makes homomorphic encryption an ideal solution for the ballots counting phase of E-voting schemes [210, 211, 212], as it enables the computation of the tally result on encrypted data while preserving both privacy and integrity.

  Several homomorphic E-voting schemes have been proposed in the literature. Cohen [213] introduced the first homomorphic E-voting scheme for elections, but this protocol failed to protect the privacy of voters' choices. [211] built upon Cohen's work and proposed the first practical receipt-free voting scheme. However, their protocol assumes the existence of a one-way secure channel between the authority and voters. [214] proposed an end-to-end E-voting scheme based on Elgamal encryption without any setup assumptions or the use of a random oracle. To reduce the workload in the tallying process, [215] proposed using a homomorphic signcryption scheme in E-voting, where the number of verifications is fixed to the number of candidates rather than the number of voters. These homomorphic E-voting schemes leverage the properties of homomorphic encryption to ensure secure and private vote aggregation without the need for decrypting individual ballots.

- **Blockchain-based based E-voting:**

  In blockchain-based E-voting, a distributed ledger (blockchain) is utilised to securely record and store encrypted ballots, ensuring transparency, integrity, and immutability of the voting process. E-voting schemes that aim for verifiability often assume the existence of a public bulletin board [215], which is trusted to provide a consistent view to all participants. To eliminate the reliance on a trusted bulletin board, several blockchain-based E-voting schemes have been proposed to enable verifiable E-voting ([216, 192, 191, 217]). These schemes leverage the security properties of blockchain technology to achieve end-to-end verifiability and transparency in the voting process.

  One notable example is the Open Vote Network (OVN) proposed by [218], which is the first self-tallying blockchain-based E-voting system built on the Ethereum blockchain. OVN has been further optimised by [219] to enhance scalability by off-chaining some computations. However, this introduces a challenge as the off-chain computation is not validated, leaving room for potential manipulation if a voter claims incorrect computation results to be true. To address these concerns, [220] proposed an end-to-end verifiable E-voting scheme on a private blockchain, utilising threshold cryptography for fault tolerance.

  Other blockchain-based E-voting schemes such as [221, 222, 223, 13] have also been proposed with different security features and cryptographic techniques. For example, [221] leveraged voter-verified audit trails for enhanced auditability and verifiability. [222] proposed a self-tallying voting protocol in a decentralised IoT environment based on timed commitment. BroncoVote [223] used homomorphic encryption to guarantee the privacy of vote on blockchain. Yet their protocol introduces centralisation as it requires a trusted server to compute off-chain operation, which is vulnerable for single-point-of-failure. [13] proposed a UC-secure voting scheme on blockchain for blockchain treasury management, in their scheme, voters and experts need to vote on all the proposals and encrypt the ballots with a distributed key generation protocol from [38].

  Overall, blockchain-based E-voting continues to be an active area of research, with ongoing efforts to address challenges and enhance the security and efficiency of the voting process in decentralised settings.

In this chapter, we introduce a novel Two Stage Voting (TSV) Scheme (Figure 6.1), which serves as a critical component of the proposed decision-making system presented in Chapter 4. The TSV Scheme is based on blockchain technology and aims to address practical challenges observed in one-stage voting systems. Traditional E-voting schemes typically involve a single stage where voters must go through all the candidates and determine their preferences. However, in real-world scenarios with numerous candidates, some voters may opt for convenience and only vote for a limited number of candidates, neglecting the rest.

To address this issue and promote thoughtful voting in the decision-making process, we propose a two-stage approach. The first stage, called the Preferential Voting stage, allows both voters and experts to rank candidates with different weights or scores, utilising the Borda Count voting method [118]. This stage generates a shortlist of candidates based on the preferences of voters and experts. In the second stage, known as the Threshold Voting stage, each voter and expert casts one of three possible votes for each candidate in the shortlist: YES, NO, ABSTAIN.

Both stages of the TSV Scheme support delegative voting, enabling voters to delegate their decision-making authority to one or more experts. This fosters better collaborative intelligence and democracy, as voters can rely on the expertise of selected individuals in the decision-making process.

Overall, the Two Stage Voting Scheme aims to reduce voting efforts for participants while ensuring a more thoughtful and comprehensive voting process. By breaking down the voting process into two stages and incorporating blockchain technology, the TSV Scheme enhances the efficiency, transparency, and inclusivity of the decision-making system.

In developing the TSV Scheme, our primary objective was to enhance the thoughtfulness and depth of the voting process, particularly in scenarios involving numerous candidates. While the TSV Scheme effectively addresses these concerns by breaking the voting process into two distinct stages-Preferential and Threshold Voting-it is not the only conceivable method to mitigate unthoughtful voting. However, it was chosen for its particular strengths in promoting detailed voter engagement and facilitating a comprehensive evaluation of candidates.

Borda Count is utilised in the first stage of our voting system to allow voters to rank candidates in order of preference, assigning a point scale that reflects the intensity of their preferences. This method simplifies the aggregation process, as each vote translates directly into points based on rank, providing a clear, quantitative measure of voter sentiment. This approach is highly compatible with blockchain technology, where transactions (in this case, votes) are immutable and verifiable. The direct translation of rankings to points in Borda Count also facilitates a straightforward calculation that enhances transparency and auditability, essential characteristics in blockchain applications.

In the second stage, Threshold Voting is employed to allow a focused decision-making process among the top candidates identified in the first stage. This method involves voters casting a straightforward vote for each of the shortlisted candidates. Threshold Voting simplifies the final decision process, ensuring that only candidates who achieve a certain level of consensus are elected, which enhances the decisiveness and legitimacy of the election outcomes.

Prior to settling on the TSV Scheme, we considered several other methods that could potentially encourage more thoughtful voting. For instance, methods such as cumulative voting [224] and ranked-choice voting [225] were evaluated for their ability to require voters to think more critically about their choices.

Cumulative Voting was considered less suitable for our system due to its potential to encourage strategic voting, where voters might allocate all their votes to a single candidate to maximize their influence. This can skew the results in favor of more polarizing candidates and does not necessarily provide a balanced view of voter preferences across a wider range of candidates. Furthermore, the management of multiple vote allocations per voter can introduce additional complexity in vote tallying on a blockchain, potentially increasing the cost and time of transaction processing.

Ranked-Choice Voting, while advantageous in reducing the risk of wasted votes in traditional systems by reallocating votes from eliminated candidates, introduces significant computational complexity. In blockchain systems, where each transaction must be processed and recorded individually, the iterative vote redistribution and tallying required by RCV can be both time-consuming and expensive. Additionally, RCV's iterative nature could complicate the verification process, making it less transparent and harder to audit compared to direct methods like Borda Count and Threshold Voting.

The TSV Scheme was specifically chosen because it not only encourages thoughtful voting by having voters engage with the candidates in a more focused manner during the Preferential Voting stage but also allows for a simplified and decisive selection process in the Threshold Voting stage. This two-stage approach is particularly well-suited for blockchain implementation, where transaction efficiency and verifiable transparency are paramount. Moreover, the incorporation of the Borda Count and the simplistic yes-no-abstain decision in the second stage effectively balances depth of choice with operational efficiency, making it ideal for our proposed decision-making system.

Thus, while the TSV Scheme is our chosen approach for the reasons outlined, it is part of a broader landscape of voting methodologies that could potentially address the issue of unthoughtful voting. The design choice for the TSV Scheme reflects a balance of our specific goals: enhancing voter engagement, maintaining scalability, and ensuring the integrity and transparency of the voting process within a blockchain environment.

TSV Scheme ensures end-to-end verifiability without relying on a trusted tallying authority. All the ballots from experts and voters are submitted to the blockchain in an encrypted form, along with zero-knowledge proofs to demonstrate that these ballots are generated honestly and correctly without revealing the original secret votes. Since the ballots are stored on the blockchain, they are tamper-resistant and cannot be altered or deleted by any party. Any participant can verify the correctness of the encrypted ballots and the accompanying zero-knowledge proofs.

The TSV Scheme allows both experts and voters to submit multiple ballots, but only the most recent one will be processed for each participant. To ensure fault tolerance, a voting committee is responsible for generating distributed keys used for encrypting the ballots and computing the tally results. If the majority of the voting committee members remain honest, the TSV Scheme guarantees the security and privacy of voters' and experts' ballots. A detailed analysis of the honesty of the voting committee will be provided in this chapter.

Additionally, the voting committee is required to submit non-interactive zero-knowledge (NIZK) proofs during the jointly decryption process. This ensures that anyone with access to the blockchain can verify that all the encrypted ballots are correctly counted in the final tally results, leveraging the homomorphic property of Elgamal encryption. By employing these cryptographic techniques and the transparency of the blockchain, the TSV Scheme achieves a robust and verifiable E-voting process, fostering trust and confidence in the decision-making system.

We compared our voting scheme with the existing voting schemes, in terms of basic security requirements including privacy, fairness, end-to-end (E2E) verifiability, and new properties including universal composability (UC) security, flexibility and 2-stage voting. All the voting schemes guarantee ballots privacy and end-to-end verifiability, some of the schemes cannot guarantee fairness which gives voters additional advantage. For example, Yu etal. [102] introduced a single voting administrator to trigger and reveal tally, which breaks fairness if it reveals partial tally to some voters. We check if the schemes are proved under the UC framework and find that only [13, 226] are universal composable. The comparison results in in Table 6.1 shows that our voting scheme is the only one that provides UC security and flexible 2-stage voting to save voting efforts and improve voting efficiency besides satisfying all the security properties.

In detail, we examine how our voting scheme satisfies the design properties:

- **Privacy:** Voters' ballots are encrypted by Lift Elgamal encryption with public keys generated by voting committee. Based on DDH assumption, it is infeasible to infer the original message from the ciphertexts. Moreover, during the tally phase, voting committee members compute tally based on additively homomorphic property without decrypting ballots. If majority of the voting committee members are uncorrupted, ballots privacy is guaranteed;

- **Fairness:** In the pre-voting epoch, the final proposals are made public by deadline based on a two-stage project proposing procedure, which separates submission of proposal commitment from revealing the proposals on blockchain. Therefore, proposers cannot know other proposals in advance based on the hiding property of commitment. In the voting epoch, voters and experts should submit encrypted ballots together with zero-knowledge proofs. Because of DDH assumption, no one can infer original messages from ciphertexts, therefore voters and experts cannot change their ballots by counting on others' outputs. Additionally, each ballot contains a NIZK proof, even if some party directly copy-pastes and randomises others' ballots to avoid duplication, it cannot provide valid NIZK proof. Moreover, the finally tally results are only revealed at the end of voting epoch, voters and experts cannot change their decisions after seeing the final tally results;

- **Efficiency and Flexibility:** In the pre-voting epoch, we use a commitment based two-stage project proposing procedure to avoid advantage over late submission and

guarantee fairness, which improves the proposal submission efficiency. In the voting epoch, we introduce a new two-stage voting to save voters and experts' voting effort, which improves overall voting efficiency. Besides, we propose a new DBKG protocol to generate distributed keys with amortised communication cost of $\mathcal{O}(n)$. Moreover, our Handover protocol supports changing voting committee members flexibly in each round;

- **End-to-end verifiability:**

  Individual Verifiability. Voters and experts can verify if their ballots are recorded on blockchain, which is guaranteed by the immutability, traceability and auditability of blockchain. As mentioned before, the honesty of an untrusted voting device can be assured by cryptographic techniques such as Benaloh challenge [227, 228] and the protocol proposed in [229]. In addition, voters and experts can validate the correctness of tally results and get decrypted delegated voting power of all experts by checking NIZK proofs. Therefore, they can check if the correct encrypted tally results contain their ballots by additively homomorphically computing based on all the encrypted ballots, voting power of voters and delegated voting power.

  Universal Verifiability. Everyone can check the messages posted on blockchain to verify fairness of proposal submission and voting, including proposal commitment, encrypted ballots, and final tally results. Based on the encrypted ballots, voters' voting power, experts' delegated voting power, public keys, decryption shares and final tally results on blockchain, everyone can verify correctness of final tally result.

  Eligibility Verifiability. To participant voting, voters and experts are required to lock stakes on blockchain and submit encrypted ballots to blockchain. As all transactions on blockchain are signed by the sender's secret key, everyone can check if the final tally contains ballots from valid parties together with universal verifiability.

Table 6.1: Comparison with other voting schemes.

Schemes	[230]	[231]	[87]	[232]	[218]	[102]	[233]	[13]	[200]	[234]	[235]	[236]	[237]	[226]	[222]	[238]	[239]	[90]	Ours
Privacy	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Fairness	✓	✗	✗	✗	✓	✗	✗	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
E2E Verifiability	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
UC Security	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
Flexibility	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
2-stage Voting	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

This chapter provides the construction and security modelling of the TSV scheme, organised as follows:

- Section 6.2 details the construction of the first stage, Preferential Voting. The Preferential Voting Functionality $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ is introduced, defining the required functionality for this stage. Moreover, the Preferential Voting Protocol $\Pi_{\text{VOTE1}}^{c,\mu,s,n}$ is presented, demonstrating how it realises the voting functionality $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}$ in the first stage. Within $\Pi_{\text{VOTE1}}^{c,\mu,s,n}$, various zero-knowledge proofs are employed, including the Batched 0 or 1 Encryption Proof, which is used for the Valid Ballot Proof for experts and voters in the Preferential Voting Protocol. Additionally, the Unit Vector Proof from [13] is utilised for experts' and voters' non-interactive zero-knowledge (NIZK) proofs in both stages. Furthermore, the Valid Ballot Proof is described, which verifies the authenticity of experts' and voters' ballots in the Preferential Voting process. The security analysis of Preferential Voting is performed within UC framework.

- Section 6.3 outlines the construction of the second stage, Threshold Voting. It introduces the Threshold Voting functionality $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$, which defines the necessary functionality for this stage. Furthermore, it presents the Threshold Voting Protocol $\Pi_{\text{VOTE2}}^{c,\mu,s,n}$, demonstrating how it realises the Threshold Voting functionality $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$. The security analysis of Threshold Voting is conducted within UC framework.

Overall, this chapter provides a comprehensive understanding of the TSV scheme's construction, its constituent stages, and the security measures implemented to ensure the integrity and privacy of the voting process.

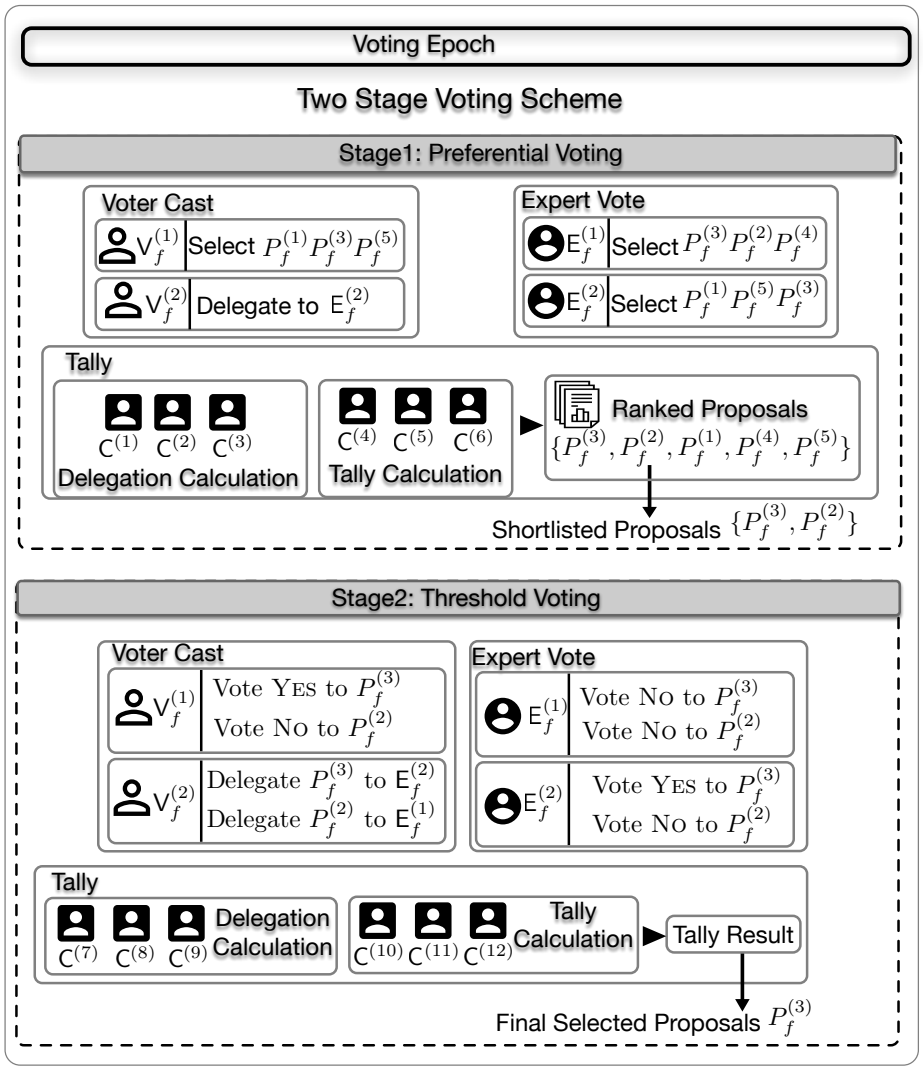

Figure 6.1: Two Stage Voting Scheme Example.

6.2 Preferential Voting Construction

This section presents the construction of Preferential Voting in the voting epoch. In Section 6.2.1, we design the Preferential Voting Functionality, denoted as $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$, with UC framework. Subsequently, Section 6.2.2 introduces the Preferential Voting Protocol, denoted as $\Pi_{\text{VOTE1}}^{c,\mu,s,n}$, which *UC-realises* $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ under the assumption of static corruption. The zero-knowledge proofs utilised in $\Pi_{\text{VOTE1}}^{c,\mu,s,n}$ are elucidated in Section 6.2.3. Additionally, we provide a security analysis of Preferential Voting, demonstrating the indistinguishability of its real-world execution with $\Pi_{\text{VOTE1}}^{c,\mu,s,n}$ and its ideal-world execution with $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$.

6.2.1 Preferential Voting Functionality $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$

The Voting functionality, denoted as $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$, is designed to encompass various tasks required for the operation of Preferential Voting. These tasks include initialisation, voter casting, expert voting, and tallying. $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ interacts with voters represented as $\mathcal{V}_{\mathsf{fld}}^{[v]} := \{\mathsf{V}_{\mathsf{fld}}^{(i)}\}_{i=1}^{v}$, experts represented as $\mathcal{E}_{\mathsf{fld}}^{[e]} := \{\mathsf{E}_{\mathsf{fld}}^{(j)}\}_{j=1}^{e}$, the voting committee represented as $\mathcal{C}^{[c]} := \{\mathsf{C}^{(t)}\}_{t=1}^{c}$ (with a threshold of μ, implying that the number of corrupted voting committee members should be less than μ), and the adversary represented as S.

The Voting functionality, $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$, is parameterised three committee flag sets: $\mathbf{C}_{key}$, $\mathbf{C}_{del}$, and $\mathbf{C}_{tally}$, which are initially set to $\emptyset$. Additionally, there are valid voter casting set, $\mathbf{V}$, and valid expert voting set, $\mathbf{E}$, which are also initially empty. The functionality requires a delegation calculation algorithm, denoted as DelAlg_1, and a tally algorithm, denoted as $\mathsf{TallyAlg}_1$. The set of corrupted voting committee members is represented as $\mathcal{C}\mathsf{cor}$, and the set of honest voting committee members is represented as $\mathcal{C}_{\mathsf{honest}}$. The parameters n and s indicate the number of proposal lists generated in the pre-voting epoch and the size of the shortlist, respectively.

$\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ works as follows:

- **Initialisation Phase**. To initiate the voting process, a voting committee member $\mathsf{C}^{(t)}$ sends a message $(\text{INIT}, \mathsf{sid})$ to the functionality $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$. Upon receiving this message, $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ notifies the adversary S by sending $(\text{INITNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$. The voting process commences, and $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ proceeds to the next step only when all voting committee members have sent their initialisation messages. At each step, $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ updates the set $\mathbf{C}key$ by including the voting committee member $\mathsf{C}^{(t)}$. The process continues until the cardinality of $\mathbf{C}key$ becomes equal to c.

- **Voter Cast Phase**. A voter $\mathsf{V}_{\mathsf{fld}}^{(i)} \in \mathcal{V}_{\mathsf{fld}}^{[v]}$ participates in the voting process by sending its ballots $\mathbf{a_i}^{[s+1]}$ and voting power η_i to the functionality $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ using the message $(\text{CAST}, \mathsf{sid}, \mathbf{a_i}^{[s+1]}, \eta_i)$. Here, $\mathbf{a_i}^{[s+1]}$ represents the voter's choices for selecting the shortlisted proposals and its delegation choices. For example, in Figure 4.3-(a), $\mathbf{a_1}^{[4]}$ is $\{P_f^{(1)}, P_f^{(1)}, P_f^{(1)}, \perp\}$ for $\mathsf{V}_f^{(1)}$, and $\mathbf{a_2}^{[4]}$ is $\{\perp, \perp, \perp, \mathsf{E}_f^{(1)}\}$ for $\mathsf{V}_f^{(2)}$.

Upon receiving the message, $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ stores it in the set $\mathbf{V}$ as $(\mathsf{V}^{(i)}\mathsf{fld}, \mathbf{a}_i^{[s+1]}, \eta_i)$ and notifies the adversary S by sending $(\text{CASTNOTIFY}, \mathsf{V}_{\mathsf{fld}}^{(i)}, \mathsf{sid}, \eta_i)$. If more than μ voting committee members are corrupted ($|\mathcal{C}\mathsf{cor}| \geq \mu$), $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ also leaks the voter's inputs to S by sending $(\text{LEAK}, \mathsf{V}_{\mathsf{fld}}^{(i)}, \text{CAST}, \mathsf{sid}, \mathbf{a}_i^{[s+1]}, \eta_i)$.

- **Expert Vote Phase**. An expert $\mathsf{E}_{\mathsf{fld}}^{(j)} \in \mathcal{E}_{\mathsf{fld}}^{[e]}$ participates in the voting process by sending its ballots $\mathbf{b_j}^{[s]}$ to $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ using the message $(\text{VOTE}, \mathsf{sid}, \mathbf{b_j}^{[s]})$, where $\mathbf{b_i}$ represents the expert's choices for the shortlisted proposals. For example, in Figure 4.3-(a), $\mathbf{b_1}^{[3]}$ is $\{P_f^{(3)}, P_f^{(2)}, P_f^{(4)}\}$ for $\mathsf{Exp}_f^{(1)}$.

  Upon receiving the message, $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ stores it in the set $\mathbf{E}$ as $(\mathsf{E}_{\mathsf{fld}}^{(j)}, \mathbf{b_j}^{[s]})$, and notifies the adversary S by sending $(\text{VOTENOTIFY}, \mathsf{E}_{\mathsf{fld}}^{(j)}, \mathsf{sid})$. If more than μ voting committee members are corrupted ($|\mathcal{C}_{\mathsf{cor}}| \geq \mu$), $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ additionally leaks the expert's inputs to S.

- **Tally Phase**.

  - **Delegation Computation**. $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ first computes the delegation power of each expert when it receives the command $(\text{CALDEL}, \mathsf{sid})$ from a voting committee member $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$. It updates the set of committee members involved in delegation, $\mathbf{C}_{del}$, by setting $\mathbf{C}_{del} := \mathbf{C}_{del} \cup \mathsf{C}^{(t)}$ and then sends $(\text{CALDELNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ to S.

    Once there are more than μ voting committee members involved in delegation ($|\mathbf{C}_{del}| \geq \mu$), $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ can compute the delegation power by evaluating the function $\mathsf{DelAlg}_1(n, s, e, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v})$ (Figure 6.3), which yields the delegation powers $\{D_j\}_{j=1}^{e}$ for the experts:

    $$\{D_j\}_{j=1}^{e} \leftarrow \mathsf{DelAlg}_1(n, s, e, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v}). \tag{6.1}$$

    If the number of corrupted committee members involved in delegation exceeds μ ($|\mathbf{C}_{del} \cap \mathcal{C}_{\mathsf{cor}}| \geq \mu$), experts' delegation powers are revealed to S.

  - **Tally Computation**. $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ begins to compute the tally results of each proposal and notifies S by sending $(\text{TALLYNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ once it receives $(\text{TALLY}, \mathsf{sid}, \mathbf{T})$ from a voting committee member $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$. It updates the set of committee members involved in tallying, $\mathbf{C}_{tally}$, by setting $\mathbf{C}_{tally} := \mathbf{C}_{tally} \cup \mathsf{C}^{(t)}$. When the number of voting committee members involved in tallying exceeds μ ($|\mathbf{C}_{tally}| \geq \mu$) and at least μ of them send $(\text{TALLY}, \mathsf{sid}, \mathbf{T})$, $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ calculates the tally results of each proposal using the function $\mathsf{TallyAlg}_1$ in Figure 6.4, which yields the tally results $\{\mathbf{f_l}\}_{l=1}^{n}$:

    $$\{\mathbf{f_l}\}_{l=1}^{n} \leftarrow \mathsf{TallyAlg}_1(n, s, \gamma, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v}, \{\mathbf{b_j}^{[s]}, D_j\}_{j=1}^{e}, \mathbf{T}). \tag{6.2}$$

    If the number of corrupted committee members involved in tallying exceeds μ ($|C_{tally} \cap \mathcal{C}_{\mathsf{cor}}| \geq \mu$), the tally results are revealed to S.

- Any party can read the tally results and experts' decisions by sending $(\text{READTALLY}, \mathsf{sid})$ and $(\text{REVEAL}, \mathsf{sid}, \mathsf{E}_{\mathsf{fld}}^{(j)})$ to $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$, $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ responds to these requests by returning messages $(\text{READTALLYRETURN}, \mathsf{sid}, \{\mathbf{f_l}\}_{l=1}^{n})$ and $(\text{REVEALEXPERT}, \mathsf{sid}, \mathbf{b_j}^{[s]})$ to the requester.

Preferential Voting Ideal Functionality $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$

$\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ interacts with voters $\mathcal{V}_{\mathsf{fld}}^{[v]} := \{\mathsf{V}_{\mathsf{fld}}^{(i)}\}_{i=1}^{v}$, experts $\mathcal{E}_{\mathsf{fld}}^{[e]} := \{\mathsf{E}_{\mathsf{fld}}^{(j)}\}_{j=1}^{e}$, voting committee $\mathcal{C}^{[c]} := \{\mathsf{C}^{(t)}\}_{t=1}^{c}$ of which the threshold is μ, and adversary S. $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ is parameterised with three committee flag sets $\mathbf{C}_{key}$, $\mathbf{C}_{del}$, $\mathbf{C}_{tally}$, a valid voter casting set $\mathbf{V}$, a valid expert voting set $\mathbf{E}$ which are all set to $\emptyset$ initially, a delegation calculation algorithm DelAlg_1, a tally algorithm $\mathsf{TallyAlg}_1$, corrupted voting committee $\mathcal{C}_{\mathsf{cor}}$, honest voting committee $\mathcal{C}_{\mathsf{honest}}$, the number of proposal list generated in pre-voting epoch n, and the size of shortlist s ($s \leq n$ in preferential voting, $s = 1$ in threshold voting).
$\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ does the following:

Initialisation Phase:

- Upon receiving $(\text{INIT}, \mathsf{sid})$ from a voting committee member, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, send $(\text{INITNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ to S, and set $\mathbf{C}_{key} := \mathbf{C}_{key} \cup \{\mathsf{C}^{(t)}\}$, continue to next phase until $|\mathbf{C}_{key}| = c$.

Voter Cast Phase:

- Upon receiving $(\text{CAST}, \mathsf{sid}, \mathbf{a_i}^{[s+1]}, \eta_i)$ from a voter, $\mathsf{V}_{\mathsf{fld}}^{(i)} \in \mathcal{V}_{\mathsf{fld}}^{[v]}$, set $\mathbf{V} := \mathbf{V} \cup \{(\mathsf{V}_{\mathsf{fld}}^{(i)}, \mathbf{a}_i^{[s+1]}, \eta_i)\}$, and send $(\text{CASTNOTIFY}, \mathsf{V}_{\mathsf{fld}}^{(i)}, \mathsf{sid}, \eta_i)$ to S. Send $(\text{LEAK}, \mathsf{V}_{\mathsf{fld}}^{(i)}, \text{CAST}, \mathsf{sid}, \mathbf{a}_i^{[s+1]}, \eta_i)$ to S if $|\mathcal{C}_{\mathsf{cor}}| \geq \mu$.

Expert Vote Phase:

- Upon receiving $(\text{VOTE}, \mathsf{sid}, \mathbf{b_j}^{[s]})$ from an expert, $\mathsf{E}_{\mathsf{fld}}^{(j)} \in \mathcal{E}_{\mathsf{fld}}^{[e]}$, set $\mathbf{E} := \mathbf{E} \cup \{(\mathsf{E}_{\mathsf{fld}}^{(j)}, \mathbf{b_j}^{[s]})\}$, and send $(\text{VOTENOTIFY}, \mathsf{E}_{\mathsf{fld}}^{(j)}, \mathsf{sid})$ to S. Send $(\text{LEAK}, \mathsf{E}_{\mathsf{fld}}^{(j)}, \text{VOTE}, \mathsf{sid}, \mathbf{b_j}^{[s]})$ to S if $|\mathcal{C}_{\mathsf{cor}}| \geq \mu$.

Tally Phase:

- Upon receiving $(\text{CALDEL}, \mathsf{sid})$ from a voting committee member, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, does the following:
  - Set $\mathbf{C}_{del} := \mathbf{C}_{del} \cup \{\mathsf{C}^{(t)}\}$, send $(\text{CALDELNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ to S;
  - If $|\mathbf{C}_{del}| \geq \mu$, compute $\{\mathbf{D_j}\}_{j=1}^{e} \leftarrow \mathsf{DelAlg}_1(n, s, e, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v})$, Cf. Figure 6.3. if $|\mathbf{C}_{del} \cap \mathcal{C}_{\mathsf{cor}}| \geq \mu$, send $(\text{LEAKDEL}, \mathsf{sid}, \{\mathbf{D_j}\}_{j=1}^{e})$ to S.
- Upon receiving $(\text{TALLY}, \mathsf{sid}, \mathbf{T})$ from a voting committee member $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, does the following:
  - Set $\mathbf{C}_{tally} := \mathbf{C}_{tally} \cup \{\mathsf{C}^{(t)}\}$, send $(\text{TALLYNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ to S;
  - If $|\mathbf{C}_{tally}| \geq \mu$, compute $\{\mathbf{f_l}\}_{l=1}^{n} \leftarrow \mathsf{TallyAlg}_1(n, s, \gamma, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v}, \{\mathbf{b_j}^{[s]}, D_j\}_{j=1}^{e}, \mathbf{T})$, Cf. Figure 6.4. If $|C_{tally} \cap \mathcal{C}_{\mathsf{cor}}| \geq \mu$, send $(\text{LEAKCASTING}, \mathsf{sid}, \{\mathbf{f_l}\}_{l=1}^{n})$ to S.
- Upon receiving $(\text{READTALLY}, \mathsf{sid})$ from any party, returns $(\text{READTALLYRETURN}, \mathsf{sid}, \{\mathbf{f_l}\}_{l=1}^{n})$ to the requester;
- Upon receiving $(\text{REVEAL}, \mathsf{sid}, \mathsf{E}_{\mathsf{fld}}^{(j)})$ from any party, return $(\text{REVEALEXPERT}, \mathsf{sid}, \mathbf{b_j}^{[s]})$.

Figure 6.2: The ideal functionality $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$.

In the Preferential Voting functionality $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$, two algorithms are utilised: the Delegation Calculation Algorithm DelAlg_1 and the Tally Calculation Algorithm $\mathsf{TallyAlg}_1$.

- **Delegation Calculation Algorithm,** $\mathsf{DelAlg}_1(n, s, e, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v})$**.**

  The Delegation Calculation Algorithm DelAlg_1 is responsible for computing the delegation power of experts based on voters' ballots and voters' voting power.

  As illustrated in Figure 6.3, $\mathsf{DelAlg}_1(n, s, e, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v})$ computes the delegation power of all experts based on voters' ballots and their voting power. It takes the number of proposal list generated in pre-voting epoch n, the size of shortlist s, the number of experts e, voters' ballots and voting power $\{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v}$ as inputs. The output of DelAlg_1 is the delegation power for each expert, denoted as $D_j{}_{j=1}^{e}$.

  For $i \in [v]$, voter's ballot, $\mathbf{a_i}^{[s+1]}$, is parsed to s shortlist vectors with size n denoted by $\{v_{i,l,k}\}_{l=1,k=1}^{s,n}$ and a delegation vector with size e denoted by $\{v'_{i,j}\}_{j=1}^{e}$.

  Experts' delegation power can be directly computed based the delegation vector by

$$D_j := \sum_{i=1}^{v} v'_{i,j} \cdot \eta_i. \tag{6.3}$$

Algorithm $\mathsf{DelAlg}_1(n, s, e, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v})$

Input:

- The number of proposal list generated in pre-voting epoch, n;
- The size of shortlist, s;
- The number of experts, e;
- The voters' ballots and voting power, $\{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v}$.

Delegation Calculation:

- For $i \in [v]$, parse $\mathbf{a}_i^{[s+1]}$ to $(\{v_{i,l,k}\}_{l=1,k=1}^{s,n}, \{v'_{i,j}\}_{j=1}^{e})$;
- For $j \in [e]$, compute $D_j := \sum_{i=1}^{v} v'_{i,j} \cdot \eta_i$.

Output:

- Experts' delegation power: $\{D_j\}_{j=1}^{e}$.

Figure 6.3: Delegation Calculation Algorithm in Preferential Voting.

- **Tally Calculation Algorithm,** $\mathsf{TallyAlg}_1(n, s, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v}, \{\mathbf{b_j}^{[s]}, D_j\}_{j=1}^{e}, \mathbf{T})$**.**

The Tally Calculation Algorithm $\mathsf{TallyAlg}_1$ is responsible for computing the tally results based on voters' ballots and voters' voting power, experts' ballots, and experts' delegation power.

As shown in Figure 6.4, Tally calculation algorithm computes the tally results of each proposal, $\{\mathbf{f_k}\}_{k=1}^{n}$, based on voters' and experts' ballots. It takes the number of proposal list generated in pre-voting epoch n, the size of the shortlist s, voters' ballots and voting power, $\{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v}$, experts' ballots and voting power, $\{\mathbf{b_j}^{[s]}, D_j\}_{j=1}^{e}$, the weight of each selected proposals in Borda Count Voting, $\mathbf{T} := \{\tau_l\}_{l=1}^{s}$, as inputs. The output of $\mathsf{TallyAlg}_1$ is tally result (Borda Count Voting points) of each proposal, denoted as $\{f_k\}_{k=1}^{n}$.

To address the issue where wealthier voters might disproportionately influence decision-making processes due to their greater locked stakes, a sigmoid function is utilised to modify the original tally result distribution. This mathematical transformation is designed to moderate the rate at which voting power escalates, ensuring that the increase in influence progresses at a decreasing rate. This approach aims to provide a more equitable representation by reducing the extent to which financial resources can affect voting outcomes.

In $\mathsf{TallyAlg}_1(n, s, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v}, \{\mathbf{b_j}^{[s]}, D_j\}_{j=1}^{e}, \mathbf{T})$, $\mathbf{T} := \{\tau_l\}_{l=1}^{s}$ is the weight set defined by Borda Count Voting. For example, $\mathbf{T} := \{s, \ldots, 1\}$ means that the top 1 ranking proposal gets s points and the last one gets 1 point. Voter's ballot, $\mathbf{a_i}^{[s+1]}$, is parsed to s shortlist vectors with size, n, denoted by $\{v_{i,l,k}\}_{l=1,k=1}^{s,n}$, and a delegation vector with size, e, denoted by $\{v'_{i,j}\}_{j=1}^{e}$ for $i \in [v]$. Expert's ballot, $\mathbf{b_j}^{[s]}$, is parsed to s shortlist vectors with size, n, denoted by $\{p_{j,l,k}\}_{l=1,k=1}^{s,n}$ for $j \in [e]$. For $k \in [n]$, denote smoothing factor as ϵ, the tally result (point) of each proposal is computed by

$$f_k := \frac{1}{1+\epsilon^{-D_{j,l}}} \cdot \left((\sum_{i=1}^{v} (\sum_{l=1}^{s} v_{i,l,k} \cdot \tau_l) \cdot \eta_i) + (\sum_{j=1}^{e} (\sum_{l=1}^{s} p_{j,l,k} \cdot \tau_l) \cdot D_j)) \right). \quad (6.4)$$

Algorithm $\mathsf{TallyAlg}_1(n, s, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v}, \{\mathbf{b_j}^{[s]}, D_j\}_{j=1}^{e}, \mathbf{T})$

Input:

- The number of proposal list generated in pre-voting epoch n;
- The size of shortlist s;
- Voters' ballots and voting power $\{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v}$;
- Experts' ballots and voting power $\{\mathbf{b_j}^{[s]}, D_j\}_{j=1}^{e}$;
- The weight of each selected proposals $\mathbf{T} := \{\tau_l\}_{l=1}^{s}$.
- Smoothing factor ϵ.

Tally Calculation:

- Parse $\mathbf{a}_i^{[s+1]}$ to $(\{v_{i,l,k}\}_{l=1,k=1}^{s,n}, \{v'_{i,j}\}_{j=1}^{e})$ for $i \in [v]$;
- Parse $\mathbf{b_j}^{[s]}$ to $(\{p_{j,l,k}\}_{l=1,k=1}^{s,n})$ for $j \in [e]$;
- For $k \in [n]$, compute
  $f_k := \frac{1}{1+\epsilon^{-D_{j,l}}} \cdot \Big((\sum_{i=1}^{v}(\sum_{l=1}^{s} v_{i,l,k} \cdot \tau_l) \cdot \eta_i) + (\sum_{j=1}^{e}(\sum_{l=1}^{s} p_{j,l,k} \cdot \tau_l) \cdot D_j))\Big)$.

Output:

- Tally results of each proposal: $\{f_k\}_{k=1}^{n}$.

Figure 6.4: Tally Calculation Algorithm in Preferential Voting.

6.2.2 Preferential Voting Protocol $\Pi_{\mathbf{VOTE}1}^{c,\mu,s,n}$

Let $\mathsf{TallyAlg}_1$ be short for $\mathsf{TallyAlg}_1(n, s, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v}, \{\mathbf{b_j}^{[s]}, D_j\}_{j=1}^{e}, \mathbf{T})$, and DelAlg_1 be short for $\mathsf{DelAlg}_1(n, s, e, \{\mathbf{a_i}^{[s+1]}, \eta_i\}_{i=1}^{v})$. Figure 6.7 and Figure 6.8 present Preferential Voting protocol, $\Pi_{\text{VOTE1}}^{c,\mu,s,n}$, to realise $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$ (Figure 6.2) in $\{\mathcal{F}_{\text{BC}}, \mathcal{F}_{\text{DBKG}}^{c,\mu,1}\}$-*hybrid world*.

The voting process consists of four phases: the Initialisation Phase, Voter Cast Phase, Expert Vote Phase, and Tally Phase, as described below:

- **Initialisation Phase**.

  To initiate the voting, voting committee members, denoted as $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, start the key generation process. They generate their partial secret keys, which will be used to encrypt ballots in the Voter Cast Phase and Expert Vote Phase. Each voting committee member sends two messages, $(\text{KEYGEN}, \mathsf{sid}, \mathsf{C}^{(t)})$ and $(\text{READKEYSHARE}, \mathsf{sid}, \mathsf{C}^{(t)})$, to the Key Generation functionality $\mathcal{F}_{\text{DBKG}}^{c,\mu,1}$ when it receives the command $(\text{INIT}, \mathsf{sid})$ from the environment $\mathcal{Z}$. Subsequently, $\mathcal{F}_{\text{DBKG}}^{c,\mu,1}$ returns the partial secret key back to $\mathsf{C}^{(t)}$ by sending $(\text{READKEYSHARERETURN}, \mathsf{sid}, \mathsf{psk}_i)$.

- **Voter Cast Phase**.

  When $\mathcal{Z}$ sends $(\text{CAST}, \text{sid}, \mathbf{a}_i^{[s+1]}, \eta_i)$ to a voter, $\mathsf{V}_{\mathsf{fld}}^{(i)} \in \mathcal{V}_{\mathsf{fld}}^{[v]}$, it casts ballot based on $\mathbf{a}_i^{[s+1]}$. Firstly, $\mathsf{V}^{(i)}$ sends $(\text{READPK}, \text{sid})$ to $\mathcal{F}_{\text{DBKG}}^{c,\mu,1}$, which returns the global public key, and partial public keys of voting committee members, $(\text{READPKRETURN}, \mathsf{gpk}, \{\mathsf{ppk}_a\}_{a=1}^{c})$. Next, $\mathsf{V}^{(i)}$ parses $\mathbf{a}_i^{[s+1]}$ to $(\{\mathbf{v_{i,k}}^{[n]}\}_{k=1}^{s}, \mathbf{v'_i}^{[e]})$, where $\{\mathbf{v_{i,k}}^{[n]}\}_{k=1}^{s}$ are s shortlist vectors with size n and $\mathbf{v'_i}^{[e]}$ is a delegation vector with size e. $\mathsf{V}_{\mathsf{fld}}^{(i)}$ encrypts all ballots with the global public key, gpk, generated by voting committee, $\mathcal{C}^{[c]}$, and outputs ciphertexts:

$$\begin{aligned}(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]}) &:= \mathsf{LEG.Enc}_{\mathsf{gpk}}(\mathbf{v_{i,k}}^{[n]}; \mathbf{r_{i,k}}^{[n]}) \text{ for } k \in [s], \\ (\mathbf{B_i}^{[e]}, \mathbf{B'_i}^{[e]}) &:= \mathsf{LEG.Enc}_{\mathsf{gpk}}(\mathbf{v'_i}^{[e]}; \mathbf{r'_i}^{[e]}),\end{aligned} \tag{6.5}$$

  where $\{\mathbf{r_{i,k}}^{[n]}\}_{k=1}^{s}$ and $\mathbf{r'_i}^{[e]}$ are randomly selected. Additionally, $\mathsf{V}_{\mathsf{fld}}^{(i)}$ generates a Valid Ballot NIZK proof, σ_i, (Cf. Section 6.2.3.3) to show its ballot is valid:

  - For $k \in [s]$, $(\mathbf{A_{i,k}}^{[n]} || \mathbf{B_i}^{[e]}, \mathbf{A'_{i,k}}^{[n]} || \mathbf{B'_i}^{[e]})$ encrypts a unit vector ([13]), in which only one element is 1, the rest are 0;
  - For $l \in [n]$, $(\prod_{k=1}^{s} A_{i,k,l}, \prod_{k=1}^{s} A'_{i,k,l})$ encrypts either 0 or 1, where $\{(A_{i,k,l}, A'_{i,k,l})\}_{k=1,l=1}^{s,n} := (\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})$.

  A simple example is given in Figure 6.5 to show how the ballots should be proved. Afterwards, $\mathsf{V}_{\mathsf{fld}}^{(i)}$ posts $\{(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})\}_{k=1}^{s}$, $(\mathbf{B_i}^{[e]}, \mathbf{B'_i}^{[e]})$ and σ_i to $\mathcal{F}_{\text{BC}}$.

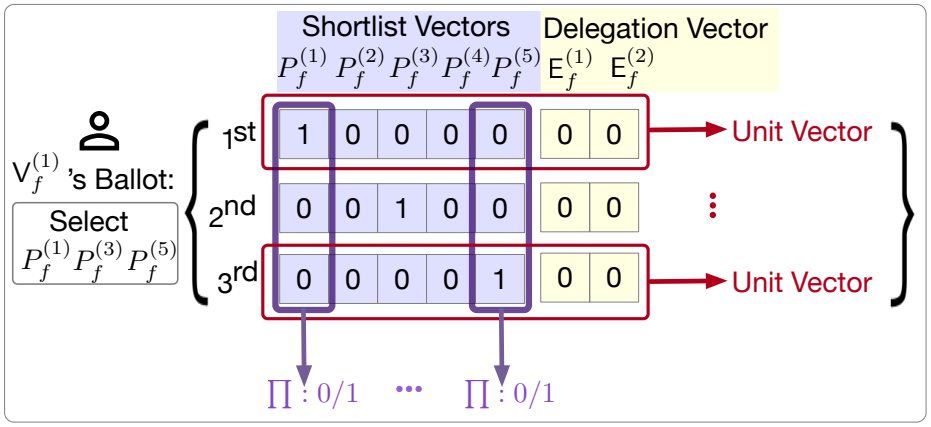

Figure 6.5: Voter's Ballot NIZK Example ($n = 5, s = 3, e = 2$) in Preferential Voting.

- **Expert Vote Phase**.

An expert, $\mathsf{E}_{\mathsf{fld}}^{(j)} \in \mathcal{E}_{\mathsf{fld}}^{[e]}$, begins to vote when it gets $(\textsc{Vote}, \mathsf{sid}, \mathbf{b_j}^{[s]})$ from $\mathcal{Z}$. $\mathsf{E}_{\mathsf{fld}}^{(j)}$ gets public key to encrypt its ballot by requesting $(\textsc{ReadPK}, \mathsf{sid})$ from $\mathcal{F}_{\text{DBKG}}^{c,\mu,1}$, and receives $(\textsc{ReadPKReturn}, \mathsf{gpk}, \{\mathsf{ppk}_a\}_{a=1}^{c})$. Afterwards, $\mathsf{E}_{\mathsf{fld}}^{(j)}$ parses $\mathbf{b_j}^{[s]}$ to $\{\mathbf{p_{j,k}}^{[n]}\}_{k=1}^{s}$ and encrypts the ballots with gpk,

$$(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]}) := \mathsf{LEG.Enc}_{\mathsf{gpk}}(\mathbf{p_{j,k}}^{[n]}; \mathbf{r_{j,k}}^{[n]}) \text{ for } k \in [s]. \tag{6.6}$$

After, $\mathsf{E}_{\mathsf{fld}}^{(j)}$ generates a Valid Ballot NIZK proof, δ_j, to show its ballot is valid:

- For $k \in [s]$, $(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})$ encrypts a unit vector;
- For $k \in [s]$, $(\prod_{k=1}^{s} K_{j,k,l}, \prod_{k=1}^{s} K'_{j,k,l})$ encrypts either 0 or 1 for $l \in [n]$, where $\{(K_{j,k,l}, K'_{j,k,l})\}_{l=1}^{n} := (\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})$.

Figure 6.6 gives an example to show how to prove expert' ballots. Lastly, $\mathsf{E}_{\mathsf{fld}}^{(j)}$ posts the encrypted ballot, $\{\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]}\}$, and NIZK proof, δ_j, to $\mathcal{F}_{\text{BC}}$.

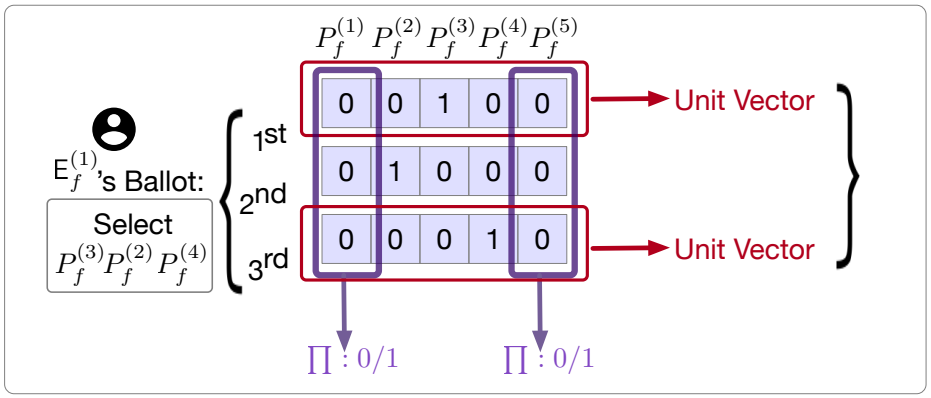

Figure 6.6: Expert's Ballot NIZK Example ($n = 5$, $s = 3$) in Preferential Voting.

- **Tally Phase**.

  In this phase, voting committee compute the delegation power of experts and jointly compute the tally results. Voting committee member, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, begins to compute the delegation power of each expert once it gets $(\textsc{CalDel}, \mathsf{sid})$ from $\mathcal{Z}$. First, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$ gets all the encrypted ballots sent by experts and voters,

$$\{(\{(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})\}_{k=1}^{s}, (\mathbf{B_i}^{[e]}, \mathbf{B'_i}^{[e]}), \sigma_i, \eta_i)\}_{i=1}^{v},$$
$$\{(\{(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})\}_{k=1}^{s}, \delta_j)\}_{j=1}^{e}.$$

Afterwards, it checks the NIZK proofs, removes all the invalid or repeated ballots, and renames the rest as

$$\{(\{(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})\}_{k=1}^{s}, (\mathbf{B_i}^{[\beta]}, \mathbf{B'_i}^{[\beta]}), \sigma_i, \eta_i)\}_{i=1}^{\alpha},$$
$$\{(\{(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})\}_{k=1}^{s}, \delta_j)\}_{j=1}^{\beta}.$$

– **Delegation Calculation**.
$\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$ computes encrypted delegation power based on additively homomorphic property,

$$\begin{aligned} \mathbf{I}^{[\beta]} &:= \prod_{i=1}^{\alpha} (\mathbf{B_i}^{[\beta]})^{\eta_i}, \\ \mathbf{I'}^{[\beta]} &:= \prod_{i=1}^{\alpha} (\mathbf{B'_i}^{[\beta]})^{\eta_i}. \end{aligned} \tag{6.7}$$

The whole voting committee can jointly decrypt $(\mathbf{I}^{[\beta]}, \mathbf{I'}^{[\beta]})$ and get experts' delegation power, $\{m_j\}_{j=1}^{\beta}$ by

$$\{m_j\}_{j=1}^{\beta} := \mathsf{LEG.Dec}(\mathbf{I}^{[\beta]}, \mathbf{I'}^{[\beta]}) \tag{6.8}$$

– **Tally Calculation.**
When $\mathcal{Z}$ sends $(\textsc{Tally}, \mathsf{sid})$ to a voting committee member, $\mathsf{C}^{(t)}$, it can compute the tally results and reveal it to public. For $l \in [n]$, $\mathsf{C}^{(t)}$ computes

$$\begin{aligned} S_l &:= \prod_{i=1}^{\alpha} \prod_{k=1}^{s} (\mathbf{A_{i,k}})^{\tau_k \cdot \eta_i} \cdot \prod_{j=1}^{\beta} \prod_{k=1}^{s} (\mathbf{K_{j,k}})^{\tau_k \cdot m_j}, \\ S'_l &:= \prod_{i=1}^{\alpha} \prod_{k=1}^{s} (\mathbf{A'_{i,k}})^{\tau_k \cdot \eta_i} \cdot \prod_{j=1}^{\beta} \prod_{k=1}^{s} (\mathbf{K'_{j,k}})^{\tau_k \cdot m_j}. \end{aligned} \tag{6.9}$$

Voting committee, $\mathcal{C}^{[c]}$, jointly decrypt (S_l, S'_l) to f_l for $l \in [n]$ by

$$f_l := \mathsf{LEG.Dec}(S_l, S'_l). \tag{6.10}$$

Afterwards, $\mathcal{C}^{[c]}$ post $(\{f_l\}_{l=1}^{n})$ to $\mathcal{F}_{\text{BC}}$.

In the voting functionality $\mathcal{F}_{\text{VOTE1}}^{c,\mu,s,n}$ (6.2), experts' choices are made public so that voters can make better delegation decisions based on experts' voting histories. In the protocol, when the environment $\mathcal{Z}$ sends the message $(\textsc{Reveal}, \mathsf{sid}, \mathsf{E}_{\mathsf{fld}}^{(j)})$ to an expert $\mathsf{E}_{\mathsf{fld}}^{(j)} \in \mathcal{E}_{\mathsf{fld}}^{[e]}$, the expert posts the randomnesses used to encrypt its ballots and its ballots, denoted as $\{\mathbf{r_{j,k}}^{[n]}, \mathbf{p_{j,k}}^{[n]}\}_{k=1}^{s}$, to the blockchain functionality $\mathcal{F}_{\text{BC}}$. Subsequently, the expert returns the message $(\textsc{RevealExpert}, \mathsf{sid}, \{\mathbf{r_{j,k}}^{[n]}, \mathbf{p_{j,k}}^{[n]}\}_{k=1}^{s})$ to $\mathcal{Z}$. The

validation of experts' ballots can be checked based on the randomness they revealed and the ciphertexts they posted before.

Moreover, when $\mathcal{Z}$ sends $(\text{READTALLY}, \mathsf{sid})$ to some party, P, to read the tally results, it returns $(\text{READTALLYRETURN}, \mathsf{sid}, (\{f_l\}_{l=1}^{n})$ to $\mathcal{Z}$ after fetching $(\{f_l\}_{l=1}^{n})$. The final shortlist is generated based on the tally result, and other conditions, such as funding asked by each proposal and the total budget in a blockchain funding decision-making system.

Stage 1: Preferential Voting Protocol $\Pi^{c,\mu,s,n}_{\text{VOTE1}}$ (Part 1)

Initialisation Phase:

- Upon receiving $(\text{INIT}, \mathsf{sid})$ from $\mathcal{Z}$, the voting committee member, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, sends $(\text{KEYGEN}, \mathsf{sid}, \mathsf{C}^{(t)})$ and $(\text{READKEYSHARE}, \mathsf{sid}, \mathsf{C}^{(t)})$ to $\mathcal{F}^{c,\mu,1}_{\text{DBKG}}$, then receives $(\text{READKEYSHARERETURN}, \mathsf{sid}, \mathsf{psk}_i)$.

Voter Cast Phase:

- Upon receiving $(\text{CAST}, \mathsf{sid}, \mathbf{a}_i^{[s+1]}, \eta_i)$ from $\mathcal{Z}$, the voter, $\mathsf{V}^{(i)}_{\mathsf{fld}} \in \mathcal{V}^{[v]}_{\mathsf{fld}}$, does the following:
  - Send $(\text{READPK}, \mathsf{sid})$ to $\mathcal{F}^{c,\mu,1}_{\text{DBKG}}$ and receive $(\text{READPKRETURN}, \mathsf{gpk}, \{\mathsf{ppk}_a\}^c_{a=1})$;
  - Parse $\mathbf{a}_i^{[s+1]}$ to $(\{\mathbf{v_{i,k}}^{[n]}\}^s_{k=1}, \mathbf{v_i'}^{[e]})$;
  - Select $\{\mathbf{r_{i,k}}^{[n]}\}^s_{k=1} \leftarrow (\mathbb{Z}_q)^{[n\cdot s]}$, and $\mathbf{r_i'}^{[e]} \leftarrow (\mathbb{Z}_q)^{[e]}$;
  - For $k \in [s]$, compute $(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]}) := \mathsf{LEG.Enc}_{\mathsf{gpk}}(\mathbf{v_{i,k}}^{[n]}; \mathbf{r_{i,k}}^{[n]})$;
  - Compute $(\mathbf{B_i}^{[e]}, \mathbf{B_i'}^{[e]}) := \mathsf{LEG.Enc}_{\mathsf{gpk}}(\mathbf{v_i'}^{[e]}; \mathbf{r_i'}^{[e]})$;
  - Generate Valid Casting NIZK proof, σ_i:
    - For $k \in [s]$, $(\mathbf{A_{i,k}}^{[n]} || \mathbf{B_i}^{[e]}, \mathbf{A'_{i,k}}^{[n]} || \mathbf{B_i'}^{[e]})$ encrypts a unit vector;
    - For $l \in [n]$, $(\prod^s_{k=1} A_{i,k,l}, \prod^s_{k=1} A'_{i,k,l})$ encrypts either 0 or 1, where $\{(A_{i,k,l}, A'_{i,k,l})\}^{s,n}_{k=1,l=1} := (\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})$.
  - Send $(\mathsf{Write}, \mathsf{sid}, (\{(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})\}^s_{k=1}, (\mathbf{B_i}^{[e]}, \mathbf{B_i'}^{[e]}), \sigma_i, \eta_i))$ to $\mathcal{F}_{\text{BC}}$.

Expert Vote Phase:

- Upon receiving $(\text{VOTE}, \mathsf{sid}, \mathbf{b_j}^{[s]})$ from $\mathcal{Z}$, the expert, $\mathsf{E}^{(j)}_{\mathsf{fld}} \in \mathcal{E}^{[e]}_{\mathsf{fld}}$, does the following:
  - Send $(\text{READPK}, \mathsf{sid})$ to $\mathcal{F}^{c,\mu,1}_{\text{DBKG}}$, and receive $(\text{READPKRETURN}, \mathsf{gpk}, \{\mathsf{ppk}_a\}^c_{a=1})$;
  - Parse $\mathbf{b_j}^{[s]}$ to $(\{\mathbf{p_{j,k}}^{[n]}\}^s_{k=1})$;
  - Select $\{\mathbf{r_{j,k}}^{[n]}\}^s_{k=1} \leftarrow (\mathbb{Z}_q)^{[n*s]}$;
  - For $k \in [s]$, compute $(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]}) := \mathsf{LEG.Enc}_{\mathsf{gpk}}(\mathbf{p_{j,k}}^{[n]}; \mathbf{r_{j,k}}^{[n]})$;
  - Generate Valid Voting NIZK proof, δ_j:
    - For $k \in [s]$, $(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})$ encrypts a unit vector;
    - For $k \in [s]$, $(\prod^s_{k=1} K_{j,k,l}, \prod^s_{k=1} K'_{j,k,l})$ encrypts either 0 or 1 for $l \in [n]$, where $\{(K_{j,k,l}, K'_{j,k,l})\}^n_{l=1} := (\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})$.
  - Send $(\mathsf{Write}, \mathsf{sid}, (\{(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})\}^s_{k=1}, \delta_j))$ to $\mathcal{F}_{\text{BC}}$.

Figure 6.7: Stage 1: Preferential Voting protocol $\Pi^{c,\mu,s,n}_{\text{VOTE1}}$ in $\{\mathcal{F}_{\text{BC}}, \mathcal{F}^{c,\mu,1}_{\text{DBKG}}\}$-*hybrid world* (Part 1).

Stage 1: Preferential Voting Protocol $\Pi^{c,\mu,s,n}_{\text{VOTE1}}$ (Part 2)

Tally Phase:

- Upon receiving $(\text{CALDEL}, \mathsf{sid})$ from $\mathcal{Z}$, the voting committee member, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, does the following:
  - Send $(\mathsf{Read}, \mathsf{sid})$ to $\mathcal{F}_{\text{BC}}$, get $\{(\{(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})\}^{s}_{k=1}, (\mathbf{B_i}^{[e]}, \mathbf{B'_i}^{[e]}), \sigma_i, \eta_i)\}^{v}_{i=1}$ and $\{(\{(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})\}^{s}_{k=1}, \delta_j)\}^{e}_{j=1}$;
  - Check if $\mathsf{Verify}(\{(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})\}^{s}_{k=1}, (\mathbf{B_i}^{[e]}, \mathbf{B'_i}^{[e]}), \sigma_i) = 1$ for $i \in [v]$, remove all the invalid casting ballots. If there are repeated ciphertexts in $\{(\{(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})\}^{s}_{k=1}, (\mathbf{B_i}^{[e]}, \mathbf{B'_i}^{[e]}))\}^{v}_{i=1}$, remove all the repeated casting ballots except the first one sent to $\mathcal{F}_{\text{BC}}$. Set $\mathsf{VI}^{[\alpha]}_{\mathsf{fld}}$ as a set of voter index in new ascending order who provides valid ballots;
  - Check if $\mathsf{Verify}(\{(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})\}^{s}_{k=1}, \delta_j) = 1$ for $j \in [e]$, remove all the invalid voting ballots. If there are repeated ciphertexts in $\{(\{(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})\}^{s}_{k=1})\}^{e}_{j=1}$, remove all the repeated voting ballots except the first one sent to $\mathcal{F}_{\text{BC}}$. Set $\mathsf{EI}^{[\beta]}_{\mathsf{fld}}$ as a set of voter index in new ascending order who provides valid ballots;
  - Remove the ciphertexts sent by experts/voters, and sent to invalid experts, denote the rest ciphertexts by $\{(\{(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})\}^{s}_{k=1}, (\mathbf{B_i}^{[\beta]}, \mathbf{B'_i}^{[\beta]}), \sigma_i, \eta_i)\}^{\alpha}_{i=1}$ and $\{(\{(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})\}^{s}_{k=1}, \delta_j)\}^{\beta}_{j=1}$;
  - Compute $\mathbf{I}^{[\beta]} := \prod^{\alpha}_{i=1}(\mathbf{B_i}^{[\beta]})^{\eta_i}$, $\mathbf{I'}^{[\beta]} := \prod^{\alpha}_{i=1}(\mathbf{B'_i}^{[\beta]})^{\eta_i}$;
  - $\mathcal{C}^{[c]}$ jointly compute $\{m_j\}^{\beta}_{j=1} := \mathsf{LEG.Dec}(\mathbf{I}^{[\beta]}, \mathbf{I'}^{[\beta]})$.
- Upon receiving $(\text{TALLY}, \mathsf{sid})$ from $\mathcal{Z}$, the voting committee member, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, does the following:
  - For $l \in [n]$, compute the following:
    - $S_l := \prod^{\alpha}_{i=1}\prod^{s}_{k=1}(\mathbf{A_{i,k}})^{\tau_k \cdot \eta_i} \cdot \prod^{\beta}_{j=1}\prod^{s}_{k=1}(\mathbf{K_{j,k}})^{\tau_k \cdot m_j}$;
    - $S'_l := \prod^{\alpha}_{i=1}\prod^{s}_{k=1}(\mathbf{A'_{i,k}})^{\tau_k \cdot \eta_i} \cdot \prod^{\beta}_{j=1}\prod^{s}_{k=1}(\mathbf{K'_{j,k}})^{\tau_k \cdot m_j}$;
  - $\mathcal{C}^{[c]}$ jointly compute $f_l := \mathsf{LEG.Dec}(S_l, S'_l)$ for $l \in [n]$;
  - Send $(\mathsf{Write}, \mathsf{sid}, (\{f_l\}^{n}_{l=1}))$ to $\mathcal{F}_{\text{BC}}$.
- Upon receiving $(\text{REVEAL}, \mathsf{sid}, \mathsf{E}^{(j)}_{\mathsf{fld}})$ from $\mathcal{Z}$, the expert, $\mathsf{E}^{(j)}_{\mathsf{fld}} \in \mathcal{E}^{[e]}_{\mathsf{fld}}$, sends $(\mathsf{Write}, \mathsf{sid}, (\{\mathbf{r_{j,k}}^{[n]}, \mathbf{p_{j,k}}^{[n]}\}^{s}_{k=1}))$ to $\mathcal{F}_{\text{BC}}$, returns $(\text{REVEALEXPERT}, \mathsf{sid}, \{\mathbf{r_{j,k}}^{[n]}, \mathbf{p_{j,k}}^{[n]}\}^{s}_{k=1})$ to $\mathcal{Z}$;
- Upon receiving $(\text{READTALLY}, \mathsf{sid})$ from $\mathcal{Z}$, the party, P, sends $(\mathsf{Read}, \mathsf{sid})$ to $\mathcal{F}_{\text{BC}}$ and gets $(\{f_l\}^{n}_{l=1})$. Then P returns $(\text{READTALLYRETURN}, \mathsf{sid}, (\{f_l\}^{n}_{l=1})$ to $\mathcal{Z}$.

Figure 6.8: Stage 1: Preferential Voting protocol $\Pi^{c,\mu,s,n}_{\text{VOTE1}}$ in $\{\mathcal{F}_{\text{BC}}, \mathcal{F}^{c,\mu,1}_{\text{DBKG}}\}$-*hybrid world* (Part 2).

6.2.3 Zero-Knowledge Proofs in $\Pi^{c,\mu,s,n}_{\mathbf{VOTE}1}$

In the Preferential Voting stage, both experts and voters are required to generate a non-interactive zero-knowledge proof (NIZK proof) to demonstrate that certain ciphertexts encrypt a unit vector and the product of some ciphertexts encrypts either 0 or 1. Specifically, the unit vector NIZK proof is adopted from [13], as explained in Section 6.2.3.2. Additionally, a Batched 0 or 1 Encryption Proof, presented in Section 6.2.3.1, is utilised to prove that each of the ciphertexts encrypts either 0 or 1. The combination of the Batched 0 or 1 Encryption Proof and the Unit Vector NIZK proof is employed to verify the correctness of experts' and voters' ballots in the Preferential Voting stage, as discussed in Section 6.2.3.3.

6.2.3.1 Batched 0 or 1 Encryption Proof

Given a set of ciphertexts, $\{C_{i,1} := g^{r_i}, C_{i,2} := g^{m_i} \cdot (\mathsf{pk})^{r_i}\}_{i=1}^{N}$, $\mathcal{P}$ proves to $\mathcal{V}$ that $(C_{i,1}, C_{i,2})$ encrypts either 0 or 1. The idea to construct this proof is based on the Σ protocol for knowledge of a committed value being 0 or 1 [240] with batch verification for efficiency. The argument of Batched 0 or 1 Encryption Proof is given in Figure 6.9 and its proof is presented in Theorem 11.

In the first move, $\mathcal{V}$ selects a random λ for batch verification, which can also be constructed for NIZK by computing

$$\lambda \leftarrow \mathsf{hash}(\langle\{C_{i,1} := g^{r_i}, C_{i,2} := g^{m_i} \cdot (\mathsf{pk})^{r_i}\}_{i=1}^{N}. \tag{6.11}$$

In the second move, $\mathcal{P}$ selects random $\{r'_i, m'_i, \rho_i\}_{i=1}^{N}$, and computes

$$\begin{aligned}
&C'_{i,1} := g^{r'_i}, C'_{i,2} := g^{m'_i} \cdot (\mathsf{pk})^{r'_i} \text{ for } i \in [N],\\
&a_i := g^{\rho_i}, b_i := g^{m_i \cdot m'_i} \cdot (\mathsf{pk})^{\rho_i} \text{ for } i \in [N],\\
&A := \sum_{i=1}^{N} (a_i)^{\lambda^{i-1}}, B := \sum_{i=1}^{N} (b_i)^{\lambda^{i-1}},\\
&E := \prod_{i=1}^{N} (C'_{i,1})^{\lambda^{i-1}}, F := \prod_{i=1}^{N} (C'_{i,2})^{\lambda^{i-1}},
\end{aligned} \tag{6.12}$$

Then $\mathcal{P}$ sends A, B, E, F to $\mathcal{V}$, which returns either a random challenge, e, or compute the challenge for NIZK,

$$e \leftarrow \mathsf{hash}(\langle A, B, E, F\rangle. \tag{6.13}$$

In the last move, $\mathcal{P}$ answers $\mathcal{V}$'s challenge by returning $Z_1, \langle z_{i,2}\rangle_{i=1}^{N}, Z_3$:

$$\begin{aligned}
&z_{i,1} := r'_i + r_i \cdot e, z_{i,2} := m'_i + m_i \cdot e, z_{i,3} := \rho_i + r_i \cdot (e - z_{i,2}) \text{ for } i \in [N],\\
&Z_1 := \sum_{i=1}^{N} z_{i,1} \cdot \lambda^{i-1}, Z_3 := \sum_{i=1}^{N} z_{i,3} \cdot \lambda^{i-1}.
\end{aligned} \tag{6.14}$$

To verify $\mathcal{P}$'s response, $\mathcal{V}$ computes

$$\begin{aligned}
&\lambda \leftarrow \mathsf{hash}(\langle \{C_{i,1}, C_{i,2}\}_{i=1}^{N} \text{ for } \mathsf{NIZK},\\
&e \leftarrow \mathsf{hash}(\langle A, B, E, F\rangle) \text{ for } \mathsf{NIZK},\\
&C := \prod_{i=1}^{N} (C_{i,1})^{\lambda^{i-1}}, D := \prod_{i=1}^{N} (C_{i,2})^{\lambda^{i-1}}, Z_2 := \sum_{i=1}^{N} z_{i,2} \cdot \lambda^{i-1},
\end{aligned} \tag{6.15}$$

and verifies

$$\begin{aligned}
&(C)^e \cdot E = g^{Z_1},\\
&(D)^e \cdot F = g^{Z_2} \cdot \mathsf{pk}^{Z_1},\\
&(\prod_{i=1}^{N} (C_{i,1})^{\lambda^{i-1}\cdot(e-z_{i,2})}) \cdot A = g^{Z_3},\\
&(\prod_{i=1}^{N} (C_{i,2})^{\lambda^{i-1}\cdot(e-z_{i,2})}) \cdot B = \mathsf{pk}^{Z_3}.
\end{aligned} \tag{6.16}$$

Batched 0 or 1 Encryption ZK argument

CRS: $\{g, \mathsf{pk}\} \in \mathbb{G} \setminus \{1\}$
Statement: $\{C_{i,1} := g^{r_i}, C_{i,2} := g^{m_i} \cdot (\mathsf{pk})^{r_i}\}_{i=1}^{N}$
Witness: $\{m_1, \ldots, m_N\} \in \{0,1\}^{[N]}$, and randomness set $\{r_1, \ldots, r_N\} \in (\mathbb{Z}_p)^{[N]}$
Protocol:

- $\mathcal{V} \rightarrow \mathcal{P} : \lambda \leftarrow \mathbb{Z}_q$; /* For NIZK, set $\lambda \leftarrow \mathsf{hash}(\langle\{C_{i,1} := g^{r_i}, C_{i,2} := g^{m_i} \cdot (\mathsf{pk})^{r_i}\}_{i=1}^{N}\rangle$ */
- $\mathcal{P}$ computes the following:
  - Select $\{r'_i, m'_i, \rho_i\}_{i=1}^{N} \leftarrow (\mathbb{Z}_p)^{[N \cdot 3]}$;
  - For $i \in [N]$:
    - $C'_{i,1} := g^{r'_i}$, $C'_{i,2} := g^{m'_i} \cdot (\mathsf{pk})^{r'_i}$;
    - $a_i := g^{\rho_i}$, $b_i := g^{m_i \cdot m'_i} \cdot (\mathsf{pk})^{\rho_i}$;
  - $A := \sum_{i=1}^{N} (a_i)^{\lambda^{i-1}}$, $B := \sum_{i=1}^{N} (b_i)^{\lambda^{i-1}}$, $E := \prod_{i=1}^{N} (C'_{i,1})^{\lambda^{i-1}}$, $F := \prod_{i=1}^{N} (C'_{i,2})^{\lambda^{i-1}}$;
  - $\mathcal{P} \rightarrow \mathcal{V} : \langle A, B, E, F \rangle$.
- $\mathcal{V} \rightarrow \mathcal{P} : e \leftarrow \mathbb{Z}_q$; /* For NIZK, set $e \leftarrow \mathsf{hash}(\langle A, B, E, F \rangle)$ */
- $\mathcal{P}$ computes the following:
  - For $i \in [N]$:
    - $z_{i,1} := r'_i + r_i \cdot e$, $z_{i,2} := m'_i + m_i \cdot e$, $z_{i,3} := \rho_i + r_i \cdot (e - z_{i,2})$;
  - $Z_1 := \sum_{i=1}^{N} z_{i,1} \cdot \lambda^{i-1}$, $Z_3 := \sum_{i=1}^{N} z_{i,3} \cdot \lambda^{i-1}$;
  - $\mathcal{P} \rightarrow \mathcal{V} : \langle Z_1, \langle z_{i,2} \rangle_{i=1}^{N}, Z_3 \rangle$.

Verification:
$\mathcal{V}$ computes the following:

- $\lambda \leftarrow \mathsf{hash}(\langle\{C_{i,1}, C_{i,2}\}_{i=1}^{N}$ for NIZK;
- $e \leftarrow \mathsf{hash}(\langle A, B, E, F \rangle)$ for NIZK;
- $C := \prod_{i=1}^{N} (C_{i,1})^{\lambda^{i-1}}$, $D := \prod_{i=1}^{N} (C_{i,2})^{\lambda^{i-1}}$, $Z_2 := \sum_{i=1}^{N} z_{i,2} \cdot \lambda^{i-1}$.

$\mathcal{V}$ check the following:

- $(C)^e \cdot E = g^{Z_1}$;
- $(D)^e \cdot F = g^{Z_2} \cdot \mathsf{pk}^{Z_1}$;
- $(\prod_{i=1}^{N} (C_{i,1})^{\lambda^{i-1} \cdot (e - z_{i,2})}) \cdot A = g^{Z_3}$;
- $(\prod_{i=1}^{N} (C_{i,2})^{\lambda^{i-1} \cdot (e - z_{i,2})}) \cdot B = \mathsf{pk}^{Z_3}$.

Figure 6.9: Batched 0 or 1 Encryption ZK argument.

Theorem 11 (Batched 0 or 1)**.** *Assume the DDH problem is hard. The protocol described in Fig.6.9 is an honest verifier zero-knowledge argument of knowledge of* $\mathbf{m} := \{m_1, \ldots, m_N\} \in \{0,1\}^{[N]}$*,* $\{r_1, \ldots, r_N\} \in (\mathbb{Z}_p)^{[N]}$ *such that:*

- $C_{i,1} := g^{r_i}, C_{i,2} := g^{m_i} \cdot (\mathsf{pk})^{r_i}$ *for* $i \in [N]$*;*
- $m_i \in \{0,1\}$ *for* $i \in [N]$.

Proof of Theorem 11.

- **Completeness.**
  Firstly, the following can be observed:

$$
\begin{aligned}
C &:= \prod_{i=1}^{N} (C_{i,1})^{\lambda^{i-1}} = g^{\sum_{i=1}^{N} r_i \cdot \lambda^{i-1}}, \\
D &:= \prod_{i=1}^{N} (C_{i,2})^{\lambda^{i-1}} = g^{\sum_{i=1}^{N} m_i \cdot \lambda^{i-1}} \cdot \mathsf{pk}^{\sum_{i=1}^{N} r_i \cdot \lambda^{i-1}}, \\
Z_1 &:= \sum_{i=1}^{N} (r'_i + r_i \cdot e) \cdot \lambda^{i-1}, \\
Z_3 &:= \sum_{i=1}^{N} (\rho_i + r_i \cdot (e - z_{i,2})) \cdot \lambda^{i-1}, \\
Z_2 &:= \sum_{i=1}^{N} (m'_i + m_i \cdot e) \cdot \lambda^{i-1}, z_{i,2} := m'_i + m_i \cdot e, \\
A &:= g^{\sum_{i=1}^{N} \rho_i \cdot \lambda^{i-1}}, E := \prod_{i=1}^{N} (g^{r'_i})^{\lambda^{i-1}} = g^{\sum_{i=1}^{N} r'_i \cdot \lambda^{i-1}}, \\
B &:= \prod_{i=1}^{N} (g^{m_i \cdot m'_i} \cdot (\mathsf{pk})^{\rho_i})^{\lambda^{i-1}} = g^{\sum_{i=1}^{N} m_i \cdot m'_i \cdot \lambda^{i-1}} \cdot \mathsf{pk}^{\sum_{i=1}^{N} \rho_i \cdot \lambda^{i-1}}, \\
F &:= \prod_{i=1}^{N} (g^{m'_i} \cdot (\mathsf{pk})^{r'_i})^{\lambda^{i-1}} = g^{\sum_{i=1}^{N} m'_i \cdot \lambda^{i-1}} \cdot \mathsf{pk}^{\sum_{i=1}^{N} r'_i \cdot \lambda^{i-1}}.
\end{aligned}
\tag{6.17}
$$

  By the homophobic property of Lifted Elgamal Encryption, the following equation

can be proved.

$$
\begin{aligned}
&(C)^e \cdot E = g^{\sum_{i=1}^{N} e\cdot r_i\cdot\lambda^{i-1}+\sum_{i=1}^{N} r'_i\cdot\lambda^{i-1}} = g^{Z_1},\\
&(D)^e \cdot F = g^{e\cdot\sum_{i=1}^{N} m_i\cdot\lambda^{i-1}+\sum_{i=1}^{N} m'_i\cdot\lambda^{i-1}} \cdot \mathsf{pk}^{e\cdot\sum_{i=1}^{N} r_i\cdot\lambda^{i-1}+\sum_{i=1}^{N} r'_i\cdot\lambda^{i-1}} = g^{Z_2}\cdot\mathsf{pk}^{Z_1},\\
&(\prod_{i=1}^{N}(C_{i,1})^{\lambda^{i-1}\cdot(e-z_{i,2})})\cdot A = g^{\sum_{i=1}^{N} r_i\cdot\lambda^{i-1}\cdot(e-z_{i,2})}\cdot g^{\sum_{i=1}^{N}\rho_i\cdot\lambda^{i-1}} = g^{Z_3},\\
&(\prod_{i=1}^{N}(C_{i,2})^{\lambda^{i-1}\cdot(e-z_{i,2})})\cdot B = g^{\sum_{i=1}^{N} m_i\cdot(e-z_{i,2})\cdot\lambda^{i-1}}\cdot g^{\sum_{i=1}^{N} m_i\cdot m'_i\cdot\lambda^{i-1}}\\
&\qquad\qquad\qquad\qquad\cdot\mathsf{pk}^{\sum_{i=1}^{N} r_i\cdot(e-z_{i,2})\cdot\lambda^{i-1}+\sum_{i=1}^{N}\rho_i\cdot\lambda^{i-1}}\\
&= g^{\sum_{i=1}^{N}\lambda^{i-1}(m_i\cdot(e-m'_i-m_i\cdot e)+m_i\cdot m'_i)}\cdot\mathsf{pk}^{Z_3} = g^{\sum_{i=1}^{N}\lambda^{i-1}(m_i\cdot e-m_i\cdot m'_i-{m_i}^2\cdot e+m_i\cdot m'_i)}\cdot\mathsf{pk}^{Z_3}\\
&= g^{\sum_{i=1}^{N}\lambda^{i-1}(m_i\cdot e)\cdot(1-m_i)}\cdot\mathsf{pk}^{Z_3} = \mathsf{pk}^{Z_3}.
\end{aligned}
\tag{6.18}
$$

- **Soundness.**

  The soundness of the protocol is proven by showing that the protocol is an argument of knowledge (AoK) and it has a witness-extended emulator. At first, since $\lambda \in \mathbb{Z}_q^*$ is randomly chosen by $\mathcal{V}$, based on Schwartz-Zippel lemma, $\mathcal{P}$ has negligible probability of convincing $\mathcal{V}$ unless all λ^{i-1} related variables match on each side of the equality for all $j \in [N]$.

  Assume that there exists a PPT witness-extended extractor, $\mathcal{E}$, runs $\langle\mathcal{P}^*, \mathcal{V}\rangle$ to get transcripts. In addition, if $\mathcal{P}$ is able to make an acceptable argument, then $\mathcal{E}$ can also succeed with the same probability. $\mathcal{E}$ rewinds the protocol to the first challenge phase (λ) and runs it with fresh challenges until it has n acceptable arguments. More specifically, each time $\alpha \in [N]$, $\mathcal{E}$ first gives new challenge (λ_α) and a challenge $e_{\alpha,1}$, then $\mathcal{E}$ can get:

$$
\begin{aligned}
z_{i,2}^{(\alpha,1)} &:= m'_i + m_i\cdot e_{\alpha,1} \text{ for } i \in [N],\\
Z_1^{(\alpha,1)} &:= \sum_{i=1}^{N}(r'_i + r_i\cdot e_{\alpha,1})\cdot\lambda^{i-1}.
\end{aligned}
\tag{6.19}
$$

  Then, $\mathcal{E}$ rewinds the protocol to the second challenge phase feeding new challenge $e_{\alpha,2}$ and gets:

$$
\begin{aligned}
z_{i,2}^{(\alpha,2)} &:= m'_i + m_i\cdot e_{\alpha,2} \text{ for } i \in [N],\\
Z_1^{(\alpha,2)} &:= \sum_{i=1}^{N}(r'_i + r_i\cdot e_{\alpha,2})\cdot\lambda^{i-1}.
\end{aligned}
\tag{6.20}
$$

  For $i \in [N]$, by computing $(z_{i,1}^{(\alpha,1)} - z_{i,1}^{(\alpha,2)})/(e_{\alpha,1} - e_{\alpha,2})$, $\mathcal{E}$ can get witness $\{m_i\}_{i=1}^{N}$.

Set $\mathcal{Y} := \begin{bmatrix} r_1 & \dots & r_N \end{bmatrix}$, by computing $(Z_1^{(\alpha,1)} - Z_1^{(\alpha,2)})/(e_{\alpha,1} - e_{\alpha,2})$, $\mathcal{E}$ gets

$$\Gamma_\alpha = \mathcal{Y} \cdot \begin{bmatrix} 1 \\ \lambda_\alpha \\ \dots \\ (\lambda_\alpha)^{N-1} \end{bmatrix}. \tag{6.21}$$

There is overwhelming probability that we have transcripts with n different challenges, these challenges give us a $(N) \times (N)$ invertible transposed Vandermonde polynomial matrix:

$$\Lambda = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \lambda_1 & \lambda_2 & \dots & \lambda_N \\ \vdots & \vdots & \ddots & \vdots \\ (\lambda_1)^{N-1} & (\lambda_2)^{N-1} & \dots & (\lambda_N)^{N-1} \end{pmatrix}. \tag{6.22}$$

Denote $\Omega := [\Gamma_1, \dots, \Gamma_N]$. We have $\mathcal{Y} \cdot \Lambda = \Omega$, $\mathcal{E}$ can get witness $\{r_i\}_{i=1}^N$.

- **Zero Knowledge.**

  In terms of special honest verifier zero-knowledge, we construct a simulator S that takes the challenges $\{\lambda, e\} \leftarrow \mathbb{Z}_q$ and statement $\{C_{i,k}\}_{k=0}^{t-1}$, $\{A_{i,j,b}, B_{i,j,b}\}_{j=1,b=1}^{n,\eta}$ as inputs, it should output a simulated transcript the distribution of which is indistinguishable from the real one.

  In detail, S firstly picks $\{z_{i,2}\}_{i=1}^N, Z_1, Z_3$ from $\mathbb{Z}_q$, then computes Z_2, C, D according to the protocol description, and computes the followings:

$$\begin{aligned} E &= (C)^{-e} \cdot g^{Z_1}, \\ F &= (D)^{-e} \cdot g^{Z_2} \cdot \mathsf{pk}^{Z_1}, \\ A &= (\prod_{i=1}^{N} (C_{i,1})^{\lambda^{i-1} \cdot (e - z_{i,2})})^{-1} \cdot g^{Z_3}, \\ B &= (\prod_{i=1}^{N} (C_{i,2})^{\lambda^{i-1} \cdot (e - z_{i,2})})^{-1} \cdot \mathsf{pk}^{Z_3}. \end{aligned} \tag{6.23}$$

  After that, S outputs the simulated transcripts $\langle A, B, E, F, Z_1, \{z_{i,2}\}_{i=1}^N, Z_3 \rangle$. Since $\{r'_i, m'_i, \rho_i\}_{i=1}^N, \omega, \upsilon$ are uniformly random, the distribution of simulated $\{z_{i,1}\}_{i=1}^N, Z_1, Z_3$ should also be uniformly random, hence simulated $\{z_{i,1}\}_{i=1}^N, Z_1, Z_3$ are identical to the distribution of them in the argument. In addition, we have A, B, E, F follow the same distribution in real argument, as they are uniquely determined for fixed elements from group $\mathbb{G}$. Therefore, to conclude, simulated transcripts has the same distribution as real transcripts in a real argument.

□

6.2.3.2 Unit Vector Proof

The Unit Vector Proof from [13] is used in this thesis. For completeness, we present the original proof and construction in Figure 6.10 and Theorem 12.

Theorem 12 (Unit Vector ZK, [13])**.** *The protocol described in Figure 6.10 is a* 5*-move public coin special honest verifier zero-knowledge argument of knowledge of* $\mathbf{U}_i^{[n]} = (u_{i,0}, \ldots, u_{i,n-1}) \in \{0,1\}^n$ *and* $(r_0, \ldots, r_{n-1}) \in (\mathbb{Z}_p)^n$ *such that* $C_j = \mathsf{Enc}_{\mathsf{pk}}(u_{i,j}; r_j)$*,* $j \in [0, n-1]$.

The proof of Theorem 12 can be found in [13].

Unit vector ZK argument

CRS: $\{g, \mathsf{ck}, \mathsf{pk}\} \in \mathbb{G} \setminus \{1\}$
Statement: ciphertexts $C_0 := \mathsf{Enc}_{\mathsf{pk}}(u_{i,0}; r_0), \ldots, C_{n-1} := \mathsf{LEG.Enc}_{\mathsf{pk}}(u_{i,n-1}; r_{n-1})$
Witness: $\{i_\ell\}_{\ell=1}^{\log_n}$, $\{r_j\}_{j=0}^{n-1}$
Protocol:

- $\mathcal{P}$ for $\ell \in [\log n]$, does the following:
  - Pick random $\alpha_\ell, \beta_\ell, \gamma_\ell, \delta_\ell \leftarrow \mathbb{Z}_p$;
  - Compute $I_\ell := \mathsf{PC.Com}_{\mathsf{ck}}(i_\ell; \alpha_\ell)$, $B_\ell := \mathsf{PC.Com}_{\mathsf{ck}}(\beta_\ell; \gamma_\ell)$ and $A_\ell := \mathsf{PC.Com}_{\mathsf{ck}}(i_\ell \cdot \beta_\ell; \delta_\ell)$;
  - $\mathcal{P} \to \mathcal{V}$: $\langle \langle I_\ell, B_\ell, A_\ell \rangle_{\ell=1}^{\log n} \rangle$;
- $\mathcal{V} \to \mathcal{P}$: $\langle y \leftarrow \mathbb{Z}_p \rangle$ /* For NIZK, set $y \leftarrow \mathsf{hash}(\langle g, \mathsf{ck}, \mathsf{pk}, \langle C_j \rangle_{j=0}^{n-1}, \langle I_\ell, B_\ell, A_\ell \rangle_{\ell=1}^{\log n} \rangle$ */ ;
- $\mathcal{P}$ for $\ell = 0, \ldots, \log n - 1$, does the following:
  - Select $R_\ell \leftarrow \mathbb{Z}_p$;
  - Compute $D_\ell := \mathsf{LEG.Enc}_{\mathsf{pk}}\big(\sum_{j=0}^{n-1}(p_{j,\ell} \cdot y^j); R_\ell\big)$;
  - $\mathcal{P} \to \mathcal{V}$: $\langle \{D_\ell\}_{\ell=0}^{\log n - 1} \rangle$;
- $\mathcal{V} \to \mathcal{P}$: $\langle x \leftarrow \mathbb{Z}_p \rangle$ /* For NIZK, set $x \leftarrow \mathsf{hash}(\langle D_\ell \rangle_{\ell=0}^{\log n - 1}$ */;
- $\mathcal{P}$ does the following:
  - Compute $R := \sum_{j=0}^{n-1}(r_j \cdot x^{\log n} \cdot y^j) + \sum_{\ell=0}^{\log n - 1}(R_\ell \cdot x^\ell)$;
  - For $\ell = 1, \ldots, \log n$, compute $z_\ell := i_\ell \cdot x + \beta_\ell$, $w_\ell := \alpha_\ell \cdot x + \gamma_\ell$, and $v_\ell := \alpha_\ell(x - z_\ell) + \delta_\ell$;
  - $\mathcal{P} \to \mathcal{V}$: $\langle R, \{z_\ell, w_\ell, v_\ell\}_{\ell=1}^{\log n} \rangle$.

Verification:
$\mathcal{V}$ computes the following:

- Set $y \leftarrow \mathsf{hash}(\langle g, \mathsf{ck}, \mathsf{pk}, \langle C_j \rangle_{j=0}^{n-1}, \langle I_\ell, B_\ell, A_\ell \rangle_{\ell=1}^{\log n} \rangle$ for NIZK;
- Set $x \leftarrow \mathsf{hash}(\langle D_\ell \rangle_{\ell=0}^{\log n - 1}$ for NIZK.

$\mathcal{V}$ checks the following:

- For $\ell = 1, \ldots, \log n$, do:
  - $(I_\ell)^x \cdot B_\ell = \mathsf{PC.Com}_{\mathsf{ck}}(z_\ell; w_\ell)$
  - $(I_\ell)^{x - z_\ell} \cdot A_\ell = \mathsf{PC.Com}_{\mathsf{ck}}(0; v_\ell)$
- $\prod_{j=0}^{n-1}\big((C_j)^{x^{\log n}} \cdot \mathsf{Enc}_{\mathsf{pk}}(-\prod_{\ell=1}^{\log n} z_{\ell,j_\ell}; 0)\big)^{y^j} \cdot \prod_{\ell=0}^{\log n - 1}(D_\ell)^{x^\ell} = \mathsf{Enc}_{\mathsf{pk}}(0; R)$, where $z_{j,1} = z_j$ and $z_{j,0} = x - z_j$.

Figure 6.10: Unit vector ZK argument, [13].

6.2.3.3 Valid Ballot Proof

In this proof, $\mathcal{P}$ shows $\mathcal{V}$ that its ballot is valid:

- For $k \in [s]$, $(\mathbf{A_{i,k}}^{[n]}||\mathbf{B_i}^{[e]}, \mathbf{A'_{i,k}}^{[n]}||\mathbf{B'_i}^{[e]})$ encrypts a unit vector;
- For $l \in [n]$, $(\prod_{k=1}^{s} A_{i,k,l}, \prod_{k=1}^{s} A'_{i,k,l})$ encrypts either 0 or 1,
  where $\{(A_{i,k,l}, A'_{i,k,l})\}_{k=1,l=1}^{s,n} := (\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})$.

We give the protocol in Figure 6.11 and Figure 6.12, which can be considered as two NIZK proofs, a unit vector encryption proof and a batched 0 or 1 proof. The proof is illustrated in Theorem 13.

Theorem 13. *Assume the DDH problem is hard, the protocol described in Figure 6.11 and Figure 6.12 is an honest special verifier zero-knowledge argument of knowledge of* $\{\mathbf{r_{i,k}}^{[n]}, \mathbf{v_{i,k}}^{[n]}\}_{k=1}^{s}$, $\mathbf{r'_i}^{[e]}$, $\mathbf{v'_i}^{[e]}$, *unit vectors* $\{\mathbf{v_{i,k}}^{[n]}||\mathbf{v'_i}^{[e]}\}_{k=1}^{s} \in \{0,1\}^{(n+e)\cdot s}$, η_k *is the position of value 1 in unit vector* $\mathbf{v_{i,k}}^{[n]}||\mathbf{v'_i}^{[e]}$ *for* $k \in [s]$, *such that:*

- $(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]}) = \mathsf{LEG.Enc}_{\mathsf{gpk}}(\mathbf{v_{i,k}}^{[n]}, \mathbf{r_{i,k}}^{[n]})$ *for* $k \in [s]$*;*
- $(\mathbf{B_i}^{[e]}, \mathbf{B'_i}^{[e]}) = \mathsf{LEG.Enc}_{\mathsf{gpk}}(\mathbf{v'_i}^{[e]}, \mathbf{r'_i}^{[e]})$*;*
- $(\mathbf{A_{i,k}}^{[n]}||\mathbf{B_i}^{[e]}, \mathbf{A'_{i,k}}^{[n]}||\mathbf{B'_i}^{[e]})$ *encrypts a unit vector;*
- *For* $l \in [n]$, $(\prod_{k=1}^{s} A_{i,k,l}, \prod_{k=1}^{s} A'_{i,k,l})$ *encrypts either 0 or 1,*
  where $\{(A_{i,k,l}, A'_{i,k,l})\}_{k=1,l=1}^{s,n} := (\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})$.

Theorem 13 can be proved by Theorem 12 and Theorem 11.

Valid Casting ZK argument (Part 1)

CRS: $\{g, \mathsf{gpk}, \mathsf{ck}\} \in \mathbb{G} \setminus \{1\}$
Statement: $\{(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})\}_{k=1}^{s}$, $(\mathbf{B_i}^{[e]}, \mathbf{B'_i}^{[e]})$
Witness: $\{\mathbf{r_{i,k}}^{[n]}, \mathbf{v_{i,k}}^{[n]}\}_{k=1}^{s}$, $\mathbf{r'_i}^{[e]}$, $\mathbf{v'_i}^{[e]}$, unit vectors $\{\mathbf{v_{i,k}}^{[n]}||\mathbf{v'_i}^{[e]}\}_{k=1}^{s} \in \{0,1\}^{(n+e)\cdot s}$, η_k is the position of value 1 in unit vector $\mathbf{v_{i,k}}^{[n]}||\mathbf{v'_i}^{[e]}$ for $k \in [s]$
Protocol:

- $\mathcal{P}$ does the following:
  - For $k \in [s]$, set $\{r_{k,j}\}_{k=1,j=0}^{s,n+e-1} := \mathbf{r_{i,k}}^{[n]}||\mathbf{r'_i}^{[e]}$, and $\{u_{k,j}\}_{k=1,j=0}^{s,n+e-1} := \mathbf{v_{i,k}}^{[n]}||\mathbf{v'_i}^{[e]}$, $C_{k,j} := \mathsf{LEG.Enc}_{\mathsf{gpk}}(u_{k,j}, r_{k,j})$;
  - $\mathcal{P}$ for $k \in [s]$, for $\ell = 1, \ldots, \log(n+e)$, does the following:
    - Pick random $\alpha_{k,\ell}, \beta_{k,\ell}, \gamma_{k,\ell}, \delta_{k,\ell} \leftarrow \mathbb{Z}_p$. Compute $I_{k,\ell} := \mathsf{PC.Com}_{\mathsf{ck}}(\eta_{k,\ell}; \alpha_{k,\ell})$, $B_{k,\ell} := \mathsf{PC.Com}_{\mathsf{ck}}(\beta_{k,\ell}; \gamma_{k,\ell})$ and $\bar{A}_{k,\ell} := \mathsf{PC.Com}_{\mathsf{ck}}(\eta_{k,\ell} \cdot \beta_{k,\ell}; \delta_{k,\ell})$.
  - $\mathcal{P} \rightarrow \mathcal{V} : \langle g, \mathsf{gpk}, \langle \mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]} \rangle_{k=1}^{s}, \mathbf{B_i}^{[e]}, \mathbf{B'_i}^{[e]}, \langle I_{k,\ell}, B_{k,\ell}, \bar{A}_{k,\ell} \rangle_{k=1,\ell=1}^{s,\log(n+e)} \rangle$.
- $\mathcal{V} \rightarrow \mathcal{P} : \lambda \leftarrow \mathbb{Z}_q$; /* For NIZK, set $\lambda \leftarrow \mathsf{hash}(g, \mathsf{gpk}, \langle \mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]} \rangle_{k=1}^{s}, \mathbf{B_i}^{[e]}, \mathbf{B'_i}^{[e]}, \langle I_{k,\ell}, B_{k,\ell}, \bar{A}_{k,\ell} \rangle_{k=1,\ell=1}^{s,\log(n+e)})$*/
- $\mathcal{P}$ does the following:
  - Set $\{A_{i,k,l}\}_{k=1,l=1}^{s,n} := \{\mathbf{A_{i,k}}^{[n]}\}_{k=1}^{s}$, $\{A'_{i,k,l}\}_{k=1,l=1}^{s,n} := \{\mathbf{A'_{i,k}}^{[n]}\}_{k=1}^{s}$, $\{r_{i,k,l}\}_{k=1,l=1}^{s,n} := \{\mathbf{r_{i,k}}^{[n]}\}_{k=1}^{s}$, $\{v_{i,k,l}\}_{k=1,l=1}^{s,n} := \{\mathbf{v_{i,k}}^{[n]}\}_{k=1}^{s}$;
  - For $l \in [n]$, set $D_l := \prod_{k=1}^{s} A_{i,k,l}$, $D'_l := \prod_{k=1}^{s} A'_{i,k,l}$, and $m_l := \sum_{k=1}^{s} v_{i,k,l}$, $t_l := \sum_{k=1}^{s} r_{i,k,l}$;
  - Select $\{t'_l, m'_l, \rho_l\}_{l=1}^{n} \leftarrow (\mathbb{Z}_p)^{[n\cdot 3]}$;
  - For $l \in [n]$, compute $C'_{l,1} := g^{t'_l}$, $C'_{l,2} := g^{m'_l} \cdot (\mathsf{gpk})^{t'_l}$, $a_l := g^{\rho_l}$, $b_l := g^{m_l \cdot m'_l} \cdot (\mathsf{gpk})^{\rho_l}$;
  - Compute $G := \sum_{l=1}^{n} (a_l)^{\lambda^{l-1}}$, $H := \sum_{l=1}^{n} (b_l)^{\lambda^{l-1}}$, $E := \prod_{l=1}^{n} (C'_{l,1})^{\lambda^{l-1}}$, $F := \prod_{l=1}^{n} (C'_{l,2})^{\lambda^{l-1}}$;
  - For $k \in [s]$, $\ell = 0, \ldots, \log(n+e) - 1$, does the following:
    - Select $R_{k,\ell} \leftarrow \mathbb{Z}_p$. Compute $M_{k,\ell} := \mathsf{LEG.Enc}_{\mathsf{gpk}}\big(\sum_{j=0}^{n-1} (p_{j,k,\ell} \cdot \lambda^j); R_{k,\ell}\big)$.
  - $\mathcal{P} \rightarrow \mathcal{V} : \langle G, H, E, F, \langle M_{k,\ell} \rangle_{\ell=0}^{\log(n+e)-1} \rangle$.
- $\mathcal{V} \rightarrow \mathcal{P} : e' \leftarrow \mathbb{Z}_q$; /* For NIZK, set $e' \leftarrow \mathsf{hash}(\langle G, H, E, F, \langle M_{k,\ell} \rangle_{\ell=0}^{\log(n+e)-1} \rangle)$*/

Figure 6.11: Valid Ballot ZK argument (Part 1).

Valid Ballot ZK argument (Part 2)

- $\mathcal{P}$ computes the following:
  - For $l \in [n]$, $z_{l,1} := t'_l + t_i \cdot e$, $z_{l,2} := m'_l + m_l \cdot e$, $z_{l,3} := \rho_l + t_l \cdot (e - z_{l,2})$;
  - $Z_1 := \sum_{l=1}^{n} z_{l,1} \cdot \lambda^{l-1}$, $Z_3 := \sum_{l=1}^{n} z_{l,3} \cdot \lambda^{l-1}$;
  - For $k \in [s]$, compute $R_k := \sum_{j=0}^{n-1}(r_{j,k} \cdot (e')^{\log(n+e)} \cdot \lambda^j) + \sum_{\ell=0}^{\log(n+e)-1}(R_{k,\ell} \cdot (e')^{\ell})$;
  - For $\ell \in [\log(n+e)]$, compute $z'_{k,\ell} := \eta_{k,\ell} \cdot (e') + \beta_{k,\ell}$, $w_{k,\ell} := \alpha_{k,\ell} \cdot (e') + \gamma_{k,\ell}$, and $v_{k,\ell} := \alpha_{k,\ell} \cdot ((e') - z'_{k,\ell}) + \delta_{k,\ell}$;
  - $\mathcal{P} \to \mathcal{V} : \langle Z_1, \langle z_{l,2} \rangle_{l=1}^{n}, Z_3, \langle R_k \rangle_{k=1}^{s}, \langle z'_{k,\ell}, w_{k,\ell}, v_{k,\ell} \rangle_{k=1,\ell=1}^{s,\log(n+e)} \rangle$.

Verification:

$\mathcal{V}$ computes the following:

- Set $\lambda \leftarrow \mathsf{hash}(g, \mathsf{gpk}, \langle \mathsf{A_{i,k}}^{[n]}, \mathsf{A'_{i,k}}^{[n]} \rangle_{k=1}^{s}, \mathsf{B_i}^{[e]}, \mathsf{B'_i}^{[e]}, \langle I_{k,\ell}, B_{k,\ell}, \bar{A}_{k,\ell} \rangle_{k=1,\ell=1}^{s,\log(n+e)})$ for NIZK;
- Set $e' \leftarrow \mathsf{hash}(\langle G, H, E, F, \langle M_{k,\ell} \rangle_{\ell=0}^{\log(n+e)-1} \rangle)$ for NIZK;
- Set $\{A_{i,k,l}\}_{k=1,l=1}^{s,n} := \{\mathsf{A_{i,k}}^{[n]}\}_{k=1}^{s}$, $\{A'_{i,k,l}\}_{k=1,l=1}^{s,n} := \{\mathsf{A'_{i,k}}^{[n]}\}_{k=1}^{s}$, $\{r_{i,k,l}\}_{k=1,l=1}^{s,n} := \{\mathsf{r_{i,k}}^{[n]}\}_{k=1}^{s}$, $\{v_{i,k,l}\}_{k=1,l=1}^{s,n} := \{\mathsf{v_{i,k}}^{[n]}\}_{k=1}^{s}$;
- For $l \in [n]$, set $D_l := \prod_{k=1}^{s} A_{i,k,l}$, $D'_l := \prod_{k=1}^{s} A'_{i,k,l}$, and $m_l := \sum_{k=1}^{s} v_{i,k,l}$, $t_l := \sum_{k=1}^{s} r_{i,k,l}$;
- $K := \prod_{l=1}^{n}(D_l)^{\lambda^{l-1}}$, $K' := \prod_{l=1}^{n}(D'_l)^{\lambda^{l-1}}$;
- $Z_2 := \sum_{l=1}^{n} z_{l,2} \cdot \lambda^{l-1}$.

$\mathcal{V}$ checks the following:

- $(K)^{(e')} \cdot E = g^{Z_1}$;
- $(K')^{(e')} \cdot F = g^{Z_2} \cdot \mathsf{pk}^{Z_1}$;
- $(\prod_{l=1}^{n}(D_l)^{\lambda^{l-1} \cdot ((e') - z_{l,2})}) \cdot G = g^{Z_3}$;
- $(\prod_{l=1}^{n}(D'_l)^{\lambda^{l-1} \cdot ((e') - z_{l,2})}) \cdot H = \mathsf{pk}^{Z_3}$;
- For $k \in [s]$, $\ell \in [\log(n+e)]$, do:
  - $(I_{k,\ell})^{(e')} \cdot B_{k,\ell} = \mathsf{PC.Com_{ck}}(z'_{k,\ell}; w_{k,\ell})$;
  - $(I_{k,\ell})^{(e') - z'_{k,\ell}} \cdot \bar{A}_{k,\ell} = \mathsf{PC.Com_{ck}}(0; v_{k,\ell})$;
  - $\prod_{j=0}^{n-1} \left((C_{k,j})^{(e')^{\log(n+e)}} \cdot \mathsf{Enc_{gpk}}(-\prod_{\ell=1}^{\log(n+e)} z'_{\ell,j_\ell}; 0)\right)^{\lambda^j} \cdot \prod_{\ell=0}^{\log(n+e)-1}(D_\ell)^{(e')^\ell} = \mathsf{Enc_{gpk}}(0; R_k)$, where $z'_{j,1} = z'_j$ and $z'_{j,0} = (e') - z'_j$.

Figure 6.12: Valid Ballot ZK argument (Part 2).

6.2.4 Security Analysis of Preferential Voting

Theorem 14 (Preferential Voting)**.** *Assume that Valid ballot* NIZK *is perfect complete, perfect special honest verifier zero knowledge, and computational sound with adversary advantage of* $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Ballot}(1^\kappa,\mathcal{A})$*. Assume Lifted Elgamal encryption* LEG.Enc *is IND-CPA secure with adversary advantage of* $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{LEG}}(1^\kappa,\mathcal{A})$*. The protocol* $\Pi^{c,\mu,s,n}_{\text{VOTE1}}$ *in Figure 6.7 and Figure 6.8 UC-realise* $\mathcal{F}^{c,\mu,s,n}_{\text{VOTE}}[\mathsf{TallyAlg}_1,\mathsf{DelAlg}_1]$ *in Figure 6.2 in* $\{\mathcal{F}_{\text{BC}},\mathcal{F}^{c,\mu,1}_{\text{DBKG}}\}$*-hybrid world against a static adversary with distinguishing advantage*

$$(2ns+e)\cdot\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{LEG}}(1^\kappa,\mathcal{A})+2\cdot\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Ballot}(1^\kappa,\mathcal{A}).$$

Proof of Theorem 14.
To prove theorem 14, we construct a PPT simulator S such that no nonuniform PPT environment $\mathcal{Z}$ can distinguish between 1) the ideal execution $\mathsf{EXEC}_{\mathcal{F}^{c,\mu,s,n}_{\text{VOTE}}[\mathsf{TallyAlg}_1,\mathsf{DelAlg}_1],\mathsf{S},\mathcal{Z}}$ where the parties interact with functionality $\mathcal{F}^{c,\mu,s,n}_{\text{VOTE}}[\mathsf{TallyAlg}_1,\mathsf{DelAlg}_1]$ in the ideal world and corrupted parties are controlled by the simulator S; and 2) the real execution $\mathsf{EXEC}^{\mathcal{F}_{\text{BC}},\mathcal{F}^{c,\mu,1}_{\text{DBKG}}}_{\Pi^{c,\mu,s,n}_{\text{VOTE1}},\mathcal{A},\mathcal{Z}}$ where the voters $\mathcal{V}^{[v]}_{\mathsf{fld}}:=\{\mathsf{V}^{(i)}_{\mathsf{fld}}\}^v_{i=1}$, experts $\mathcal{E}^{[e]}_{\mathsf{fld}}:=\{\mathsf{E}^{(j)}_{\mathsf{fld}}\}^e_{j=1}$, and voting committee $\mathcal{C}^{[c]}:=\{\mathsf{C}^{(t)}\}^c_{t=1}$ run protocol $\Pi^{c,\mu,s,n}_{\text{VOTE1}}$ in the $\{\mathcal{F}_{\text{BC}},\mathcal{F}^{c,\mu,1}_{\text{DBKG}}\}$*-hybrid world* and the corrupted parties are controlled by a dummy adversary $\mathcal{A}$ who simply forwards messages from/to $\mathcal{Z}$.

Simulator. The simulator S internally runs $\mathcal{A}$, forwarding messages to/from the environment $\mathcal{Z}$. The simulator S simulates the following interactions with $\mathcal{A}$: The simulator S simulates the followings interactions with $\mathcal{Z}$:

Initialisation Phase:

- Upon receiving $(\text{INITNOTIFY},\mathsf{sid},\mathsf{C}^{(t)})$ from $\mathcal{F}^{c,\mu,s,n}_{\text{VOTE}}[\mathsf{TallyAlg}_1,\mathsf{DelAlg}_1]$ about an honest voting committee member $\mathsf{C}^{(t)}$, S simulates $\mathcal{F}^{c,\mu,1}_{\text{DBKG}}$ to generate $(\mathsf{gpk},\mathsf{gsk}_t)$ for $\mathsf{C}^{(t)}$.

Voter Cast Phase:

- Upon receiving $(\text{CASTNOTIFY},\mathsf{V}^{(i)}_{\mathsf{fld}},\mathsf{sid},\eta_i)$ from $\mathcal{F}^{c,\mu,s,n}_{\text{VOTE}}[\mathsf{TallyAlg}_1,\mathsf{DelAlg}_1]$ about an honest voter $\mathsf{V}^{(i)}_{\mathsf{fld}}$, S does the following:
  - Select $\mathbf{a'}_i^{[s+1]}\leftarrow\mathbb{Z}_q^{[s+1]}$, and compute $(\{\mathbf{b_{i,k}}^{[n]}\}^s_{k=1},\mathbf{b'_i}^{[e]})\leftarrow\mathsf{Decode}^{\mathsf{V}}_1(\mathbf{a'}_i^{[s+1]})$;
  - Select $\{\mathbf{c_{i,k}}^{[n]}\}^s_{k=1}\leftarrow(\mathbb{Z}_q)^{[n*s]}$, and $\mathbf{c'_i}^{[e]}\leftarrow(\mathbb{Z}_q)^{[e]}$;
  - For $k\in[s]$, compute $(\mathbf{C_{i,k}}^{[n]},\mathbf{C'_{i,k}}^{[n]})=\mathsf{LEG.Enc}_{\mathsf{pk}}(\mathbf{b_{i,k}}^{[n]};\mathbf{c_{i,k}}^{[n]})$;
  - Compute $(\mathbf{D_i}^{[e]},\mathbf{D'_i}^{[e]})=\mathsf{LEG.Enc}_{\mathsf{pk}}(\mathbf{b'_i}^{[e]};\mathbf{c'_i}^{[e]})$;
  - Simulate Valid Ballot NIZK proof σ'_i;

  - Post $(\{(\mathbf{C_{i,k}}^{[n]}, \mathbf{C'_{i,k}}^{[n]})\}_{k=1}^{s}, (\mathbf{D_i}^{[e]}, \mathbf{D'_i}^{[e]}), \sigma'_i, \eta_i)$ to $\mathcal{F}_{\text{BC}}$.

- Once the simulated ledger $\mathcal{F}_{\text{BC}}$ receives $(\{(\mathbf{A_{j,k}}^{[n]}, \mathbf{A'_{j,k}}^{[n]})\}_{k=1}^{s}, (\mathbf{b_j}^{[s]}, \mathbf{B'_j}^{[e]}), \sigma_j, \eta_j)$ from a corrupted voter $\mathsf{V}_{\mathsf{fld}}^{(j)}$, S decrypts the ciphertexts $(\{(\mathbf{A_{j,k}}^{[n]}, \mathbf{A'_{j,k}}^{[n]})\}_{k=1}^{s}, (\mathbf{b_j}^{[s]}, \mathbf{B'_j}^{[e]}))$ with the secret key shares of all honest voting committee members to get $\mathbf{a_j}^{[s+1]}$. S then sends $(\text{CAST}, \mathsf{sid}, \mathbf{a_j}^{[s+1]}, \eta_j)$ to $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$ on behalf of $\mathsf{V}_{\mathsf{fld}}^{(j)}$.

Expert Vote Phase:

- Upon receiving $(\text{VOTENOTIFY}, \mathsf{E}_{\mathsf{fld}}^{(j)}, \mathsf{sid})$ from $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$ about an honest expert $\mathsf{E}_{\mathsf{fld}}^{(j)}$, S does the following:
  - Select $\mathbf{b'_j}^{[s]} \leftarrow (\mathbb{Z}_q)^{[s]}$, compute $(\{\mathbf{p'_{j,k}}^{[n]}\}_{k=1}^{s}) \leftarrow \mathsf{Decode}_1^{\mathsf{E}}(\mathbf{b'_j}^{[s]})$;
  - Select $\{\mathbf{a_{j,k}}^{[n]}\}_{k=1}^{s} \leftarrow (\mathbb{Z}_q)^{[n*s]}$;
  - For $k \in [n]$, compute $(\mathbf{F_{j,k}}^{[n]}, \mathbf{F'_{j,k}}^{[n]}) := \mathsf{LEG.Enc}_{\mathsf{pk}}(\mathbf{p'_{j,k}}^{[n]}; \mathbf{a_{j,k}}^{[n]})$;
  - Simulate Valid Ballot NIZK proof δ'_j;
  - Post $(\{(\mathbf{F_{j,k}}^{[n]}, \mathbf{F'_{j,k}}^{[n]})\}_{k=1}^{s}, \delta'_j)$ to $\mathcal{F}_{\text{BC}}$.

- Once the simulated ledger $\mathcal{F}_{\text{BC}}$ receives $(\{(\mathbf{K_{j,k}}^{[n]}, \mathbf{K'_{j,k}}^{[n]})\}_{k=1}^{s}, \delta_j)$ from a corrupted expert $\mathsf{E}_{\mathsf{fld}}^{(j)}$, S decrypts the ciphertexts with the secret key shares of all honest voting committee members to get $\mathbf{b_j}^{[s]}$. S then sends $(\text{VOTE}, \mathsf{sid}, \mathbf{b_j}^{[s]})$ back to $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$ on behalf of $\mathsf{E}_{\mathsf{fld}}^{(j)}$.

Tally Phase:

- Upon receiving $(\text{CALDELNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ from $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$ about an honest voting committee member $\mathsf{C}^{(t)}$, S follows the protocol on behalf of $\mathsf{C}^{(t)}$ as if it receives the message from $\mathcal{Z}$, construct the valid voter and expert set $\mathsf{VI}_{\mathsf{fld}}^{[\alpha]}$ and $\mathsf{EI}_{\mathsf{fld}}^{[\beta]}$, and compute $\{m_j\}_{j=1}^{\beta}$;

- Upon receiving $(\text{TALLYNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ from $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$ about an honest voting committee member $\mathsf{C}^{(t)}$, S follows the protocol on behalf of $\mathsf{C}^{(t)}$ as if it receives the message from $\mathcal{Z}$, and post $(\{f_l\}_{l=1}^{n})$ to $\mathcal{F}_{\text{BC}}$;

- Once the simulated ledger $\mathcal{F}_{\text{BC}}$ receives $(\{f_l\}_{l=1}^{n})$ from a corrupted voting committee member $\mathsf{C}^{(t)}$, S sends $(\text{TALLY}, \mathsf{sid})$ back to $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$ on behalf of $\mathsf{C}^{(t)}$.

- Upon receiving $(\text{LEAKDEL}, \mathsf{sid}, \{D_j\}_{j=1}^{e})$ from $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$, S randomly select $\mu+1$ voting committee members denoted by R, and randomly select $\mathsf{C}_{\mathsf{fld}}^{(f)} \in \mathsf{R}$. For $j \in [e]$, compute $R := (\prod_{\mathsf{C}_{\mathsf{fld}}^{(k)} \in \mathsf{R} \setminus \{\mathsf{C}_{\mathsf{fld}}^{(f)}\}} R_k^{\mathcal{L}_k(0)})^{-1}$, where $R_K := (I_j)^{\mathsf{sk}_k}$, $\mathcal{L}_k(0)$ are Lagrange coefficients (Cf. Fig. 2.8), and compute $R'_f = (I'_j/(g^{D_j \cdot R}))^{-L_f(0)}$ as the new decryption share of $\mathsf{C}_{\mathsf{fld}}^{(f)}$ instead of $R_f := (I_j)^{\mathsf{sk}_f}$. S simulates the Valid Ballot NIZK proof σ_f about $\{(B_{i,j}, B'_{i,j})\}_{i=1}^{v}$;
- Upon receiving $(\text{LEAKCASTING}, \mathsf{sid}, \{f_l\}_{l=1}^{n})$ from $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$, S changes the decryption share for one of the voting committee member similarly to the last step.

Indistinguishability.

Proof. The indistinguishability is proven through a series of hybrid worlds $\mathcal{H}_0, \ldots, \mathcal{H}_6$.

Hybrid $\mathcal{H}_0$: It is the real protocol execution $\mathsf{EXEC}^{\mathcal{F}_{\text{BC}}, \mathcal{F}_{\text{DBKG}}^{c,\mu,1}}_{\Pi_{\text{VOTE1}}^{c,\mu,s,n}, \mathcal{A}, \mathcal{Z}}$.

Hybrid $\mathcal{H}_1$: $\mathcal{H}_1$ is the same as $\mathcal{H}_0$ except that in $\mathcal{H}_1$, during the Voters Cast Shortlist Phase, S posted different ciphertexts $\{(\mathbf{C_{i,k}}^{[n]}, \mathbf{C'_{i,k}}^{[n]})\}_{k=1}^{s}, (\mathbf{D_i}^{[e]}, \mathbf{D'_i}^{[e]})$ to ledger instead of real ciphertexts $\{(\mathbf{A_{i,k}}^{[n]}, \mathbf{A'_{i,k}}^{[n]})\}_{k=1}^{s}, (\mathbf{B_i}^{[e]}, \mathbf{B'_i}^{[e]})$. In addition, the messages $\mathbf{a_j}^{[s+1]}$ sent to $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$ by S are computed from the ciphertexts instead of the real messages.
Claim: If the lifted ElGamal encryption scheme is IND-CPA secure with adversarial advantage $\mathsf{Adv}_{\mathsf{LEG}}^{\mathsf{IND\text{-}CPA}}(1^\kappa, \mathcal{A})$, Valid Ballot NIZK is computational sound with adversarial advantage $\mathsf{Adv}_{\mathsf{NIZK},Ballot}^{\mathsf{Sound}}(1^\lambda, \mathcal{A})$, then $\mathcal{H}_1$ and $\mathcal{H}_0$ are indistinguishable with distinguishing advantage at most $(n \cdot s + e) \cdot \mathsf{Adv}_{\mathsf{LEG}}^{\mathsf{IND\text{-}CPA}}(1^\kappa, \mathcal{A}) + \mathsf{Adv}_{\mathsf{NIZK},Ballot}^{\mathsf{Sound}}(1^\lambda, \mathcal{A})$.
Proof: In $\mathcal{H}_1$, we have changed $n*s+e$ ciphertexts which encrypted random strings, therefore, if any adversary $\mathcal{A}$ can distinguish $\mathcal{H}_1$ from $\mathcal{H}_0$, then we can construct an adversary $\mathcal{B}$, who can break IND-CPA game of Lifted Elgamal encryption scheme. Additionally, the probability that adversary can submit incorrect ciphertexts while pass the soundness property of Valid Ballot NIZK proof is negligible, hence no adversary can differentiate the messages sent by S gaining from the ciphertexts and real messages. The overall adversary advantage in $\mathcal{H}_1$ is $(n \cdot s + e) \cdot \mathsf{Adv}_{\mathsf{LEG}}^{\mathsf{IND\text{-}CPA}}(1^\kappa, \mathcal{A}) + \mathsf{Adv}_{\mathsf{NIZK},Ballot}^{\mathsf{Sound}}(1^\lambda, \mathcal{A})$. ■

Hybrid $\mathcal{H}_2$: $\mathcal{H}_2$ is the same as $\mathcal{H}_1$ except that in $\mathcal{H}_2$, during the Voters Cast Shortlist Phase, S simulated Valid Ballot NIZK proof σ'_i.
Claim: $\mathcal{H}_2$ and $\mathcal{H}_1$ are perfectly indistinguishable.
Proof: Since Valid Ballot NIZK proof is perfect complete and perfect special honest verifier zero knowledge, if any adversary $\mathcal{A}$ can distinguish $\mathcal{H}_2$ from $\mathcal{H}_1$, then we can construct an adversary $\mathcal{B}$, who can break the ZK property of NIZK proof σ_i. ■

Hybrid $\mathcal{H}_3$: $\mathcal{H}_3$ is the same as $\mathcal{H}_2$ except that in $\mathcal{H}_3$, during the Experts Vote Phase, S posts new ciphertexts encrypted random messages $(\{\mathbf{F_{j,k}}^{[n]}, \mathbf{F'_{j,k}}^{[n]}\}_{k=1}^{s})$, S computes $\mathbf{b_j}^{[s]}$ from the ciphertexts and sends them to $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$.
Claim: If the lifted ElGamal encryption scheme is IND-CPA secure with adversarial advantage $\mathsf{Adv}_{\mathsf{LEG}}^{\mathsf{IND\text{-}CPA}}(1^\kappa, \mathcal{A})$, if the Valid Ballot NIZK is computational sound with adversarial advantage $\mathsf{Adv}_{\mathsf{NIZK},Ballot}^{\mathsf{Sound}}(1^\lambda, \mathcal{A})$, then $\mathcal{H}_3$ and $\mathcal{H}_2$ are indistinguishable with distinguishing advantage at most $(s \cdot n) \cdot \mathsf{Adv}_{\mathsf{LEG}}^{\mathsf{IND\text{-}CPA}}(1^\kappa, \mathcal{A}) + \mathsf{Adv}_{\mathsf{NIZK},Ballot}^{\mathsf{Sound}}(1^\lambda, \mathcal{A})$.
Proof: The proof is similar to the proof in $\mathcal{H}_1$. ■

Hybrid $\mathcal{H}_4$: $\mathcal{H}_4$ is the same as $\mathcal{H}_3$ except that in $\mathcal{H}_4$, during the Expert Vote Phase, S simulated Valid Ballot NIZK proof δ'_i
Claim: $\mathcal{H}_4$ and $\mathcal{H}_3$ are perfectly indistinguishable.
Proof: The proof is similar to the proof in $\mathcal{H}_2$. ■

Hybrid $\mathcal{H}_5$: $\mathcal{H}_5$ is the same as $\mathcal{H}_4$ except that in $\mathcal{H}_5$, during the Tally Phase, S computes an honest voting committee members' decryption shares based on the leaked delegation from $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$, and simulate its Valid Ballot NIZK proof.
Claim: $\mathcal{H}_5$ and $\mathcal{H}_4$ are perfectly indistinguishable.
Proof: Firstly, the decryption shares in these two worlds follow identical distribution. Secondly, similar to the proof in $\mathcal{H}_2$, $\mathcal{H}_5$ and $\mathcal{H}_4$ are perfectly indistinguishable. ■

Hybrid $\mathcal{H}_6$: $\mathcal{H}_6$ is the same as $\mathcal{H}_5$ except that in $\mathcal{H}_6$, during the Tally Phase, S computes an honest voting committee members' decryption shares based on the leaked tally from $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1, \mathsf{DelAlg}_1]$, and simulate related Valid Ballot NIZK proof.
Claim: $\mathcal{H}_6$ and $\mathcal{H}_5$ are perfectly indistinguishable.
Proof: The proof is similar to the proof in Hybrid $\mathcal{H}_5$. ■

The adversary's view of $\mathcal{H}_6$ is identical to the simulated view $\mathsf{EXEC}_{\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}[\mathsf{TallyAlg}_1,\mathsf{DelAlg}_1],\mathsf{S},\mathcal{Z}}$. Therefore, no $\mathsf{PPT}\,\mathcal{Z}$ can distinguish the view of the ideal execution from the view of the real execution with more than advantage

$$(2ns + e) \cdot \mathsf{Adv}_{\mathsf{LEG}}^{\mathsf{IND\text{-}CPA}}(1^\kappa, \mathcal{A}) + 2 \cdot \mathsf{Adv}_{\mathsf{NIZK},Ballot}^{\mathsf{Sound}}(1^\lambda, \mathcal{A}).$$

□

This concluded our proof of Theorem 14. □

6.3 Threshold Voting Construction

In this section, we introduce the Threshold Voting Functionality in Section 6.3.1 and the Threshold Voting Protocol in Section 6.3.2. These components are essential for constructing the Threshold Voting stage in the second phase of the TSV scheme. To assess the security of Threshold Voting, we analyse it within the UC framework, as detailed in Section 6.3.3.

6.3.1 Threshold Voting Functionality $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$

The Threshold Voting Functionality, denoted as $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ and illustrated in Figure 6.13, encompasses the functional tasks required for the Threshold Voting stage. These tasks include initialisation, voter cast, expert vote, and tally. $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ interacts with voters, denoted by $\mathcal{V}_{\mathsf{fld}}^{[v]} := \{\mathsf{V}_{\mathsf{fld}}^{(i)}\}_{i=1}^{v}$, experts denoted by $\mathcal{E}_{\mathsf{fld}}^{[e]} := \{\mathsf{E}_{\mathsf{fld}}^{(j)}\}_{j=1}^{e}$, the voting committee represented as $\mathcal{C}^{[c]} := \{\mathsf{C}^{(t)}\}_{t=1}^{c}$, where the threshold is μ, and the adversary denoted as S.

$\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ is parameterised with three committee flag sets, $\mathbf{C}_{key}$, $\mathbf{C}_{del}$, $\mathbf{C}_{tally}$, a valid voter casting set, $\mathbf{V}$, a valid expert voting set, $\mathbf{E}$, which are all set to $\emptyset$ initially, a delegation calculation algorithm, DelAlg_2, a tally algorithm, $\mathsf{TallyAlg}_2$, corrupted voting committee, $\mathcal{C}_{\mathsf{cor}}$, honest voting committee, $\mathcal{C}_{\mathsf{honest}}$, the size of the shortlist generated in Preferential Voting stage, s.

$\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ works as follows:

- **Initialisation Phase**.

  To initiate the voting, voting committee member, $\mathcal{C}^{[c]}$, sends message, $(\text{INIT}, \mathsf{sid})$, to $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$. $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ notifies S by $(\text{INITNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$. Voting process starts until all the voting committee members send the initialisation message, $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ sets $\mathbf{C}_{key} := \mathbf{C}_{key} \cup \{\mathsf{C}^{(t)}\}$, continue to next step until $|\mathbf{C}_{key}| = c$;

- **Voter Cast Phase**.

  Voter, $\mathsf{V}_{\mathsf{fld}}^{(i)} \in \mathcal{V}_{\mathsf{fld}}^{[v]}$, sends its ballots, $\mathbf{a_i}^{[s]}$, and voting power, η_i, to $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ by $(\text{CAST}, \mathsf{sid}, \mathbf{a_i}^{[s]}, \eta_i)$. $\mathbf{a_i}^{[s]}$ is voter's ballots. For example, in Figure 4.3-(b), $\mathbf{a_1}^{[2]} := \{\text{NO}, \text{YES}\}$ for $\mathsf{V}_f^{(1)}$. $\mathbf{a_2}^{[2]} := \{\mathsf{E}_F^{(1)}, \mathsf{E}_F^{(2)}\}$ for $\mathsf{V}_f^{(2)}$. $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ saves this message by set $\mathbf{V} := \mathbf{V} \cup \{(\mathsf{V}_{\mathsf{fld}}^{(i)}, \mathbf{a_j}^{[s]}, \eta_i)\}$, and notifies S by sending $(\text{CASTNOTIFY}, \mathsf{V}_{\mathsf{fld}}^{(i)}, \mathsf{sid}, \eta_i)$. If more than μ voting committee members are corrupted ($|\mathcal{C}_{\mathsf{cor}}| \geq \mu$), $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ additionally leaks voter's inputs to S by sending $(\text{LEAK}, \mathsf{V}_{\mathsf{fld}}^{(i)}, \text{CAST}, \mathsf{sid}, \mathbf{a_j}^{[s]}, \eta_i)$;

- **Expert Vote Phase**.

  Expert, $\mathsf{E}_{\mathsf{fld}}^{(j)} \in \mathcal{E}_{\mathsf{fld}}^{[e]}$, sends its ballots, $\mathbf{b_j}^{[s]}$, to $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ by $(\text{VOTE}, \mathsf{sid}, \mathbf{b_j}^{[s]})$. For example, in Figure 4.3-(b), $\mathbf{b_j}^{[s]} := \{\text{NO}, \text{NO}\}$ for $\mathsf{E}_f^{(1)}$. $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ sets $\mathbf{E} := \mathbf{E} \cup \{(\mathsf{E}_{\mathsf{fld}}^{(j)}, \mathbf{b_j}^{[s]})\}$, and send $(\text{VOTENOTIFY}, \mathsf{E}_{\mathsf{fld}}^{(j)}, \mathsf{sid})$ to S. If more than μ voting committee member are corrupted ($|\mathcal{C}_{\mathsf{cor}}| \geq \mu$), $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ additionally leaks expert's inputs to S;

- **Tally Phase**.

  - **Delegation Computation**.
    $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ firstly computes the delegation power of each expert, when it gets command, $(\text{CALDEL}, \mathsf{sid})$, from a voting committee member, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$. It sets $\mathbf{C}_{del} := \mathbf{C}_{del} \cup \{\mathsf{C}^{(t)}\}$ and sends $(\text{CALDELNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ to S. If there are more than μ voting committee members ($|\mathbf{C}_{del}| \geq \mu$), $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ can compute the delegation power (Figure 6.14) by

    $$\{D_{j,l}\}_{j=1,l=1}^{e,s} \leftarrow \mathsf{DelAlg}_2(s, e, \{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v}), \tag{6.24}$$

    where $D_{j,l}$ means the delegation power of expert, $\mathsf{Exp}_{\mathsf{fld}}^{(j)}$, for proposal P_l in the shortlist. Experts' delegation power are revealed to S if the corruption exceeds μ ($|\mathbf{C}_{del} \cap \mathcal{C}_{\mathsf{cor}}| \geq \mu$);

  - **Tally Computation**.
    $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ begins to compute the tally results of each proposal and notify S by $(\text{TALLYNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$, once it gets $(\text{TALLY}, \mathsf{sid})$ from a voting committee member, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$. $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ sets $\mathbf{C}_{tally} := \mathbf{C}_{tally} \cup \{\mathsf{C}^{(t)}\}$, when more than μ voting committee members ($|\mathbf{C}_{tally}| \geq \mu$) send $(\text{TALLY}, \mathsf{sid})$, $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ calculates tally results of each proposal (Figure 6.15) by

    $$\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{n} \leftarrow \mathsf{TallyAlg}_2(s, \gamma, \{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v}, \{\mathbf{b_j}^{[s]}\}_{j=1}^{e}, \{D_{j,l}\}_{j=1,l=1}^{e,s}), \tag{6.25}$$

    where $\{f_{l,1}, f_{l,2}, f_{l,3}\}$ means the number of votes for YES, NO and ABSTAIN respectively. The tally results are leaked to S if more than μ members are corrupted ($|C_{tally} \cap \mathcal{C}_{\mathsf{cor}}| \geq \mu$).

- Any party can read the tally results and experts' decisions by sending $(\text{READTALLY}, \mathsf{sid})$ and $(\text{REVEAL}, \mathsf{sid}, \mathsf{E}_{\mathsf{fld}}^{(j)})$ to $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$, $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ answers the requests by returning messages, $(\text{READTALLYRETURN}, \mathsf{sid}, \{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{n})$ and $(\text{REVEALEXPERT}, \mathsf{sid}, \mathbf{b_j}^{[s]})$ to the requester.

Voting Ideal Functionality $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$

$\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ interacts with voters $\mathcal{V}_{\mathsf{fld}}^{[v]} := \{\mathsf{V}_{\mathsf{fld}}^{(i)}\}_{i=1}^{v}$, experts $\mathcal{E}_{\mathsf{fld}}^{[e]} := \{\mathsf{E}_{\mathsf{fld}}^{(j)}\}_{j=1}^{e}$, voting committee $\mathcal{C}^{[c]} := \{\mathsf{C}^{(t)}\}_{t=1}^{c}$ of which the threshold is μ, and adversary S. $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ is parameterised with three committee flag sets $\mathbf{C}_{key}$, $\mathbf{C}_{del}$, $\mathbf{C}_{tally}$, a valid voter casting set $\mathbf{V}$, a valid expert voting set $\mathbf{E}$ which are all set to $\emptyset$ initially, and a voting stage index $\gamma \in \{1, 2\}$, a delegation calculation algorithm DelAlg_2, a tally algorithm $\mathsf{TallyAlg}_2$, corrupted voting committee $\mathcal{C}_{\mathsf{cor}}$, honest voting committee $\mathcal{C}_{\mathsf{honest}}$, the number of candidate proposals n, and the number of selected proposals s ($s \leq n$ in preferential voting, $s = 1$ in threshold voting).
$\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$ does the following:

Initialisation Phase:

- Upon receiving $(\text{INIT}, \mathsf{sid})$ from a voting committee member $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, send $(\text{INITNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ to S, and set $\mathbf{C}_{key} := \mathbf{C}_{key} \cup \{\mathsf{C}^{(t)}\}$, continue to next step until $|\mathbf{C}_{key}| = c$.

Voter Cast Phase:

- Upon receiving $(\text{CAST}, \mathsf{sid}, \mathbf{a_i}^{[s]}, \eta_i)$ from a voter $\mathsf{V}_{\mathsf{fld}}^{(i)} \in \mathcal{V}_{\mathsf{fld}}^{[v]}$, set $\mathbf{V} := \mathbf{V} \cup \{(\mathsf{V}_{\mathsf{fld}}^{(i)}, \mathbf{a_j}^{[s]}, \eta_i)\}$, and send $(\text{CASTNOTIFY}, \mathsf{V}_{\mathsf{fld}}^{(i)}, \mathsf{sid}, \eta_i)$ to S. Send $(\text{LEAK}, \mathsf{V}_{\mathsf{fld}}^{(i)}, \text{CAST}, \mathsf{sid}, \mathbf{a_j}^{[s]}, \eta_i)$ to S if $|\mathcal{C}_{\mathsf{cor}}| \geq \mu$.

Expert Vote Phase:

- Upon receiving $(\text{VOTE}, \mathsf{sid}, \mathbf{b_j}^{[s]})$ from an expert $\mathsf{E}_{\mathsf{fld}}^{(j)} \in \mathcal{E}_{\mathsf{fld}}^{[e]}$, set $\mathbf{E} := \mathbf{E} \cup \{(\mathsf{E}_{\mathsf{fld}}^{(j)}, \mathbf{b_j}^{[s]})\}$, and send $(\text{VOTENOTIFY}, \mathsf{E}_{\mathsf{fld}}^{(j)}, \mathsf{sid})$ to S. Send $(\text{LEAK}, \mathsf{E}_{\mathsf{fld}}^{(j)}, \text{VOTE}, \mathsf{sid}, \mathbf{b_j}^{[s]})$ to S if $|\mathcal{C}_{\mathsf{cor}}| \geq \mu$.

Tally Phase:

- Upon receiving $(\text{CALDEL}, \mathsf{sid})$ from a voting committee member $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, does the following:
  - Set $\mathbf{C}_{del} := \mathbf{C}_{del} \cup \{\mathsf{C}^{(t)}\}$, send $(\text{CALDELNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ to S;
  - If $|\mathbf{C}_{del}| \geq \mu$, compute $\{D_{j,l}\}_{j=1,l=1}^{e,s} \leftarrow \mathsf{DelAlg}_2(s, e, \{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v})$, Cf. Figure 6.14;
  - Send $(\text{LEAKDEL}, \mathsf{sid}, \{D_{j,l}\}_{j=1,l=1}^{e,s})$ to S if $|\mathbf{C}_{del} \cap \mathcal{C}_{\mathsf{cor}}| \geq \mu$.
- Upon receiving $(\text{TALLY}, \mathsf{sid})$ from a voting committee member $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, does the following:
  - Set $\mathbf{C}_{tally} := \mathbf{C}_{tally} \cup \{\mathsf{C}^{(t)}\}$, send $(\text{TALLYNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ to S;
  - If $|\mathbf{C}_{tally}| \geq \mu$, compute $\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{n} \leftarrow \mathsf{TallyAlg}_2(s, \gamma, \{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v}, \{\mathbf{b_j}^{[s]}\}_{j=1}^{e}, \{D_{j,l}\}_{j=1,l=1}^{e,s})$, Cf. Figure 6.15. If $|C_{tally} \cap \mathcal{C}_{\mathsf{cor}}| \geq \mu$, send $(\text{LEAKCASTING}, \mathsf{sid}, \{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{n})$ to S.
- Upon receiving $(\text{READTALLY}, \mathsf{sid})$ from any party, return $(\text{READTALLYRETURN}, \mathsf{sid}, \{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{n})$ to the requester;
- Upon receiving $(\text{REVEAL}, \mathsf{sid}, \mathsf{E}_{\mathsf{fld}}^{(j)})$ from any party, return $(\text{REVEALEXPERT}, \mathsf{sid}, \mathbf{b_j}^{[s]})$ to the requester.

Figure 6.13: The ideal functionality $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}$.

In Figure 6.13, we use two algorithms in the Tally phase: Delegation Calculation Algorithm to compute experts' delegation power for each proposal in the shortlist, and Tally Calculation Algorithm to compute tally results regarding to the number of votes for YES, NO and ABSTAIN, based on voters' ballots and voting power, experts' ballots and delegation power.

- **Delegation Calculation Algorithm,** $\mathsf{DelAlg}_2(s, e, \{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v})$**.**

  As shown in Figure 6.14, delegation calculation algorithm takes the size of the shortlist generated in Preferential Voting stage, s, the number of experts, e, voters' ballots and voting power, $\{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v}$ as inputs and outputs experts' delegation power $\{D_{j,l}\}_{j=1,l=1}^{e,s}$.

  For $i \in [v]$, voter's ballot, $\mathbf{a_i}^{[s]}$, is parsed to s vectors with size $3+e$ denoted by $\{w_{i,l,k}\}_{l=1,k=1}^{s,3+e}$. For each shortlisted proposal, $\{P_l\}_{l\in[s]}$, for $j \in [e]$, $l \in [s]$, expert's delegation power regarding to each proposal can be computed by

$$D_{j,l} := \sum_{i=1}^{v} w_{i,l,j+2} \cdot \eta_i. \tag{6.26}$$

Algorithm $\mathsf{DelAlg}_2(s, e, \{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v})$

Input:

- The size of the shortlist generated in Preferential Voting stage, s;
- The number of experts, e;
- Voters' ballots and voting power $\{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v}$.

Delegation Calculation:

- For $i \in [v]$, parse $\mathbf{a_i}^{[s]}$ to $(\{w_{i,l,k}\}_{l=1,k=1}^{s,3+e})$;
- For $j \in [e]$, $l \in [s]$, compute $D_{j,l} := \sum_{i=1}^{v} w_{i,l,j+2} \cdot \eta_i$.

Output: Experts' delegation power: $\{D_{j,l}\}_{j=1,l=1}^{e,s}$.

Figure 6.14: Delegation Calculation Algorithm in Threshold Voting.

- **Tally Calculation Algorithm,** $\mathsf{TallyAlg}_2(s, \{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v}, \{\mathbf{b_j}^{[s]}\}_{j=1}^{e}, \{D_{j,l}\}_{j=1,l=1}^{e,s})$**.**

  Tally calculation algorithm in Figure 6.15 computes tally results of each proposal, $\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{n}$, based on voters' and experts' ballots. It takes the size of the shortlist generated in Preferential Voting stage, s, voters' ballots and voting power, $\{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v}$,

experts' ballots and voting power, $\{\mathbf{b_j}^{[s]}\}_{j=1}^{e}, \{D_{j,l}\}_{j=1,l=1}^{e,s}$ as inputs, and outputs tally results of each proposal, $\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{n}$.

To mitigate the potential dominance of wealthier voters in decision-making processes due to their larger locked stakes, a sigmoid function is applied to the original voting power distribution $\{\eta_i\}_{i=1}^{v}$ and $\{\{D_{j,l}\}_{j=1,l=1}^{e,s}$. This mathematical adjustment ensures that voting influence increases at a diminishing rate, promoting a more balanced representation irrespective of financial status.

Voter's ballot, $\mathbf{a_i}^{[s]}$, is parsed to s shortlist vectors with size, $3+e$, denoted by $\{w_{i,l,k}\}_{l=1,k=1}^{s,3+e}$ for $i \in [v]$. Expert's ballot, $\mathbf{b_j}^{[s]}$, is parsed to s shortlist vectors with size, 2, denoted by $\{q_{j,l,k}\}_{l=1,k=1}^{s,2}$ for $j \in [e]$. Define ϵ as a smoothing factor, for $l \in [n]$, the tally result of each proposal is computed by

$$
\begin{aligned}
f_{l,1} &:= (\sum_{i=1}^{v} w_{i,l,1} \cdot \frac{1}{1+\epsilon^{-\eta_i}}) + (\sum_{j=1}^{e} q_{j,l,1} \cdot \frac{1}{1+\epsilon^{-D_{j,l}}}), \\
f_{l,2} &:= (\sum_{i=1}^{v} w_{i,l,2} \cdot \frac{1}{1+\epsilon^{-\eta_i}}) + (\sum_{j=1}^{e} q_{j,l,2} \cdot \frac{1}{1+\epsilon^{-D_{j,l}}}), \\
f_{l,3} &:= (\sum_{i=1}^{v} w_{i,l,3} \cdot \frac{1}{1+\epsilon^{-\eta_i}}) + (\sum_{j=1}^{e} q_{j,l,3} \cdot \frac{1}{1+\epsilon^{-D_{j,l}}}),
\end{aligned}
\tag{6.27}
$$

where $f_{l,1}$ is the number of "YES" votes, $f_{l,2}$ is the number of "NO" votes, and $f_{l,3}$ is the number of "ABSTAIN" votes, for the l-th proposal.

Algorithm $\mathsf{TallyAlg}_2(s, \{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v}, \{\mathbf{b_j}^{[s]}\}_{j=1}^{e}, \{D_{j,l}\}_{j=1,l=1}^{e,s})$

Input: The number of candidate proposals n, the number of selected proposals s, the voting stage index γ, voters' ballots and voting power $\{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v}$, experts' ballots and voting power $\{\mathbf{b_j}^{[s]}\}_{j=1}^{e}, \{D_{j,l}\}_{j=1,l=1}^{e,s}$, ϵ as a smoothing factor.

Tally Calculation:

- Parse $\mathbf{a_i}^{[s]}$ to $(\{w_{i,l,k}\}_{l=1,k=1}^{s,3+e})$ for $i \in [v]$;
- Parse $\mathbf{b_j}^{[s]}$ to $(\{q_{j,l,k}\}_{l=1,k=1}^{s,2})$ for $j \in [e]$;
- For $l \in [n]$, compute the following:
  - $f_{l,1} := (\sum_{i=1}^{v} w_{i,l,1} \cdot \frac{1}{1+\epsilon^{-\eta_i}}) + (\sum_{j=1}^{e} q_{j,l,1} \cdot \frac{1}{1+\epsilon^{-D_{j,l}}})$;
  - $f_{l,2} := (\sum_{i=1}^{v} w_{i,l,2} \cdot \frac{1}{1+\epsilon^{-\eta_i}}) + (\sum_{j=1}^{e} q_{j,l,2} \cdot \frac{1}{1+\epsilon^{-D_{j,l}}})$;
  - $f_{l,3} := (\sum_{i=1}^{v} w_{i,l,3} \cdot \frac{1}{1+\epsilon^{-\eta_i}}) + (\sum_{j=1}^{e} q_{j,l,3} \cdot \frac{1}{1+\epsilon^{-D_{j,l}}})$.
- For $l \in [n]$, set $\mathbf{f_l} := \{f_{l,1}, f_{l,2}, f_{l,3}\}$.

Output: Tally results of each proposal: $\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{n}$.

Figure 6.15: Tally Calculation Algorithm.

6.3.2 Threshold Voting Protocol $\Pi_{\mathbf{VOTE}2}^{c,\mu,s,n}$

In this stage, voters and experts need to vote on this shortlist and generate the final winning proposals. Let the size of the size of the shortlist generated in Preferential Voting stage be s, assume that disqualified voters and experts who didn't follow $\Pi_{\text{VOTE1}}^{c,\mu,s,n}$ have been banned from participating the second stage. Let $\mathsf{TallyAlg}_2$ be short for $\mathsf{TallyAlg}_2(s, \{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v}, \{\mathbf{b_j}^{[s]}\}_{j=1}^{e}, \{D_{j,l}\}_{j=1,l=1}^{e,s})$, and DelAlg_2 be short for $\mathsf{DelAlg}_2(s, e, \{\mathbf{a_i}^{[s]}, \eta_i\}_{i=1}^{v})$. Threshold Voting protocol, $\Pi_{\text{VOTE2}}^{c,\mu,s,n}$, is presented in Figure 6.18, Figure 6.19 and Figure 6.20 to *UC-realise* $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}[\mathsf{DelAlg}_2, \mathsf{TallyAlg}_2]$ in $\{\mathcal{F}_{\text{BC}}, \mathcal{F}_{\text{DBKG}}^{c,\mu,s}\}$-*hybrid world*.

- **Initialisation Phase**.

  To start with, $\mathcal{Z}$ sends $(\text{INIT}, \mathsf{sid})$ to voting committee member, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, and initiates threshold voting. $\mathsf{C}^{(t)}$ sends $(\text{KEYGEN}, \mathsf{sid}, \mathsf{C}^{(t)})$ to $\mathcal{F}_{\text{DBKG}}^{c,\mu,s}$ for initialising DKG process, and then sends $(\text{READKEYSHARE}, \mathsf{sid}, \mathsf{C}^{(t)})$ to $\mathcal{F}_{\text{DBKG}}^{c,\mu,s}$ asking for partial secret keys. $\mathsf{C}^{(t)}$ gets $(\text{READKEYSHARERETURN}, \mathsf{sid}, \{\mathsf{psk}_{l,i}\}_{l=1}^{s})$ from $\mathcal{F}_{\text{DBKG}}^{c,\mu,s}$.

- **Voter Cast Phase**.

The voter, $\mathsf{V}_{\mathsf{fld}}^{(i)} \in \mathcal{V}_{\mathsf{fld}}^{[v]}$, begins to vote once it gets $(\textsc{Cast}, \mathsf{sid}, \mathbf{v_i}, \eta_i)$ from $\mathcal{Z}$. $\mathsf{V}_{\mathsf{fld}}^{(i)}$ gets s public keys from $\mathcal{F}_{\text{DBKG}}^{c,\mu,s}$ to encrypt its ballots. $\mathsf{V}_{\mathsf{fld}}^{(i)}$ sends $(\textsc{ReadPK}, \mathsf{sid})$ to $\mathcal{F}_{\text{DBKG}}^{c,\mu,s}$, and receives $(\textsc{ReadPKReturn}, \mathsf{sid}, \{\mathsf{gpk}_l\}_{l=1}^{s}, \{\mathsf{ppk}_{l,a}\}_{l=1,a=1}^{s,c})$. Afterwards, $\mathsf{V}_{\mathsf{fld}}^{(i)}$ parses $\mathbf{v_i}$ from $\mathcal{Z}$ to $(\{\mathbf{w_{i,l}}^{[3+e]}\}_{l=1}^{s})$, and encrypts the ballots for $l \in [s]$,

$$(\mathbf{X_{i,l}}^{[3+e]}, \mathbf{X'_{i,l}}^{[3+e]}) := \mathsf{LEG.Enc}_{\mathsf{gpk}_l}(\mathbf{w_{i,l}}^{[3+e]}, \mathbf{r_{i,l}}^{[3+e]}), \tag{6.28}$$

where $\{\mathbf{r_{i,l}}^{[3+e]}\}_{l=1}^{s}$ are randomly selected for encryption. Moreover, $\mathsf{V}_{\mathsf{fld}}^{(i)}$ needs to prove that it either voted one choice from $\textsc{Yes}, \textsc{No}, \textsc{Abstain}$ or delegated to one expert for each proposal in the shortlist. More specifically, Figure 6.16 gives an example to show how $\mathsf{V}_{\mathsf{fld}}^{(i)}$ proves its ballot, it generates Unit Vector NIZK proof ([13]) to prove that for $l \in [s]$, the cipher-texts, $(\mathbf{X_{i,l}}^{[3+e]}, \mathbf{X'_{i,l}}^{[3+e]})$, encrypt a unit vector (only one element in the vector is 1, the rest are 0). Then $\mathsf{V}_{\mathsf{fld}}^{(i)}$ posts the ciphertexts, NIZK proof and its voting power, $(\{(\mathbf{X_{i,l}}^{[3+e]}, \mathbf{X'_{i,l}}^{[3+e]})\}_{l=1}^{s}, \Delta_i, \eta_i)$, to $\mathcal{F}_{\text{BC}}$.

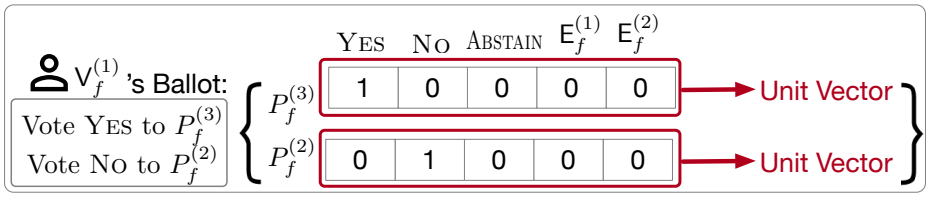

Figure 6.16: Voter's Ballot NIZK Example ($s = 2, e = 2$) in Threshold Voting.

- **Expert Vote Phase**.

  Once $\mathcal{Z}$ sends $(\textsc{Vote}, \mathsf{sid}, \mathbf{b_j}^{[s]})$, the expert $\mathsf{E}_{\mathsf{fld}}^{(j)} \in \mathcal{E}_{\mathsf{fld}}^{[e]}$ votes $\mathbf{b_j}^{[s]}$. It gets public keys, $(\textsc{ReadPKReturn}, \mathsf{sid}, \{\mathsf{gpk}_l\}_{l=1}^{s}, \{\mathsf{ppk}_{l,a}\}_{l=1,a=1}^{s,c})$, from $\mathcal{F}_{\text{DBKG}}^{c,\mu,s}$ by asking $(\textsc{ReadPK}, \mathsf{sid})$. Then the ballots is generated by parsing $\mathbf{v_j}^{[s]}$ to $(\{\mathbf{q_{j,l}}^{[3]}\}_{l=1}^{s}$. Last, $\mathsf{E}_{\mathsf{fld}}^{(j)}$ encrypts the ballots,

$$(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]}) = \mathsf{LEG.Enc}_{\mathsf{gpk}_l}(\mathbf{q'_{j,l}}^{[3]}; \mathbf{r'_{j,l}}^{[3]}) \text{ for } l \in [s] \tag{6.29}$$

where $\{\mathbf{r'_{j,l}}^{[3]}\}_{l=1}^{s}$ are random. Additionally, $\mathsf{E}_{\mathsf{fld}}^{(j)}$ should prove that it only voted one choice from $\textsc{Yes}, \textsc{No}, \textsc{Abstain}$ (See example in Figure 6.17) and generate Unit Vector Encryption NIZK proof γ_j to prove $(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]})$ encrypts a unit vector for $l \in [s]$. Then it posts $(\{(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]})\}_{l=1}^{s}, \gamma_j)$ to $\mathcal{F}_{\text{BC}}$.

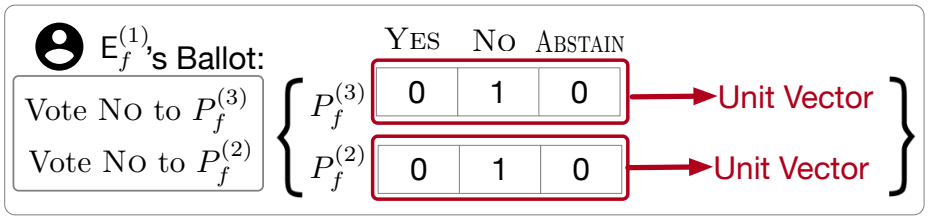

Figure 6.17: Expert's Ballot NIZK Example in Threshold Voting.

- **Tally Phase**.

  In the Tally Phase, voting committee member, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, fetches encrypted ballots from experts and voters,

$$\{(\{(\mathbf{X_{i,l}}^{[3+e]}, \mathbf{X'_{i,l}}^{[3+e]})\}_{l=1}^{s}, \Delta_i, \eta_i)\}_{i=1}^{v}$$
$$\{(\{(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]})\}_{l=1}^{s}, \gamma_j)\}_{j=1}^{e},$$

  once it gets $(\textsc{CalDel}, \mathsf{sid})$ from $\mathcal{Z}$.

  - **Delegation Calculation**.
    $\mathsf{C}^{(t)}$ continue to compute the delegation power until it validate the ciphertexts and NIZK proofs, by checking if $\mathsf{Verify}(\{(\mathbf{X_{i,l}}^{[3+e]}, \mathbf{X'_{i,l}}^{[3+e]})\}_{l=1}^{s}, \Delta_i, \eta_i) = 1$ for $i \in [v]$, and $\mathsf{Verify}(\{(\{(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]})\}_{l=1}^{s}, \gamma_j) = 1$ for $j \in [e]$. It removes all the invalid and repeated casting ballots, and sets $\mathsf{VI}_{\mathsf{fld}}^{[v']}$, $\mathsf{EI}_{\mathsf{fld}}^{[e]}$ as a set of voter/expert index in new ascending order who provided valid ballots. The rest ciphertexts are denoted by $\{(\{(\mathbf{X_{i,l}}^{[3+e']}, \mathbf{X'_{i,l}}^{[3+e']})\}_{l=1}^{s}, \Delta_i, \eta_i)\}_{i=1}^{v'}$ and $\{(\{(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]})\}_{j=1}^{e'}$. Next, $\mathsf{C}^{(t)}$ compute the encrypted delegation power,

$$\begin{aligned} I_{j,l} &:= \prod_{i=1}^{v'} (X_{i,l,3+j})^{\eta_i} \text{ for } j \in [e'], \text{ for } l \in [s], \\ I'_{j,l} &:= \prod_{i=1}^{v'} (X'_{i,l,3+j})^{\eta_i} \text{ for } j \in [e'], \text{ for } l \in [s]. \end{aligned} \tag{6.30}$$

    Afterwards, $\mathcal{C}^{[c]}$ jointly decrypt $(I_{j,l}, I'_{j,l})$ to delegation power, $m_{j,l}$, for $j \in [e']$, for $l \in [s]$.

  - **Tally Calculation**. When $\mathcal{Z}$ sends $(\textsc{Tally}, \mathsf{sid})$ to compute the tally results, for

$l \in [s]$, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$ compute

$$
\begin{aligned}
S_{l,1} &:= (\prod_{i=1}^{v'}(X_{i,l,1})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y_{i,l,1})^{m_{j,l}}),\\
S'_{l,1} &:= (\prod_{i=1}^{v'}(X'_{i,l,1})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y'_{i,l,1})^{m_{j,l}}),\\
S_{l,2} &:= (\prod_{i=1}^{v'}(X_{i,l,2})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y_{i,l,2})^{m_{j,l}})\\
S'_{l,2} &:= (\prod_{i=1}^{v'}(X'_{i,l,2})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y'_{i,l,2})^{m_{j,l}})\\
S_{l,3} &:= (\prod_{i=1}^{v'}(X_{i,l,3})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y_{i,l,3})^{m_{j,l}})\\
S'_{l,3} &:= (\prod_{i=1}^{v'}(X'_{i,l,3})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y'_{i,l,3})^{m_{j,l}})
\end{aligned}
\tag{6.31}
$$

For $l \in [s]$, $\mathcal{C}^{[c]}$ jointly decrypt $\{(S_{l,1}, S'_{l,1}), (S_{l,2}, S'_{l,2}), (S_{l,3}, S'_{l,3})\}$ to $\{f_{l,1}, f_{l,2}, f_{l,3}\}$, and post the final tally results, $(\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{s})$, to $\mathcal{F}_{\text{BC}}$.

Once an expert $\mathsf{E}_{\mathsf{fld}}^{(j)} \in \mathcal{E}_{\mathsf{fld}}^{[e]}$ gets $(\text{REVEAL}, \mathsf{sid}, \mathsf{E}_{\mathsf{fld}}^{(j)})$ from $\mathcal{Z}$, it posts $\{\mathbf{r_{j,k}}^{[s]}, \mathbf{p_{j,k}}^{[s]}\}_{k=1}^{s}$ to $\mathcal{F}_{\text{BC}}$, and returns $(\text{REVEALEXPERT}, \mathsf{sid}, \{\mathbf{r_{j,k}}^{[s]}, \mathbf{p_{j,k}}^{[s]}\}_{k=1}^{s})$ to $\mathcal{Z}$. $\mathcal{Z}$ can ask any party about the tally results by sending $(\text{READTALLY}, \mathsf{sid})$, the party gets the tally results from blockchain and returns $(\text{READTALLYRETURN}, \mathsf{sid}, (\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{s})$ to $\mathcal{Z}$.

Stage 2: Threshold Voting protocol $\Pi^{c,\mu,s,n}_{\text{VOTE2}}$ (Part 1)

Initialisation Phase:

- Upon receiving $(\text{INIT}, \mathsf{sid})$ from $\mathcal{Z}$, the voting committee member, $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$, sends $(\text{KEYGEN}, \mathsf{sid}, \mathsf{C}^{(t)})$ and $(\text{READKEYSHARE}, \mathsf{sid}, \mathsf{C}^{(t)})$ to $\mathcal{F}^{c,\mu,s}_{\text{DBKG}}$, and receives $(\text{READKEYSHARERETURN}, \mathsf{sid}, \{\mathsf{psk}_{l,i}\}^{s}_{l=1})$ from $\mathcal{F}^{c,\mu,s}_{\text{DBKG}}$.

Voter Cast Phase:

- Upon receiving $(\text{CAST}, \mathsf{sid}, \mathbf{v_i}, \eta_i)$ from $\mathcal{Z}$, the voter, $\mathsf{V}^{(i)}_{\mathsf{fld}} \in \mathcal{V}^{[v]}_{\mathsf{fld}}$, does the following:
  - Send $(\text{READPK}, \mathsf{sid})$ to $\mathcal{F}^{c,\mu,s}_{\text{DBKG}}$;
  - Receive $(\text{READPKRETURN}, \mathsf{sid}, \{\mathsf{gpk}_l\}^{s}_{l=1}, \{\mathsf{ppk}_{l,a}\}^{s,c}_{l=1,a=1})$ from $\mathcal{F}^{c,\mu,s}_{\text{DBKG}}$;
  - Parse $\mathbf{v_i}$ to $(\{\mathbf{w_{i,l}}^{[3+e]}\}^{s}_{l=1})$;
  - Select $\{\mathbf{r_{i,l}}^{[3+e]}\}^{s}_{l=1} \leftarrow (\mathcal{Z}_p)^{[(3+e)*s]}$;
  - For $l \in [s]$, compute $(\mathbf{X_{i,l}}^{[3+e]}, \mathbf{X'_{i,l}}^{[3+e]}) := \mathsf{LEG.Enc}_{\mathsf{gpk}_l}(\mathbf{w_{i,l}}^{[3+e]}, \mathbf{r_{i,l}}^{[3+e]})$;
  - Generate Unit Vector Encryption NIZK proof, Δ_i, to prove $(\mathbf{X_{i,l}}^{[3+e]}, \mathbf{X'_{i,l}}^{[3+e]})$ encrypts a unit vector for $l \in [s]$;
  - Send $(\mathsf{Write}, \mathsf{sid}, (\{(\mathbf{X_{i,l}}^{[3+e]}, \mathbf{X'_{i,l}}^{[3+e]})\}^{s}_{l=1}, \Delta_i, \eta_i))$ to $\mathcal{F}_{\text{BC}}$.

Expert Vote Phase:

- Upon receiving $(\text{VOTE}, \mathsf{sid}, \mathbf{b_j}^{[s]})$ from $\mathcal{Z}$, the expert, $\mathsf{E}^{(j)}_{\mathsf{fld}} \in \mathcal{E}^{[e]}_{\mathsf{fld}}$, does the following:
  - Send $(\text{READPK}, \mathsf{sid})$ to $\mathcal{F}^{c,\mu,s}_{\text{DBKG}}$;
  - Receive $(\text{READPKRETURN}, \mathsf{sid}, \{\mathsf{gpk}_l\}^{s}_{l=1}, \{\mathsf{ppk}_{l,a}\}^{s,c}_{l=1,a=1})$ from $\mathcal{F}^{c,\mu,s}_{\text{DBKG}}$;
  - Parse $\mathbf{v_j}^{[s]}$ to $(\{\mathbf{q_{j,l}}^{[3]}\}^{s}_{l=1}$;
  - Select $\{\mathbf{r'_{j,l}}^{[3]}\}^{s}_{l=1} \leftarrow (\mathcal{Z}_p)^{[s*3]}$;
  - For $l \in [s]$, compute $(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]}) = \mathsf{LEG.Enc}_{\mathsf{gpk}_l}(\mathbf{q'_{j,l}}^{[3]}; \mathbf{r'_{j,l}}^{[3]})$;
  - Generate Unit Vector Encryption NIZK proof, γ_j, to prove $(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]})$ encrypts a unit vector for $l \in [s]$;
  - Post $(\mathsf{Write}, \mathsf{sid}, (\{(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]})\}^{s}_{l=1}, \gamma_j))$ to $\mathcal{F}_{\text{BC}}$.

Figure 6.18: Stage 2: Threshold Voting protocol $\Pi^{c,\mu,s,n}_{\text{VOTE2}}$ $\{\mathcal{F}_{\text{BC}}, \mathcal{F}^{c,\mu,s}_{\text{DBKG}}\}$-*hybrid world* (Part 1).

Stage 2: Threshold Voting protocol $\Pi^{c,\mu,s,n}_{\text{VOTE2}}$ (Part 2)

Tally Phase:

- Upon receiving $(\text{CALDEL}, \mathsf{sid})$ from $\mathcal{Z}$, the voting committee member $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$ does the following:
  - Send $(\mathsf{Read}, \mathsf{sid})$ to $\mathcal{F}_{\text{BC}}$;
  - Receive $\{(\{(\mathbf{X_{i,l}}^{[3+e]}, \mathbf{X'_{i,l}}^{[3+e]})\}_{l=1}^{s}, \Delta_i, \eta_i)\}_{i=1}^{v}$ and $\{(\{(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]})\}_{l=1}^{s}, \gamma_j)\}_{j=1}^{e}$ from $\mathcal{F}_{\text{BC}}$;
  - Check if $\mathsf{Verify}(\{(\mathbf{X_{i,l}}^{[3+e]}, \mathbf{X'_{i,l}}^{[3+e]})\}_{l=1}^{s}, \Delta_i, \eta_i) = 1$ for $i \in [v]$, remove all the invalid casting ballots. If there are repeated ciphertexts in $\{(\{(\mathbf{X_{i,l}}^{[3+e]}, \mathbf{X'_{i,l}}^{[3+e]})\}_{l=1}^{s}, \Delta_i, \eta_i)\}_{i=1}^{v}$, remove all the repeated casting ballots except the first one sent to $\mathcal{F}_{\text{BC}}$. Set $\mathsf{VI}_{\mathsf{fld}}^{[v']}$ as a set of voter index in new ascending order who provided valid ballots;
  - Check if $\mathsf{Verify}(\{(\{(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]})\}_{l=1}^{s}, \gamma_j) = 1$ for $j \in [e]$, remove all the invalid voting ballots. If there are repeated ciphertexts in $\{(\{(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]})\}_{l=1}^{s})\}_{j=1}^{e}$, remove all the repeated voting ballots except the first one sent to $\mathcal{F}_{\text{BC}}$. Set $\mathsf{EI}_{\mathsf{fld}}^{[e]}$ as a set of voter index in new ascending order who provided valid ballots;
  - Remove the ciphertexts sent by experts/voters, and sent to invalid experts, denote the rest ciphertexts by $\{(\{(\mathbf{X_{i,l}}^{[3+e']}, \mathbf{X'_{i,l}}^{[3+e']})\}_{l=1}^{s}, \Delta_i, \eta_i)\}_{i=1}^{v'}$ and $\{(\{(\mathbf{Y_{j,l}}^{[3]}, \mathbf{Y'_{j,l}}^{[3]})\}_{j=1}^{e'}$;
  - For $j \in [e']$, for $l \in [s]$, compute $I_{j,l} := \prod_{i=1}^{v'}(X_{i,l,3+j})^{\eta_i}$, $I'_{j,l} := \prod_{i=1}^{v'}(X'_{i,l,3+j})^{\eta_i}$;
  - For $j \in [e']$, for $l \in [s]$, $\mathcal{C}^{[c]}$ jointly compute $m_{j,l} := \mathsf{LEG.Dec}(I_{j,l}, I'_{j,l})$.

Figure 6.19: Stage 2: Threshold Voting protocol $\Pi^{c,\mu,s,n}_{\text{VOTE2}}$ $\{\mathcal{F}_{\text{BC}}, \mathcal{F}^{c,\mu,s}_{\text{DBKG}}\}$-*hybrid world* (Part 2).

Stage 2: Threshold Voting protocol $\Pi^{c,\mu,s,n}_{\text{VOTE2}}$ (Part 3)

Tally Phase:

- Upon receiving $(\text{TALLY}, \mathsf{sid})$ from $\mathcal{Z}$, the voting committee member $\mathsf{C}^{(t)} \in \mathcal{C}^{[c]}$ does the following:
  - For $l \in [s]$, compute the following:
    - $S_{l,1} := (\prod_{i=1}^{v'}(X_{i,l,1})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y_{i,l,1})^{m_{j,l}})$,
      $S'_{l,1} := (\prod_{i=1}^{v'}(X'_{i,l,1})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y'_{i,l,1})^{m_{j,l}})$;
    - $S_{l,2} := (\prod_{i=1}^{v'}(X_{i,l,2})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y_{i,l,2})^{m_{j,l}})$,
      $S'_{l,2} := (\prod_{i=1}^{v'}(X'_{i,l,2})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y'_{i,l,2})^{m_{j,l}})$;
    - $S_{l,3} := (\prod_{i=1}^{v'}(X_{i,l,3})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y_{i,l,3})^{m_{j,l}})$,
      $S'_{l,3} := (\prod_{i=1}^{v'}(X'_{i,l,3})^{\eta_i}) \cdot (\prod_{j=1}^{e'}(Y'_{i,l,3})^{m_{j,l}})$;
  - For $l \in [s]$, $\mathcal{C}^{[c]}$ jointly compute the following:
    - $f_{l,1} := \mathsf{LEG.Dec}(S_{l,1}, S'_{l,1})$;
    - $f_{l,2} := \mathsf{LEG.Dec}(S_{l,2}, S'_{l,2})$;
    - $f_{l,3} := \mathsf{LEG.Dec}(S_{l,3}, S'_{l,3})$.
  - Post $(\mathsf{Write}, \mathsf{sid}, (\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{s}))$ to $\mathcal{F}_{\text{BC}}$.
- Upon receiving $(\text{REVEAL}, \mathsf{sid}, \mathsf{E}^{(j)}_{\mathsf{fld}})$ from $\mathcal{Z}$, the expert, $\mathsf{E}^{(j)}_{\mathsf{fld}} \in \mathcal{E}^{[e]}_{\mathsf{fld}}$, sends $(\mathsf{Write}, \mathsf{sid}, (\{\mathbf{r_{j,k}}^{[s]}, \mathbf{p_{j,k}}^{[s]}\}_{k=1}^{s}))$ to $\mathcal{F}_{\text{BC}}$, and sends $(\text{REVEALEXPERT}, \mathsf{sid}, \{\mathbf{r_{j,k}}^{[s]}, \mathbf{p_{j,k}}^{[s]}\}_{k=1}^{s})$ to $\mathcal{Z}$;
- Upon receiving $(\text{READTALLY}, \mathsf{sid})$ from $\mathcal{Z}$, the party P sends $(\mathsf{Read}, \mathsf{sid}, (\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{s}))$ to $\mathcal{F}_{\text{BC}}$, and sends $(\text{READTALLYRETURN}, \mathsf{sid}, (\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{s})$ to $\mathcal{Z}$.

Figure 6.20: Stage 2: Threshold Voting protocol $\Pi^{c,\mu,s,n}_{\text{VOTE2}}$ $\{\mathcal{F}_{\text{BC}}, \mathcal{F}^{c,\mu,s}_{\text{DBKG}}\}$-*hybrid world* (Part 3).

6.3.3 Security Analysis of Threshold Voting

Theorem 15 (Threshold Voting)**.** *Assume Unit Vector* NIZK *is perfect complete, perfect special honest verifier zero knowledge, and computational sound with adversary advantage of* $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Unit}(1^\kappa, \mathcal{A})$*. Assume Lifted Elgamal encryption* Enc *is IND-CPA secure with adversary advantage of* $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A})$*. The protocol* $\Pi^{c,\mu,s,n}_{\text{VOTE2}}$ *in Figure 6.18, Figure 6.19 and Figure 6.20 UC-realise* $\mathcal{F}^{c,\mu,s}_{\text{VOTE2}}[\mathsf{DelAlg}_2, \mathsf{TallyAlg}_2]$ *in Figure 6.13 in* $\{\mathcal{F}^{c,\mu,s}_{\text{DBKG}}, \mathcal{F}_{\text{BC}}\}$-*hybrid world against a static adversary with distinguishing advantage*

$$((3+e)\cdot s + 3s)\cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A}) + 2\cdot \mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Unit}(1^\kappa, \mathcal{A}).$$

To prove theorem 15, we construct a PPT simulator S such that no nonuniform PPT environment $\mathcal{Z}$ can distinguish between 1) the ideal execution $\mathsf{EXEC}_{\mathcal{F}_{\text{VOTE}}^{c,\mu,n,s},\mathsf{S},\mathcal{Z}}$ where the parties interact with functionality $\mathcal{F}_{\text{VOTE}}^{c,\mu,n,s}$ in the ideal world and corrupted parties are controlled by the simulator S; and 2) the real execution $\mathsf{EXEC}^{\mathcal{F}_{\text{BC}},\mathcal{F}_{\text{DBKG}}^{c,\mu,s}}_{\Pi_{\text{VOTE2}}^{c,\mu,n,s},\mathcal{A},\mathcal{Z}}$ where the voters $\mathcal{V}_{\mathsf{fld}}^{[v]} := \{\mathsf{V}_{\mathsf{fld}}^{(i)}\}_{i=1}^{v}$, experts $\mathcal{E}_{\mathsf{fld}}^{[e]} := \{\mathsf{E}_{\mathsf{fld}}^{(j)}\}_{j=1}^{e}$, and voting committee $\mathcal{C}^{[c]} := \{\mathsf{C}^{(t)}\}_{t=1}^{c}$ run protocol $\Pi_{\text{VOTE2}}^{c,\mu,s,n}$ in the $\{\mathcal{F}_{\text{BC}}, \mathcal{F}_{\text{DBKG}}^{c,\mu,s}\}$-*hybrid world* and the corrupted parties are controlled by a dummy adversary $\mathcal{A}$ who simply forwards messages from/to $\mathcal{Z}$.

Simulator. The simulator S internally runs $\mathcal{A}$, forwarding messages to/from the environment $\mathcal{Z}$. The simulator S simulates the following interactions with $\mathcal{A}$: The simulator S simulates the followings interactions with $\mathcal{Z}$:

Initialisation Phase:

- Upon receiving $(\text{INITNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ from $\mathcal{F}_{\text{VOTE}}^{c,\mu,n,s}$ about an honest voting committee member $\mathsf{C}^{(t)} \in \mathcal{C}_{\mathsf{honest}}$, S uses simulated $\mathcal{F}_{\text{DBKG}}^{c,\mu,s}$ to generate $(\{\mathsf{gpk}_l, \mathsf{gsk}_{l,t}\}_{l=1}^{s})$ for $\mathsf{C}^{(t)}$.

Voter Cast Phase:

- Upon receiving $(\text{CASTNOTIFY}, \mathsf{V}_{\mathsf{fld}}^{(i)}, \mathsf{sid}, \eta_i)$ from $\mathcal{F}_{\text{VOTE}}^{c,\mu,n,s}$ about an honest voter $\mathsf{V}_{\mathsf{fld}}^{(i)}$, S does the following:
  - Select $\{\mathsf{w}_{\mathsf{i},\mathsf{l}}{}^{[3+e]}\}_{l=1}^{s}, \{\mathsf{r}_{\mathsf{i},\mathsf{l}}{}^{[3+e]}\}_{l=1}^{s} \leftarrow \mathbb{Z}_q^{[(3+e)\cdot 2\cdot s]}$;
  - Compute $(\mathsf{A}_{\mathsf{i},\mathsf{l}}{}^{[3+e]}, \mathsf{A}'_{\mathsf{i},\mathsf{l}}{}^{[3+e]}) = \mathsf{Enc}_{\mathsf{pk}_l}(\mathsf{w}_{\mathsf{i},\mathsf{l}}{}^{[3+e]}; \mathsf{r}_{\mathsf{i},\mathsf{l}}{}^{[3+e]})$ for $l \in [s]$;
  - Simulate Unit Vector NIZK proof Δ'_i;
  - Post $(\{(\mathsf{A}_{\mathsf{i},\mathsf{l}}{}^{[3+e]}, \mathsf{A}'_{\mathsf{i},\mathsf{l}}{}^{[3+e]})\}_{l=1}^{s}, \Delta'_i, \eta_i)$ to $\mathcal{F}_{\text{BC}}$.
- Once the simulated ledger $\mathcal{F}_{\text{BC}}$ receives $(\{(\mathsf{X}_{\mathsf{j},\mathsf{l}}{}^{[3+e]}, \mathsf{X}'_{\mathsf{j},\mathsf{l}}{}^{[3+e]})\}_{l=1}^{s}, \Delta_j, \eta_j)$ from a corrupted voter $\mathsf{V}_{\mathsf{fld}}^{(j)}$, S decrypts the ciphertexts $(\mathsf{X}_{\mathsf{j},\mathsf{l}}{}^{[3+e]}, \mathsf{X}'_{\mathsf{j},\mathsf{l}}{}^{[3+e]})$ with the secret key shares of all honest voting committee members to get $\mathsf{v}_{\mathsf{j}}{}^{[s+e]}$. S then sends $(\text{CASTSHORTLIST}, \mathsf{sid}, \mathsf{v}_{\mathsf{j}}{}^{[s+e]}, \eta_j)$ back to $\mathcal{F}_{\text{VOTE}}^{c,\mu,n,s}$ on behalf of $\mathsf{V}_{\mathsf{fld}}^{(j)}$.

Experts Vote Shortlist Phase:

- Upon receiving $(\text{VOTENOTIFY}, \mathsf{E}_{\mathsf{fld}}^{(j)}, \mathsf{sid})$ from $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}[\mathsf{DelAlg}_2, \mathsf{TallyAlg}_2]$ about an honest expert $\mathsf{E}_{\mathsf{fld}}^{(j)}$, S does the following:
  - Select $\{\{\mathsf{q}'_{\mathsf{j},\mathsf{l}}{}^{[3]}\}_{l=1}^{s}, \{\mathsf{r}'_{\mathsf{j},\mathsf{l}}{}^{[3]}\}_{l=1}^{s}\} \leftarrow (\mathbb{Z}_q)^{[s\cdot 6]}$;
  - Compute $(\mathsf{P}_{\mathsf{j},\mathsf{l}}{}^{[3]}, \mathsf{P}'_{\mathsf{j},\mathsf{l}}{}^{[3]}) = \mathsf{Enc}_{\mathsf{pk}_l}(\mathsf{q}'_{\mathsf{j},\mathsf{l}}{}^{[3]}; \mathsf{r}'_{\mathsf{j},\mathsf{l}}{}^{[3]})$ for $l \in [s]$;

  - Simulate Unit Vector NIZK proof γ'_j;
  - Post $(\{(\mathsf{P_{j,l}}^{[3]}, \mathsf{P'_{j,l}}^{[3]})\}_{l=1}^{s}, \delta'_j)$ to $\mathcal{F}_{\text{BC}}$.

- Once the simulated ledger $\mathcal{F}_{\text{BC}}$ receives $(\{(\mathsf{Y_{j,l}}^{[3]}, \mathsf{Y'_{j,l}}^{[3]})\}_{l=1}^{s}, \gamma_j)$ from a corrupted expert $\mathsf{E}_{\mathsf{fld}}^{(j)}$, S decrypts the ciphertexts with the secret key shares of all honest voting committee members to get $\mathsf{b_j}^{[s]}$. S then sends $(\text{VOTESHORTLIST}, \mathsf{sid}, \mathsf{b_j}^{[s]})$ back to $\mathcal{F}_{\text{VOTE}}^{c,\mu,n,s}$ on behalf of $\mathsf{E}_{\mathsf{fld}}^{(j)}$.

Tally Phase:

- Upon receiving $(\text{CALDELNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ from $\mathcal{F}_{\text{VOTE}}^{c,\mu,n,s}$ about an honest voting committee member $\mathsf{C}^{(t)}$, S follows the protocol on behalf of $\mathsf{C}^{(t)}$ as if it receives the message from $\mathcal{Z}$, construct the valid voter and expert set $\mathsf{VI}_{\mathsf{fld}}^{[v']}$ and $\mathsf{EI}_{\mathsf{fld}}^{[e]}$, and compute $\{m_{j,l}\}_{j=1,l=1}^{e,s}$;

- Upon receiving $(\text{TALLYNOTIFY}, \mathsf{sid}, \mathsf{C}^{(t)})$ from $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}[\mathsf{DelAlg}_2, \mathsf{TallyAlg}_2]$ about an honest voting committee member $\mathsf{C}^{(t)}$, S follows the protocol on behalf of $\mathsf{C}^{(t)}$ as if it receives the message from $\mathcal{Z}$, and post $(\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{s})$ to $\mathcal{F}_{\text{BC}}$;

- Once the simulated ledger $\mathcal{F}_{\text{BC}}$ receives $(\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{s})$ from a corrupted voting committee member $\mathsf{C}^{(t)}$, S sends $(\text{TALLY}, \mathsf{sid})$ back to $\mathcal{F}_{\text{VOTE}}^{c,\mu,n,s}$ on behalf of $\mathsf{C}^{(t)}$.

- Upon receiving $(\text{LEAKDEL}, \mathsf{sid}, \{\mathsf{D_j}\}_{j=1}^{e})$ from $\mathcal{F}_{\text{VOTE}}^{c,\mu,n,s}$, S computes $\{(I_{j,l}, I'_{j,l})\}_{j=1,l=1}^{e,s}$ following the protocol based on $\{(\{(\mathsf{X_{i,l}}^{[3+e]}, \mathsf{X'_{i,l}}^{[3+e]})\}_{l=1}^{s}, \Delta_i, \eta_i)\}_{i=1}^{v'}$ and $\{(\{(\mathsf{Y_{j,l}}^{[3]}, \mathsf{Y'_{j,l}}^{[3]})\}_{j=1}^{e}$, then it computes new decryption share of one honest committee member according to the $(I_{j,l}, I'_{j,l})$ to $m_{j,l}$. S simulates the Unit Vector NIZK proof σ_f about $\{(\{(\mathsf{X_{i,l}}^{[3+e]}, \mathsf{X'_{i,l}}^{[3+e]})\}_{l=1}^{s}, \Delta_i, \eta_i)\}_{i=1}^{v'}$ and $\{(\{(\mathsf{Y_{j,l}}^{[3]}, \mathsf{Y'_{j,l}}^{[3]})\}_{j=1}^{e}$;

- Upon receiving $(\text{LEAKCASTING}, \mathsf{sid}, (\{f_{l,1}, f_{l,2}, f_{l,3}\}_{l=1}^{s}))$ from $\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}[\mathsf{DelAlg}_2, \mathsf{TallyAlg}_2]$, S changes the decryption share for one of the voting committee member similarly to the last step.

Indistinguishability.

Proof. The indistinguishability is proven through a series of hybrid worlds $\mathcal{H}_0, \ldots, \mathcal{H}_7$.

Hybrid $\mathcal{H}_0$: It is the real protocol execution $\mathsf{EXEC}^{\mathcal{F}_{\text{BC}}, \mathcal{F}_{\text{DBKG}}^{c,\mu,s}}_{\Pi_{\text{VOTE2}}^{c,\mu,n,s}, \mathcal{A}, \mathcal{Z}}$.

Hybrid $\mathcal{H}_1$: $\mathcal{H}_1$ is the same as $\mathcal{H}_0$ except that in $\mathcal{H}_1$, during the Voters Cast Shortlist Phase, S posted different ciphertexts $(\{(\mathsf{A_{i,l}}^{[3+e]}, \mathsf{A'_{i,l}}^{[3+e]})\}_{l=1}^{s})$ to ledger instead of real ciphertexts $(\{(\mathsf{X_{i,l}}^{[3+e]}, \mathsf{X'_{i,l}}^{[3+e]})\}_{l=1}^{s})$. In addition, the messages $\mathsf{v_i}^{[s+e]}$ sent to $\mathcal{F}_{\text{VOTE}}^{c,\mu,n,s}$ by S are computed from the ciphertexts instead of the real messages.

Claim: If the lifted ElGamal encryption scheme is IND-CPA secure with adversarial advantage $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A})$, Unit Vector$\mathsf{NIZK}$ is computational sound with adversarial advantage $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Unit}(1^\lambda, \mathcal{A})$, then $\mathcal{H}_2$ and $\mathcal{H}_1$ are indistinguishable with distinguishing advantage at most $((3+e)\cdot s)\cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A}) + \mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Unit}(1^\lambda, \mathcal{A})$.
Proof: In $\mathcal{H}_2$, we have changed $(3+e)\cdot s$ ciphertexts which encrypted random strings, therefore, if any adversary $\mathcal{A}$ can distinguish $\mathcal{H}_2$ from $\mathcal{H}_1$, then we can construct an adversary $\mathcal{B}$, who can break IND-CPA game of Lifted Elgamal encryption scheme. Additionally, the probability that adversary can submit incorrect ciphertexts while pass the soundness property of Unit Vector NIZK proof is negligible, hence no adversary can differentiate the messages sent by S gaining from the ciphertexts and real messages. The overall adversary advantage in $\mathcal{H}_2$ is $((3+e)\cdot s)\cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A}) + \mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Unit}(1^\lambda, \mathcal{A})$. ■

Hybrid $\mathcal{H}_2$: $\mathcal{H}_2$ is the same as $\mathcal{H}_1$ except that in $\mathcal{H}_2$, during the Voters Cast Shortlist Phase, S simulated Unit Vector NIZK proof Δ'_i.
Claim: $\mathcal{H}_2$ and $\mathcal{H}_1$ are perfectly indistinguishable.
Proof: Since Unit Vector NIZK proof is perfect complete and perfect special honest verifier zero knowledge, if any adversary $\mathcal{A}$ can distinguish $\mathcal{H}_2$ from $\mathcal{H}_1$, then we can construct an adversary $\mathcal{B}$, who can break the ZK property of NIZK proof Δ_i. ■

Hybrid $\mathcal{H}_3$: $\mathcal{H}_3$ is the same as $\mathcal{H}_2$ except that in $\mathcal{H}_3$, during the Experts Vote Phase, S posts new ciphertexts encrypted random messages $(\{(\mathsf{P}_{\mathsf{j,l}}{}^{[3]}, \mathsf{P}'_{\mathsf{j,l}}{}^{[3]})\}^s_{l=1})$, S computes $\mathsf{b_j}^{[s]}$ from the ciphertexts and sends them to $\mathcal{F}^{c,\mu,n,s}_{\text{VOTE}}$.
Claim: If the lifted ElGamal encryption scheme is IND-CPA secure with adversarial advantage $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A})$, if the Unit Vector NIZK is computational sound with adversarial advantage $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Unit}(1^\lambda, \mathcal{A})$, then $\mathcal{H}_4$ and $\mathcal{H}_3$ are indistinguishable with distinguishing advantage at most $(3s)\cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A}) + \mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Unit}(1^\lambda, \mathcal{A})$.
Proof: The proof is similar to the proof in $\mathcal{H}_2$. ■

Hybrid $\mathcal{H}_4$: $\mathcal{H}_4$ is the same as $\mathcal{H}_3$ except that in $\mathcal{H}_4$, during the Expert Vote Phase, S simulated Unit Vector NIZK proof δ'_i.
Claim: $\mathcal{H}_4$ and $\mathcal{H}_3$ are perfectly indistinguishable.
Proof: The proof is similar to the proof in $\mathcal{H}_3$. ■

Hybrid $\mathcal{H}_5$: $\mathcal{H}_5$ is the same as $\mathcal{H}_4$ except that in $\mathcal{H}_5$, during the Tally Phase, S computes an honest voting committee members' decryption shares based on the leaked delegation from $\mathcal{F}^{c,\mu,n,s}_{\text{VOTE}}$, and simulate its Unit Vector NIZK proof.
Claim: $\mathcal{H}_5$ and $\mathcal{H}_4$ are perfectly indistinguishable.
Proof: Firstly, the decryption shares in these two worlds follow identical distribution. Secondly, similar to the proof in $\mathcal{H}_2$, $\mathcal{H}_5$ and $\mathcal{H}_4$ are perfectly indistinguishable. ■

Hybrid $\mathcal{H}_6$: $\mathcal{H}_6$ is the same as $\mathcal{H}_5$ except that in $\mathcal{H}_6$, during the Tally Phase, S computes an honest voting committee members' decryption shares based on the leaked tally from $\mathcal{F}_{\text{VOTE}}^{c,\mu,n,s}$, and simulate related Unit VectorNIZK proof.
Claim: $\mathcal{H}_6$ and $\mathcal{H}_5$ are perfectly indistinguishable.
Proof: The proof is similar to the proof in Hybrid $\mathcal{H}_5$. ■

The adversary's view of $\mathcal{H}_6$ is identical to the simulated view $\mathsf{EXEC}_{\mathcal{F}_{\text{VOTE2}}^{c,\mu,s}[\mathsf{DelAlg}_2,\mathsf{TallyAlg}_2],\mathsf{S},\mathcal{Z}}$. Therefore, no $\mathsf{PPT}\,\mathcal{Z}$ can distinguish the view of the ideal execution from the view of the real execution with more than advantage

$$((3+e)\cdot s+3s)\cdot \mathsf{Adv}_{\mathsf{Enc}}^{\mathsf{IND\text{-}CPA}}(1^{\kappa},\mathcal{A})+2\cdot \mathsf{Adv}_{\mathsf{NIZK},Unit}^{\mathsf{Sound}}(1^{\lambda},\mathcal{A}).$$

This concluded our proof of Theorem 15. □

6.4 Summary

This chapter delved into the intricate design of the Two Stage Voting (TSV) scheme, a critical component of the envisioned privacy-preserving decision-making system during the voting epoch. With meticulous attention to detail, we commenced by thoroughly exploring the construction of the first stage, Preferential Voting, which plays a pivotal role in the TSV scheme.

To facilitate the Preferential Voting stage, we crafted the Preferential Voting functionality, meticulously incorporating the necessary functions for this stage to succeed. Two key algorithms were introduced, enabling the computation of both the delegation power of experts and the tally results for each proposal. These algorithms form the backbone of the Preferential Voting functionality, ensuring accurate and reliable results.

Subsequently, we laid out the Preferential Voting protocol, a well-crafted and robust implementation that faithfully realises the Preferential Voting functionality while effectively resisting static corruption. The protocol's design leverages cryptographic techniques and zero-knowledge proofs to ensure the integrity and privacy of the voting process.

Moving on, we expounded upon the functionality and protocol for the second stage of the TSV scheme, aptly named Threshold Voting. This stage further solidifies the foundation of the TSV scheme, ensuring a comprehensive and secure voting process. The Threshold Voting functionality and protocol were presented in a meticulous manner, accompanied by a rigorous security analysis with UC framework.

In the Preferential Voting stage, every voter computes its ballots with $\mathcal{O}(n \cdot s + e)$ cost. The cost of Valid Casting NIZK proof for voter's ballot validation is $\mathcal{O}(s \cdot \log(n+e) + n + s)$ for proving, verifier's computation cost is $\mathcal{O}(s \cdot \log(n+e))$. An expert costs $\mathcal{O}(n \cdot s)$ to generate ballots, the cost of Valid Voting NIZK proof for expert's ballot validation is $\mathcal{O}(s \cdot \log n + n + s)$ for proving, $\mathcal{O}(s \cdot \log n)$ for computation on verifier's side. In the Tally phase, the computation cost is $\mathcal{O}(e \cdot s)$, communication cost is $\mathcal{O}(s)$. Hence, overall cost of $\Pi^{c,\mu,s,n}_{\text{VOTE1}}$ is $\mathcal{O}(v \cdot s \cdot n)$. Similarly, the overall cost of $\Pi^{c,\mu,s,n}_{\text{VOTE2}}$ is $\mathcal{O}(v \cdot s \cdot e)$.

Through the seamless integration of the TSV scheme, its key components, and the groundbreaking Evolving Committee mechanisms, our ultimate goal is to establish a decision-making system that embodies the principles of robustness, privacy-preservation, and trustworthiness. By ensuring the highest standards of accuracy, fairness, and security, we aim to provide a reliable and transparent platform for decision-making that fosters collective intelligence and democratic participation.

In the security framework of our voting system, the DDH assumption plays a crucial role, particularly in the implementation of the lifted ElGamal encryption scheme utilized throughout the system. This assumption, fundamental to ensuring the confidentiality and integrity of the voting process, posits that it is computationally hard to distinguish tuples of the form $(g, g^a, g^b, g^{ab}))$ from random tuples of the form $(g, g^a, g^b, g^c))$, where g is a generator of a group and a, b, c are randomly chosen exponents.

Within our system, several theorems rely on the strength of the DDH assumption to prove the security properties of the cryptographic protocols employed. For instance, one key theorem asserts that under the DDH assumption, an adversary cannot distinguish between the encryptions of two different votes, thus providing semantic security for the voting protocol. This is particularly important in ensuring that the system resists potential attacks where an adversary attempts to analyse encrypted votes to determine voter choices.

Further, the integration of ZKPs with the lifted ElGamal encryption mechanism enhances this security feature. The ZKPs are designed to confirm the proper format and validity of encrypted votes without revealing any substantive data about the vote's content, thereby leveraging the DDH assumption to prevent information leakage during the verification process. Our rigorous security analysis includes empirical simulations and theoretical evaluations that demonstrate the robustness of the DDH assumption in maintaining the system's integrity against a variety of attack vectors, affirming its suitability in our cryptographic setting.

The study presented in [241] exposes critical vulnerabilities in the standard application of the ElGamal encryption scheme, especially when the size of the plaintexts does not adequately utilise the order of the group. This research underscores that selecting weak parameters p and g, or deficiencies in the implementation process, can drastically compromise the foundational security assumptions of the encryption method. These insights are pivotal for ensuring the security and integrity of cryptographic implementations.

To mitigate the issues related to standard ElGamal encryption, our Lifted variant introduces modifications in the encryption process. These modifications are specifically designed to address potential weaknesses, such as susceptibility to known plaintext attacks and the exploitation of the scheme's homomorphic properties. By carefully engineering these elements, our protocol enhances the ciphertext's resistance to various attack vectors, thereby preserving the confidentiality and integrity of the data even under advanced cryptographic attacks.

Ultimately, by adopting Lifted ElGamal encryption and adhering to strict cryptographic best practices, we safeguard against the vulnerabilities highlighted in the referenced study. This approach not only secures our cryptographic operations but also reinforces the overall reliability of our digital systems, ensuring that they remain secure against both current and future cryptographic challenges.

Voting committee members play a paramount role in the voting epoch, as they are entrusted with generating the cryptographic keys that guarantee the privacy and verifiability of the ballots. However, in practical scenarios, voting epochs can extend over prolonged periods, sometimes lasting as long as 30 days. During such extended durations, voting committee members are expected to remain committed and active until the entire decision-making process concludes. This requirement presents challenges and drawbacks, as it may lead to an increased risk of corruption among voting committee members due to prolonged exposure and potential external influences.

In response to this critical concern, the forthcoming chapter will introduce a groundbreaking solution known as the Evolving Committee mechanisms. This innovative element aims to enable voting committee replacement, providing a dynamic and efficient approach to manage committee members throughout the voting epoch. By adopting Evolving Committee mechanisms, the proposed decision-making system can maintain a fresh and reliable committee, reducing the risk of potential corruption and ensuring the continued integrity and security of the voting process.

The next chapter will delve into the intricacies of the Evolving Committee functionality, carefully outlining its key features and functionalities. Additionally, a meticulously designed protocol will be presented, highlighting how the Evolving Committee mechanisms effectively facilitate the replacement of committee members. As always, a comprehensive security analysis will be conducted within the robust UC framework, providing an in-depth assessment of the mechanisms' resilience and efficacy.

Chapter 7

Building Block: Evolving Committee Mechanism

> Empty your mind. Be formless, shapeless, like water. You put water into a cup, it becomes the cup. You put water into a bottle, it becomes the bottle.
>
> Bruce Lee

7.1 Overview

In Chapter 5 and Chapter 6, we emphasised the crucial role of the voting committee in the overall voting process. The voting committee is responsible for generating the global key pair(s), which includes the public keys utilised by both voters and experts to encrypt their respective ballots during the voting epoch. Additionally, the voting committee employs their partial secret keys to jointly compute the final tally results, ensuring the integrity and accuracy of the voting outcome.

However, in practice, voting epochs can span extended durations, sometimes lasting up to a month. During this prolonged period, the voting committee members are required to remain online and actively hold their partial secret keys. This extended commitment introduces inherent risks, as the longer the voting committee remains in operation, the greater the exposure to potential corruption or the risk of losing critical secret keys.

To address this critical concern and enhance the security and resilience of the voting process, we propose the concept of an evolving committee. The evolving committee mechanism enables the seamless replacement of voting committee members in each voting round. This dynamic approach ensures that committee members' involvement is limited to specific voting rounds, reducing the time they need to remain online and hold partial secret

keys. By periodically replacing committee members, the risk of potential corruption or key loss is significantly mitigated.

The evolving committee mechanism is thoughtfully designed to ensure a smooth transition between committee members without compromising the security or integrity of the voting process. Each new committee is carefully selected, ensuring that the replacement process does not introduce vulnerabilities or disruptions.

By adopting the evolving committee mechanism, the proposed decision-making system gains substantial benefits, including heightened security, reduced risk exposure, and enhanced trustworthiness. Furthermore, this dynamic approach aligns with the best practices in modern cryptographic protocols, providing a resilient and efficient solution for managing voting committee members throughout the voting epoch. Ultimately, the evolving committee mechanism reinforces the overall strength and reliability of the proposed privacy-preserving decision-making system, providing stakeholders with a robust and trustworthy platform for democratic participation and collective intelligence.

The concept of the evolving property, originally introduced in works like [242, 243, 244], revolves around the idea of sharing secrets among a set of (potentially infinite) participants whose identities are not known in advance. In each round of the process, a dealer is only required to distribute shares to newly arriving participants, enabling seamless integration of new members into the system.

Building upon the principles of evolving secret sharing, we propose an innovative evolving committee mechanism for the management of voting committee members in each round of the voting epoch. The evolving committee offers a dynamic approach to change voting committee members throughout the voting process, ensuring enhanced security, reduced exposure to risk, and improved efficiency.

To implement the evolving committee in the proposed decision-making system, the first voting committee is initially online in the first round. During this round, it performs the crucial task of generating the global key pairs using the $\Pi^{n,t,m}_{\mathrm{DBKG}}$ protocol, as described in Chapter 5. From the second round onwards, there will be two voting committees actively operating: the current voting committee from the previous round and the incoming voting committee for the next round. The outgoing voting committee shares the identities of the incoming voting committee and re-shares its partial secret keys to ensure consistency in the global secret key.

Throughout the voting epochs, when tasks necessitate the involvement of the voting committee in the Two Stage Voting Scheme, the corresponding voting committee for the specific round will respond promptly and appropriately following the prescribed protocol.

By adopting the evolving committee mechanism, the proposed decision-making system benefits from a flexible and adaptive approach to handle the changing composition of the voting committee. This mechanism ensures the smooth integration of new committee members, avoids prolonged commitments, and minimises the potential risks associated with extended involvement of committee members. As a result, the evolving committee

reinforces the overall security and reliability of the voting process, contributing to a robust and trustworthy platform for privacy-preserving, verifiable, and democratic decision-making.

Our approach to managing the evolving committee shares foundational ideas with dynamic-committee models such as those presented in CHURP [245] and Paralysis Proofs [246] but introduces significant innovations and simplifications that enhance security and flexibility.

CHURP and our model both utilise the concept where two groups hold the same secret with periodic changes in group membership. However, our model diverges fundamentally in its implementation by employing Pedersen commitments rather than the Kate-Zaverucha-Goldberg (KZG) commitment scheme used in CHURP. The choice of Pedersen commitments is strategic; it eliminates the need for a trusted setup, which is required in KZG, thereby reducing our system's assumptions and increasing its robustness. CHURP's model requires peer-to-peer reliable channels for secure communication, whereas our system leverages public, on-chain channels that enhance transparency and accessibility. Furthermore, CHURP's framework requires a bivariate polynomial of degree $(t, 2t)$, where t is the corruption threshold. In contrast, our system uses a simpler scheme with a t polynomial for reconstruction and a t' polynomial for resharing, where t and t' are the thresholds of the current and next committee, respectively. This modification simplifies the management of committee transitions and provides a clearer, more manageable security model analysed under the universal composability framework.

The Paralysis Proofs framework outlines a Dynamic Access Structure System (DASS) that, similar to our approach, allows for flexible updating of access structures without relying on a trusted third party. However, their system integrates a trust anchor such as a trusted execution environment (TEE) to address censorship resistance. While powerful, TEEs involve significant security limitations, particularly regarding availability and susceptibility to manipulation by malicious operating systems, as highlighted by their dependency on potentially compromised I/O operations. Our model does not rely on TEEs, sidestepping these vulnerabilities and focusing instead on blockchain-based solutions that do not require external trust anchors and are less susceptible to specific attack vectors like state rollbacks and timing attacks.

By circumventing the need for trusted setups and complex polynomial configurations, our model not only simplifies the technical framework but also enhances the security and integrity of the voting process. The use of public blockchain channels ensures that all transactions and committee changes are transparent and verifiable by all participants, promoting an open and democratic governance system. This robust and simplified approach provides a significant contribution to the field of cryptographic protocol design, particularly in the context of decentralised decision-making systems.

In cryptographic protocols where participant identities are publicly known, adaptive or proactive adversaries can exploit this information to devise more effective corruption strategies, potentially compromising the security of the system. To mitigate this risk and

reduce the adversary's influence on the protocols, role assignment techniques have been proposed [18, 247, 248]. These techniques aim to conceal the identities of future participants, making it harder for the adversary to target specific individuals.

In this context, we introduce a novel and dynamic approach called the Dynamic Scalable Distributed Key Management (DSKM) scheme. At the heart of the DSKM scheme lies the Designation Procedure, complemented by a Committee-Based Assembly line (CBA line) inspired by player-replaceability [249, 250]. This innovative combination ensures anonymity and scalability in the management of distributed key pairs.

The Designation Procedure begins with the self-selection of a Selection Committee, which leverages cryptographic sortition techniques. The Selection Committee then nominates the Maintenance Committee using anonymous encryption methods. This process effectively hides the identities of future participants, allowing for secure and robust key management.

In the CBA line, the Maintenance Committees play a crucial role in generating and maintaining global distributed key pairs. Unlike conventional approaches that involve the participation of all members, the DSKM scheme achieves scalability by only engaging a small portion of participants in the key generation process. As a result, both the Selection Committee and the Maintenance Committee are only required to be online once, regardless of the total number of participants. This unique feature significantly reduces communication overhead and computational complexity, making the DSKM scheme more efficient and practical.

The anonymous nomination of Maintenance Committees by the Selection Committee further enhances the scalability of the DSKM scheme. This property ensures that the threshold of the scheme can scale proportionally with the number of participants, maintaining a high level of security even in large-scale settings.

Overall, the DSKM scheme presents a versatile and robust solution for managing distributed key pairs in cryptographic protocols. By combining role assignment techniques, cryptographic sortition, and anonymous nomination, the DSKM scheme achieves privacy, scalability, and security, making it an ideal choice for privacy-preserving, verifiable, and decentralized decision-making systems.

In the remainder of this chapter, we delve into the construction of the evolving committee in Section 7.2. We outline the design of the ideal functionality, the protocol, and conduct a comprehensive security analysis to ensure the robustness and integrity of the evolving committee mechanism.

Additionally, in Section 7.3, we present a Dynamic Scalable Key Management Scheme. This scheme addresses the challenge of managing distributed key pairs in large-scale cryptographic protocols, achieving scalability and efficiency while maintaining a high level of security.

Through the combination of evolving committee and dynamic scalable key management, we aim to establish a comprehensive and practical solution for privacy-preserving, verifiable, and decentralised decision-making systems. These innovative mechanisms are crucial

building blocks in realising our vision for a trustworthy, privacy-centric, and reliable decision-making system. By incorporating these components into the Two Stage Voting (TSV) scheme introduced earlier, we can further enhance the overall security and efficiency of the decision-making process.

7.2 Evolving Committee Construction

In this section, we introduce the Evolving Functionality, denoted as $\mathcal{F}_{Evolving}[\mathbb{G}]$, which facilitates the dynamic changing of committees. We provide a comprehensive description of this functionality in Section 7.2.1. To achieve the functionality in a real-world setting, we design the protocol $\Pi_{Evolving}[\mathbb{G}]$, which ensures the security and correctness of the evolving committee mechanism. This protocol is detailed in Section 7.2.2.

To ensure the integrity and reliability of the evolving committee mechanism, we conduct a thorough security analysis of $\Pi_{Evolving}[\mathbb{G}]$ in Section 7.2.4. Furthermore, we explore the critical aspect of maintaining an honest majority in the voting committee for each round in Section 7.2.3. This aspect is crucial to prevent potential adversarial influence and maintain trustworthiness of the decision-making process.

By establishing the Evolving Functionality and the corresponding protocol, we lay the foundation for a dynamic and robust committee management system. These mechanisms enable the continuous evolution of the committee members, ensuring resistance against adversarial attacks and enhancing the security and privacy guarantees of the overall decision-making system.

7.2.1 Evolving Functionality $\mathcal{F}_{Evolving}[\mathbb{G}]$

Evolving Functionality is designed to support Evolving Committee so that voting committee members can be replaced during the voting process. As shown in Figure 7.1, Evolving Functionality, $\mathcal{F}_{Evolving}[\mathbb{G}]$, interacts with two continuous voting committees: previous committee, $\mathcal{C}_r^{[n_r]}$, and new committee, $\mathcal{C}_{r+1}^{[n_{r+1}]}$. $\mathcal{F}_{Evolving}[\mathbb{G}]$ maintains a set O (Initially set to $\emptyset$). Denote $\mathcal{C}_{\mathsf{c},r+1}$ as corrupted voting committee members in $\mathcal{C}_{r+1}^{[n_{r+1}]}$, and $\mathcal{C}_{\mathsf{h},r+1}$ as honest voting committee members in $\mathcal{C}_{r+1}^{[n_{r+1}]}$, n_{r+1} as the total number of voting committee members, so we have $|\mathcal{C}_{\mathsf{c},r+1}^{[n_{r+1}]}| + |\mathcal{C}_r^{[n_{r+1}]}| = n_{r+1}$. Set t_{r+1} as the corruption threshold, we have $|\mathcal{C}_{\mathsf{c},r+1}| \leq t_{r+1} - 1$.

In the first round, the first voting committee is online to generate global key pairs with $\Pi_{\text{DBKG}}^{n,t,m}$. Starting from the second round to the last round, there are two voting committees online: previous committee gets identities of the next committee and re-shares the global secret keys to the next committee. At the end of voting epoch, the voting committee performs the Tally tasks (Delegation Computation and Tally Computation) as defined in Voting Functionality $\mathcal{F}_{\text{VOTE}}^{c,\mu,s,n}$.

Upon receiving $(\text{HANDOVER}, sid, \{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}, \{\mathsf{psk}_{v,i}\}_{v=1}^{m})$ from previous committee, $\mathsf{C}_r^{(i)} \in \mathcal{C}_r^{[n_r]}$, $\mathcal{F}_{Evolving}[\mathbb{G}]$ first checks if it receives more than t_r values of same $\{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}$ to guarantee the number of honest members: $\mathcal{F}_{Evolving}[\mathbb{G}]$ sets $\mathsf{O} := \mathsf{O} \cup \{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}$, if there are more than t_r values of $\{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}$ in O are the same, assert all $\{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}$ to this value, and continue to next step. Then $\mathcal{F}_{Evolving}[\mathbb{G}]$

reconstructs the global secret keys $\{\mathsf{gsk}_v\}_{v=1}^m$ based on Lagrange Interpolation for $v \in [m]$:

$$\mathsf{gsk}_v := \prod_{j \in \mathsf{R}} \lambda_{v,j} \cdot \mathsf{psk}_{v,j}, \tag{7.1}$$

where R is the set of honest parties' indexes in $\mathcal{C}_r^{[n_r]}$, $|\mathsf{R}| = t_r$, $\{\lambda_{v,j}\}_{v\in[m],j\in\mathsf{R}}$ are Lagrange Interpolation coefficients. S is notified by $(\text{HANDOVERNOTIFY}, \mathsf{sid}, \mathsf{C}_r^{(i)}, \{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r})$.

Adversary can choose shares for the corrupted new committee members by sending $(\text{CORRUPTSHARES}, \mathsf{sid}, \{i, \{\mathsf{psk}_{v,i}\}_{v=1}^m\}_{\mathsf{C}_{r+1}^{(i)} \in \mathcal{C}_{\mathsf{c},r+1}^{[n_{r+1}]}})$ to $\mathcal{F}_{Evolving}[\mathbb{G}]$. For $v \in [m]$, $\mathcal{F}_{Evolving}[\mathbb{G}]$ constructs random polynomial based on all the share and gsk_v. $\mathcal{F}_{Evolving}[\mathbb{G}]$ sets $a := t_{r+1} - |\mathcal{C}_{\mathsf{c},r+1}| - 1$, $\mathcal{C}'_{\mathsf{h},r+1} \subset \mathcal{C}_{\mathsf{h},r+1}$, $|\mathcal{C}'_{\mathsf{h},r+1}| = a$, selects random $\{\mathsf{psk}_{v,i}\}_{\mathsf{C}_{r+1}^{(i)} \in \mathcal{C}'_{\mathsf{h},r+1}, v\in[m]}$. For $v \in [m]$, $\mathcal{F}_{Evolving}[\mathbb{G}]$ constructs random polynomial $F_v(z) := \sum_{t=0}^{t_{r+1}-1} a_{v,t} \cdot z^t$ under the restriction $F_v(j) = \mathsf{psk}_{j,v}$ for $\mathsf{C}_{r+1}^{(i)} \in \{\mathcal{C}'_{\mathsf{h},r+1} \cup \mathcal{C}_{\mathsf{c},r+1}\}$, and $F_v(0) = \mathsf{gsk}_v$.

Every new committee member can send $(\text{READNEWSHARE}, \mathsf{sid})$ to $\mathcal{F}_{Evolving}[\mathbb{G}]$ asking for their shares. $\mathcal{F}_{Evolving}[\mathbb{G}]$ computes partial secret key and partial global key for $\mathsf{C}_{r+1}^{(j)}$ based on the polynomials generated in last step

$$\begin{aligned} &\mathsf{psk}'_{v,j} := F_v(j) \text{ for } v \in [m], j \in [n_{r+1}], \\ &\mathsf{ppk}'_{v,j} := g^{s_{v,j}} \text{ for } v \in [m], j \in [n_{r+1}]. \end{aligned} \tag{7.2}$$

$\mathcal{F}_{Evolving}[\mathbb{G}]$ returns $(\text{READNEWSHARERETURN}, \mathsf{sid}, \{\mathsf{psk}'_{v,j}\}_{v=1}^m, \{\mathsf{ppk}'_{v,j}\}_{v=1,j=1}^{m,n_{r+1}})$ to new committee member. In addition, any party can request global public keys and partial public keys by $(\text{READPK}, \mathsf{sid})$, $\mathcal{F}_{Evolving}[\mathbb{G}]$ computes

$$\mathsf{gpk}_v := \prod_{j=1}^{t_{r+1}} (\mathsf{ppk}'_{v,j})^{\gamma_{v,j}} \text{for } v \in [m]. \tag{7.3}$$

Then $\mathcal{F}_{Evolving}[\mathbb{G}]$ sends $(\text{READPKRETURN}, \mathsf{sid}, \{\mathsf{gpk}_v\}_{v=1}^m, \{\mathsf{ppk}'_{v,i}\}_{v=1,i=1}^{m,n_{r+1}})$ to the requester.

$\mathcal{F}_{Evolving}[\mathbb{G}]$

$\mathcal{F}_{Evolving}[\mathbb{G}]$ interacts with $\mathcal{C}_r^{[n_r]}, \mathcal{C}_{r+1}^{[n_{r+1}]}$, and maintains a set O (Initially set to $\emptyset$). Denote $\mathcal{C}_{\mathsf{c},r+1}$ as corrupted voting committee members in $\mathcal{C}_{r+1}^{[n_{r+1}]}$, and $\mathcal{C}_{\mathsf{h},r+1}$ as honest voting committee members in $\mathcal{C}_{r+1}^{[n_{r+1}]}$, $|\mathcal{C}_{\mathsf{c},r+1}^{[n_{r+1}]}| + |\mathcal{C}_r^{[n_{r+1}]}| = n_{r+1}$, and $|\mathcal{C}_{\mathsf{c},r+1}| \leq t_{r+1} - 1$.

$\mathcal{F}_{Evolving}[\mathbb{G}]$ does the following:

- Upon receiving $(\text{HANDOVER}, sid, \{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}, \{\mathsf{psk}_{v,i}\}_{v=1}^{m})$ from $\mathsf{C}_r^{(i)} \in \mathcal{C}_r^{[n_r]}$:
  - Set $\mathsf{O} := \mathsf{O} \cup \{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}$;
  - If there are more than t_r values of $\{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}$ in O are the same, assert all $\{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}$ to this value, and continue to next step;
  - For $v \in [m]$, compute $\mathsf{gsk}_v := \prod_{j \in \mathsf{R}} \lambda_{v,j} \cdot \mathsf{psk}_{v,j}$, where R is the set of honest parties' indexes in $\mathcal{C}_r^{[n_r]}$, $|\mathsf{R}| = t_r$, $\{\lambda_{v,j}\}_{v \in [m], j \in \mathsf{R}}$ are Lagrange Interpolation coefficients;
  - Send $(\text{HANDOVERNOTIFY}, \mathsf{sid}, \mathsf{C}_r^{(i)}, \{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r})$ to S.
- Upon receiving $(\text{CORRUPTSHARES}, \mathsf{sid}, \{i, \{\mathsf{psk}_{v,i}\}_{v=1}^{m}\}_{\mathsf{C}_{r+1}^{(i)} \in \mathcal{C}_{\mathsf{c},r+1}^{[n_{r+1}]}})$ from S:
  - Set $a := t_{r+1} - |\mathcal{C}_{\mathsf{c},r+1}| - 1$, $\mathcal{C}'_{\mathsf{h},r+1} \subset \mathcal{C}_{\mathsf{h},r+1}$, $|\mathcal{C}'_{\mathsf{h},r+1}| = a$, select $\{\mathsf{psk}_{v,i}\}_{\mathsf{C}_{r+1}^{(i)} \in \mathcal{C}'_{\mathsf{h},r+1}, v \in [m]} \leftarrow (\mathbb{Z}_q)^{[m \cdot a]}$;
  - For $v \in [m]$, construct random polynomial $F_v(z) := \sum_{t=0}^{t_{r+1}-1} a_{v,t} \cdot z^t$ under the restriction $F_v(j) = \mathsf{psk}_{j,v}$ for $\mathsf{C}_{r+1}^{(i)} \in \{\mathcal{C}'_{\mathsf{h},r+1} \cup \mathcal{C}_{\mathsf{c},r+1}\}$, and $F_v(0) = \mathsf{gsk}_v$.
- Upon receiving $(\text{READNEWSHARE}, \mathsf{sid})$ from $\mathsf{C}_{r+1}^{(j)} \in \mathcal{C}_{\mathsf{c},r+1}^{[n_{r+1}]}$:
  - Compute $\mathsf{psk}'_{v,j} := F_v(j)$ for $v \in [m]$,$j \in [n_{r+1}]$;
  - Compute $\mathsf{ppk}'_{v,j} := g^{s_{v,j}}$ for $v \in [m]$,$j \in [n_{r+1}]$.
  - Send $(\text{READNEWSHARERETURN}, \mathsf{sid}, \{\mathsf{psk}'_{v,j}\}_{v=1}^{m}, \{\mathsf{ppk}'_{v,j}\}_{v=1,j=1}^{m,n_{r+1}})$ to $\mathsf{C}_{r+1}^{(j)}$;
- Upon receiving $(\text{READPK}, \mathsf{sid})$ from any party, compute $\mathsf{gpk}_v := \prod_{j=1}^{t_{r+1}} (\mathsf{ppk'}_{v,j})^{\gamma_{v,j}}$ for $v \in [m]$ and return $(\text{READPKRETURN}, \mathsf{sid}, \{\mathsf{gpk}_v\}_{v=1}^{m}, \{\mathsf{ppk}'_{v,i}\}_{v=1,i=1}^{m,n_{r+1}})$ to the requester.

Figure 7.1: Evolving functionality, $\mathcal{F}_{Evolving}[\mathbb{G}]$.

7.2.2 Evolving Protocol $\Pi_{Evolving}[\mathbb{G}]$

We give $\Pi_{Evolving}[\mathbb{G}]$ in $\{\mathcal{F}_{\text{BC}}\}$ -*hybrid world* to realise $\mathcal{F}_{Evolving}[\mathbb{G}]$ in Fig. 7.2. The prior committee will share their partial secret keys $\{\mathsf{psk}_{v,i}\}_{v=1}^{m}$ to new committee, and encrypt shares with recipients' public keys. New committee can verify which of the shares are valid

through verifiable secret sharing, and compute their own partial secret keys and partial public keys of new committee members.

Upon receiving $(\text{EVOLVING}, sid, \{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}, \{\mathsf{psk}_{v,i}\}_{v=1}^{m})$ from $\mathcal{Z}$, prior voting committee member, $\mathsf{C}_r^{(i)} \in \mathcal{C}_r^{[n_r]}$, re-shares its partial secret keys to the new committee members. $\mathsf{C}_r^{(i)}$ first selects random polynomial

$$F_{v,i}(z) := \sum_{t=0}^{t_{r+1}-1} a_{v,i,t} \cdot z^t, \tag{7.4}$$

where $a_{v,i,0} = \mathsf{psk}_{v,i}$, $\{a_{v,i,t}\}_{t=1}^{t_{r+1}-1} \leftarrow (\mathbb{Z}_q)^{[t_{r+1}-1]}$. Next $\mathsf{C}_r^{(i)}$ computes new shares for new committee members based on $F_{v,i}(z)$, and encrypt the shares with the public key of the related new committee member. The new shares are computed by $s_{v,i,j} := F_{v,i}(j)$ for $j \in [n_{r+1}]$, and encrypted by

$$(A_{v,i,j}, B_{v,i,j}) := \mathsf{Enc}_{\mathsf{pk}_j}(s_{v,i,j}; r_{v,i,j}) \text{ for } j \in [n_{r+1}], \tag{7.5}$$

where $\{r_{v,i,j}\}_{j=1}^{n_{r+1}}$ are randomly selected. Then $\mathsf{C}_r^{(i)}$ commits the coefficients of its polynomial by

$$H_{v,i,t} := \mathsf{Com}_{\mathsf{ck}}(a_{v,i,t}; 0) \text{ for } v \in [m],\ t \in [1, t_{r+1} - 1], \tag{7.6}$$

and submits $(\{H_{v,i,t}\}_{t=1}^{t_{r+1}-1}, \{A_{v,i,j}, B_{v,i,j}\}_{v=1,j=1}^{m,n_{r+1}}, \{\mathsf{ppk}_{v,j}\}_{j=1,v=1}^{n_r,m})$ to $\mathcal{F}_{\text{BC}}$.

Upon receiving $(\text{READNEWSHARE}, \mathsf{sid})$ from $\mathcal{Z}$, the new committee member, $\mathsf{C}_{r+1}^{(i)} \in \mathcal{C}_{r+1}^{[n_{r+1}]}$, decrypts and validates the shares from prior committee members. $\mathsf{C}_{r+1}^{(i)}$ fetches the messages $(\{\{H_{v,j,t}\}_{t=1}^{t_{r+1}-1}, A_{v,j,i}, B_{v,j,i}, \mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r})$ from prior committee members and asserts $\{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}$ following majority rule. Then $\mathsf{C}_{r+1}^{(i)}$ computes

$$s_{v,j,i} = \mathsf{Dec}_{\mathsf{sk}_j}(A_{v,j,i}, B_{v,j,i}) \text{ for } v \in [m],\ j \in [n_r]. \tag{7.7}$$

Next, $\mathsf{C}_{r+1}^{(i)}$ validates these shares based on the partial public keys on blockchain, and the polynomial commitments sent by prior committee members. For $j \in [n_r]$, for $v \in [m]$, $\mathsf{C}_{r+1}^{(i)}$ sets $H_{v,j,0} = \mathsf{ppk}_{v,j}$, checks if $g^{s_{v,j,i}} = \prod_{t=0}^{t_{r+1}-1}(H_{v,j,t})^{i^t}$. If this verification fails for $\mathsf{C}_r^{(j)}$, $\mathsf{C}_{r+1}^{(i)}$ posts $(\text{COMPLAINT}, s_{v,j,i}, \sigma_i)$ to $\mathcal{F}_{\text{BC}}$, where σ_i is Correct Decryption NIZK:

$$\sigma_i \leftarrow \mathsf{NIZK}\left\{ \begin{array}{l} (g, \mathsf{pk}_i, \{A_{v,j,i}, B_{v,j,i}\}_{j=1}^{n_r}), (\mathsf{sk}_i) : \\ s_{v,j,i} = \mathsf{Dec}_{\mathsf{sk}_i}(A_{v,j,i}, B_{v,j,i}) \ \wedge \mathsf{pk}_i = g^{\mathsf{sk}_i} \end{array} \right\} \tag{7.8}$$

If there is a valid complain on $\mathcal{F}_{\text{BC}}$ about $\mathsf{C}_r^{(j)} \in \mathcal{C}_r^{[n_r]}$, $\mathsf{C}_{r+1}^{(i)}$ sets $\mathbf{V} := [n_r] \setminus \{j\}$, and selects any t_{r+1} values from $\mathbf{V}$ as $\mathbf{V}'$. For $v \in [m]$, $\mathsf{C}_{r+1}^{(i)}$ computes

$$\mathsf{psk}'_{v,i} := \sum_{j \in \mathbf{V}'} s_{v,j,i} \cdot \gamma_{v,j} \tag{7.9}$$

by interpolation. Afterwards, $\mathsf{C}_{r+1}^{(i)}$ computes $\mathsf{ppk}'_{v,k} := \prod_{j \in \mathbf{V}'} (\prod_{t=0}^{t_{r+1}-1} (H_{v,j,t})^{k^t})^{\gamma_{v,j}}$ for $v \in [m]$, $k \in [n_{r+1}]$, and posts $(\{\{\mathsf{ppk}'_{v,k}\}_{v=1,k=1}^{m,n_{r+1}}\})$ to $\mathcal{F}_{\text{BC}}$.

Upon receiving $(\text{READPK}, \mathsf{sid})$ from any party P, P fetches $\{\mathsf{ppk}'_{v,j}\}_{v=1,j=1}^{m,n_{r+1}}$, computes $\mathsf{gpk}_v := \prod_{j=1}^{t_{r+1}} (\mathsf{ppk}'_{v,j})^{\gamma_{v,j}}$, and returns $(\text{READPKRETURN}, \mathsf{sid}, \{\mathsf{gpk}_v\}_{v \in [m]}, \{\mathsf{ppk}'_{v,j}\}_{v=1,j=1}^{m,n_{r+1}})$ to $\mathcal{Z}$.

Evolving Protocol $\Pi_{Evolving}[\mathbb{G}]$

Assume that every participant has its own key pair $(\mathsf{pk}_i, \mathsf{sk}_i)$.

- Upon receiving $(\text{EVOLVING}, sid, \{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}, \{\mathsf{psk}_{v,i}\}_{v=1}^{m})$ from $\mathcal{Z}$, $\mathsf{C}_r^{(i)} \in \mathcal{C}_r^{[n_r]}$ does the following:
  - Select random polynomial $F_{v,i}(z) := \sum_{t=0}^{t_{r+1}-1} a_{v,i,t} \cdot z^t$, where $a_{v,i,0} = \mathsf{psk}_{v,i}$, $\{a_{v,i,t}\}_{t=1}^{t_{r+1}-1} \leftarrow (\mathbb{Z}_q)^{[t_{r+1}-1]}$;
  - Compute $s_{v,i,j} := F_{v,i}(j)$ for $j \in [n_{r+1}]$;
  - Choose $\{r_{v,i,j}\}_{j=1}^{n_{r+1}} \leftarrow (\mathbb{Z}_q)^{[n_{r+1}]}$, compute $(A_{v,i,j}, B_{v,i,j}) := \mathsf{Enc}_{\mathsf{pk}_j}(s_{v,i,j}; r_{v,i,j})$ for $j \in [n_{r+1}]$;
  - Compute $H_{v,i,t} := \mathsf{Com}_{\mathsf{ck}}(a_{v,i,t}; 0)$ for $v \in [m]$, $t \in [1, t_{r+1} - 1]$;
  - Send $(\mathsf{Write}, \mathsf{sid}, (\{H_{v,i,t}\}_{t=1}^{t_{r+1}-1}, \{A_{v,i,j}, B_{v,i,j}\}_{v=1,j=1}^{m,n_{r+1}}, \{\mathsf{ppk}_{v,j}\}_{j=1,v=1}^{n_r,m}))$ to $\mathcal{F}_{\text{BC}}$.
- Upon receiving $(\text{READNEWSHARE}, \mathsf{sid})$ from $\mathcal{Z}$, $\mathsf{C}_{r+1}^{(i)} \in \mathcal{C}_{r+1}^{[n_{r+1}]}$ does the following:
  - Send $(\mathsf{Read}, \mathsf{sid})$ to $\mathcal{F}_{\text{BC}}$, get $(\{\{H_{v,j,t}\}_{t=1}^{t_{r+1}-1}, A_{v,j,i}, B_{v,j,i}, \mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r})$;
  - Assert $\{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}$ following majority rule;
  - Compute $s_{v,j,i} = \mathsf{Dec}_{\mathsf{sk}_j}(A_{v,j,i}, B_{v,j,i})$ for $v \in [m]$, $j \in [n_r]$;
  - For $j \in [n_r]$, for $v \in [m]$, set $H_{v,j,0} = \mathsf{ppk}_{v,j}$, check if $g^{s_{v,j,i}} = \prod_{t=0}^{t_{r+1}-1} (H_{v,j,t})^{i^t}$. If this verification fails for $\mathsf{C}_r^{(j)}$, send $(\mathsf{Write}, \mathsf{sid}, (\text{COMPLAINT}, s_{v,j,i}, \sigma_i))$ to $\mathcal{F}_{\text{BC}}$, where σ_i is Correct Decryption NIZK:
$$\sigma_i \leftarrow \mathsf{NIZK}\left\{ \begin{array}{l} (g, \mathsf{pk}_i, \{A_{v,j,i}, B_{v,j,i}\}_{j=1}^{n_r}), (\mathsf{sk}_i) : \\ s_{v,j,i} = \mathsf{Dec}_{\mathsf{sk}_i}(A_{v,j,i}, B_{v,j,i}) \ \wedge \mathsf{pk}_i = g^{\mathsf{sk}_i} \end{array} \right\}$$
  - If there is a valid complain to $\mathcal{F}_{\text{BC}}$ about $\mathsf{C}_r^{(j)} \in \mathcal{C}_r^{[n_r]}$, set $\mathbf{V} := [n_r] \setminus \{j\}$, select any t_{r+1} values from $\mathbf{V}$ as $\mathbf{V}'$;
  - For $v \in [m]$, compute $\mathsf{psk}'_{v,i} := \sum_{j \in \mathbf{V}'} s_{v,j,i} \cdot \gamma_{v,j}$ by interpolation;
  - Compute $\mathsf{ppk}'_{v,k} := \prod_{j \in \mathbf{V}'} (\prod_{t=0}^{t_{r+1}-1} (H_{v,j,t})^{k^t})^{\gamma_{v,j}}$ for $v \in [m]$, $k \in [n_{r+1}]$, post $(\{\{\mathsf{ppk}'_{v,k}\}_{v=1,k=1}^{m,n_{r+1}}\})$ to $\mathcal{F}_{\text{BC}}$.
- Upon receiving $(\text{READPK}, \mathsf{sid})$ from any party P, P sends $(\mathsf{Read}, \mathsf{sid})$ to $\mathcal{F}_{\text{BC}}$, and gets $\{\mathsf{ppk}'_{v,j}\}_{v=1,j=1}^{m,n_{r+1}}$, computes $\mathsf{gpk}_v := \prod_{j=1}^{t_{r+1}} (\mathsf{ppk}'_{v,j})^{\gamma_{v,j}}$, and returns $(\text{READPKRETURN}, \mathsf{sid}, \{\mathsf{gpk}_v\}_{v \in [m]}, \{\mathsf{ppk}'_{v,j}\}_{v=1,j=1}^{m,n_{r+1}})$ to $\mathcal{Z}$.

Figure 7.2: Evolving Protocol, $\Pi_{Evolving}[\mathbb{G}]$ in $\{\mathcal{F}_{\text{BC}}\}$ -*hybrid world*

7.2.3 Honest Majority

As described in Chapter 4, in our design, stake holders who want to join voting committee need to lock some stakes on blockchain, the probability of being selected is proportional to the amount of locked stakes. Denote the number of total deposited coins on blockchain by N, which are owned by S stakeholders denoted by $\mathcal{S}$. Among these S stakeholders, let S_m and S_h be the number of malicious and honest stakeholders respectively. Denote the total coins owned by S_m malicious stakeholders by C_m, and total coins owned by S_h malicious stakeholders by C_h. Let T_m be the ratio of malicious coins in total deposited coins, and T_h be the ratio of honest coins in total deposited coins, we have the following:

$$\begin{aligned} N &= N_m + N_h, \\ S &= S_m + S_h, \\ T_m &:= N_m/N, \\ T_h &:= N_h/N. \end{aligned} \tag{7.10}$$

To investigate the honesty of all the stakeholders, we randomly select a sample of coins from the total coins. Denote this random sample set by $\mathcal{S}'$, of which the size is n. Let t_m be the ratio of malicious coins in $\mathcal{S}'$, and t_h be the ratio of honest coins in $\mathcal{S}'$. We have the following:

$$\begin{aligned} t_h + t_m &= 1, \\ n \cdot t_h + n \cdot t_m &= n. \end{aligned} \tag{7.11}$$

By setting $\lambda := t_m \cdot n$ as the number of coins owned by malicious stakeholders in the sample $\mathcal{S}'$. Our target is to estimate the probability that the value λ/n is essentially larger than T_m defined in Equation 7.10, with Theorem 16 and Proposition 1.

Theorem 16. *Let N be the number of total coins, T_m be the coins owned by malicious stakeholders, n be the number of randomly sampled coins set, t_m be the ratio of coins owned by malicious stakeholders in the sampling set. The coins, λ, owns by malicious stakeholders in the randomly sampled coins set, follows binomial distribution with n and T_m.*

Proof of Theorem 16. Before giving the detailed proof, we first assume that there exists a procedure to randomly select coins from $\mathcal{S}$ one by one. Initially, we set $\mathcal{S}'$ empty. A random coin is selected from $\mathcal{S}$ and sent to $\mathcal{S}'$ until $|\mathcal{S}'| = n$.

For $k \in [n]$, we define the following random variable:

$$\lambda_k = \begin{cases} 1, & \text{if the } k\text{-th coin belongs to a malicious participant,} \\ 0, & \text{if the } k\text{-th coin belongs to a honest participant.} \end{cases} \tag{7.12}$$

Next, we prove the Bernoulli distribution of $\{\lambda_k\}_{k=1}^{n}$ by induction.

When $k = 1$, we have the following:

$$\Pr(\lambda_1) = N_m/N = T_m. \tag{7.13}$$

When $k = i$, we can infer the following based on law of total probability:

$$\begin{aligned}
\Pr(\lambda_i) &= \sum_{l=0}^{i-1} \Pr(\lambda_i = 1 | \sum_{a=1}^{i-1} \lambda_a = l) \cdot \Pr(\sum_{a=1}^{i-1} \lambda_a = l) \\
&= \sum_{l=0}^{i-1} (N_m - l)/(n - (i-l)) \cdot \begin{pmatrix} l \\ i-1 \end{pmatrix} \cdot (T_m)^l \cdot (T_h)^{i-1-l} \\
&= 1/(n-(i-1))[N_m \cdot \sum_{l=0}^{i-1} \begin{pmatrix} l \\ i-1 \end{pmatrix} \cdot (T_m)^l \cdot (T_h)^{i-1-l} \\
&- \sum_{l=0}^{i-1} l \cdot \begin{pmatrix} l \\ i-1 \end{pmatrix} \cdot (T_m)^l \cdot (T_h)^{i-1-l}] \\
&= 1/(n-(i-1))[N_m - (i-1) \cdot T_m] = N_m/(n-(i-1)) \cdot (1 - (i-1)/n) \\
&= N_m/n = T_m.
\end{aligned} \tag{7.14}$$

Therefore, $\{\lambda_k\}_{k=1}^n$ have Bernoulli distribution with T_m.

□

Proposition 1. *Let N be sufficient large*[1]*. For any integer l, such that $1 \leq l \leq n$, the next approximations hold:*

- $P(\lambda = l) \approx \begin{pmatrix} n \\ l \end{pmatrix} \cdot (T_m)^l \cdot (T_h)^{n-l}$
- $P(\lambda \geq l) \approx \sum_{i=1}^{n} \begin{pmatrix} n \\ i \end{pmatrix} \cdot (T_m)^i \cdot (T_h)^{n-i}$.

Proof of Proposition 1.
Its easy to see that

$$P(\lambda = l) = \frac{\begin{pmatrix} S_m \\ l \end{pmatrix} \begin{pmatrix} S_h \\ n-l \end{pmatrix}}{\begin{pmatrix} N \\ n \end{pmatrix}} \tag{7.15}$$

[1] In the real world environment with our evaluation, to select one hundred coins, the expected locked stake amount is tens or hundreds of millions of tokens.

Next, we assume that malicious stakeholders are form some significant minority of all stakeholders, we can approximate binomial coefficients as

$$\begin{aligned} \left(\begin{array}{c} S_m \\ l \end{array} \right) &\approx \frac{(S_m)^l}{l!} \\ \left(\begin{array}{c} S_h \\ n-l \end{array} \right) &\approx \frac{(S_h)^{n-l}}{(n-l)!} \\ \left(\begin{array}{c} N \\ n \end{array} \right) &\approx \frac{N^n}{n!} \end{aligned} \tag{7.16}$$

Then, substitution Equation 7.16 into Equation 7.15 gives us approximation

$$P(\lambda = l) \approx \frac{\frac{(S_m)^l}{l!}\frac{(S_h)^{n-l}}{(n-l)!}}{\frac{N^n}{n!}} \tag{7.17}$$

and the first approximation is proved.

Next, we use the fact that the probability of union of disjoint events is equal to the sum of corresponding probabilities, and we can get that

$$P(\lambda \geq l) = P\left(\cup_l^i (\lambda = i)\right) = \sum_{i=l}^{n} P(\lambda = i) = \sum_{i=l}^{n} \left(\begin{array}{c} n \\ i \end{array} \right) \cdot (T_m)^i \cdot (T_h)^{n-i} \tag{7.18}$$

Therefore the Proposition 1 is proved. □

We further define n as the size of voting committee, and l as the number of committee members controlled by the adversary. Given a small value, ϵ, for any ε, based on Proposition 1 and Theorem 16, we can refer that

$$\Pr(\varepsilon \geq \epsilon \cdot n) = \sum_{l=\lfloor \epsilon \cdot n \rfloor}^{n} \left(\begin{array}{c} n \\ l \end{array} \right) (T_m)^l (T_h)^{n-l} \tag{7.19}$$

Given the assumption that adversary can corrupt at most $(n-1)/2$ committees, we give numerical examples (from Table A.1 to Table A.25). For example, in Table A.2, when there are 20 voting committee members, if the adversary has 30% of the locked stakes, then it will get at least 7 committee members $(7, 8, \ldots, 20)$ with probability 0.39199. In Table A.13, when $n = 10$, if there are 55% of the stakes are honest, the probability that at least 50% of the voting committee members are honest is 0.738437.

We recommend the following values minimal acceptable for the protocol parameters:

- committee size of at least 70 members;
- using threshold signature that requires participation of at least 60% of the committee members.

Within the assumptions and recommended parameters, the probability of adversary preventing honest participants to put the corresponding transaction on blockchain is less than 0.0001 for a single decision-making epoch. The probability of adversarial control over the decision-making fund (within the same assumptions and parameters) is negligible small. In terms of security, with overwhelming probability, the majority of the voting committee members in every round are honest, which can guarantee the privacy of ballots and protocol termination. If a cheating voting committee member is detected, it will lose all the deposit, and get banned by decision-making system forever.

7.2.4 Security Analysis of Evolving Committee Mechanism

We give Theorem 17 to prove that $\Pi_{Evolving}[\mathbb{G}]$ *UC-realises* $\mathcal{F}_{Evolving}[\mathbb{G}]$ in $\{\mathcal{F}_{\text{BC}}\}$ *-hybrid world*, based on the indistinguishability between the ideal execution $\mathsf{EXEC}_{\mathcal{F}_{Evolving}[\mathbb{G}],\mathsf{S},\mathcal{Z}}$ and the real execution $\mathsf{EXEC}^{\mathcal{F}_{\text{BC}}}_{\Pi_{Evolving}[\mathbb{G}],\mathcal{A},\mathcal{Z}}$.

Theorem 17 (Evolving)**.** *Assume Lifted Elgamal encryption* Enc *is IND-CPA secure with adversary advantage of* $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A})$*. Assume Correct Decryption* NIZK *is perfect complete, perfect special honest verifier zero knowledge, and computational sound with adversary advantage of* $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Dec}(1^\kappa, \mathcal{A})$*. Assume* Enc *is IND-CPA secure with adversary advantage of* $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A})$*. The protocol* $\Pi_{Evolving}[\mathbb{G}]$ *UC-realise* $\mathcal{F}_{Evolving}[\mathbb{G}]$ *in* $\{\mathcal{F}_{\text{BC}}\}$ *-hybrid world against static corruption up to* $t-1$ *parties with distinguishing advantage upper bounded by*

$$(m \cdot n_{r+1}) \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A}) + (t-1) \cdot \mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Dec}(1^\kappa, \mathcal{A})$$

Proof of Theorem 17.
To prove theorem 17, we first construct a simulator S such that no nonuniform PPT environment $\mathcal{Z}$ can distinguish between ideal world and real world: in ideal world, the ideal execution $\mathsf{EXEC}_{\mathcal{F}_{Evolving}[\mathbb{G}],\mathsf{S},\mathcal{Z}}$ where the parties interact with functionality $\mathcal{F}_{Evolving}[\mathbb{G}]$ in the ideal world and corrupted parties are controlled by the simulator S; in the real world, the real execution $\mathsf{EXEC}^{\mathcal{F}_{\text{BC}}}_{\Pi_{Evolving}[\mathbb{G}],\mathcal{A},\mathcal{Z}}$ where the parties run protocol $\Pi_{Evolving}[\mathbb{G}]$ in the $\{\mathcal{F}_{\text{BC}}\}$-$hybrid\ world$ and the corrupted parties are controlled by a dummy adversary $\mathcal{A}$ who simply forwards messages from/to $\mathcal{Z}$.

Simulator. The simulator S internally runs $\mathcal{A}$, forwarding messages to/from the environment $\mathcal{Z}$. The simulator S simulates the following interactions with $\mathcal{A}$:

- Upon receiving $(\textsc{HandoverNotify}, \mathsf{sid}, \mathsf{C}^{(i)}_r, \{\mathsf{ppk}_{v,j}\}^{m,n_r}_{v=1,j=1})$ from the ideal functionality $\mathcal{F}_{Evolving}[\mathbb{G}]$ about an honest voting committee member $\mathsf{C}^{(i)}_r \in \mathcal{C}_{\mathsf{h},r}$, the simulator S does the following:
  - For $v \in [m]$:

  - $*$ Set $a := t_{r+1} - |\mathcal{C}_{\mathsf{c},r+1}| - 1$, $\mathcal{C}'_{\mathsf{h},r+1} \subset \mathcal{C}_{\mathsf{h},r+1}$, $|\mathcal{C}'_{\mathsf{h},r+1}| = a$;
  - $*$ Select $\{s_{v,i,j}\}_{j \in \mathcal{C}'_{\mathsf{h},r+1}} \leftarrow (\mathbb{Z}_q)^{[a]}$;
  - $*$ Compute $A_{v,i,j} := g^{s_{v,i,j}}$ for $\mathsf{C}^{(i)}_{r+1} \in \mathcal{C}'_{\mathsf{h},r+1}$;
  - $*$ Construct a degree t_{r+1} polynomial $G(\cdot)$ over $\mathbb{G}$ by Lagrange Interpolation, where $\{A_{v,i,j}\}_{j \in \mathcal{C}'_{\mathsf{h},r+1}}$ are $t_{r+1} - 1$ outputs, $\mathsf{ppk}_{v,i}$ is the free term. Denote other coefficients of $G(\cdot)$ by $\{H_{v,i,t}\}_{t=1}^{t_{r+1}-1}$.
  - $*$ Select $\{s_{v,i,j}\}_{j \in \mathcal{C}'_{\mathsf{h},r+1}} \leftarrow (\mathbb{Z}_q)^{[a]}$;
  - $*$ Select $\{r_{v,i,j}\}_{j=1}^{n_{r+1}} \leftarrow (\mathbb{Z}_q)^{[n_{r+1}]}$, compute $(A_{v,i,j}, B_{v,i,j}) := \mathsf{Enc}_{\mathsf{pk}_j}(s_{v,i,j}; r_{v,i,j})$ for $j \in [n_{r+1}]$;
  - Post $(\{H_{v,i,t}\}_{t=1}^{t_{r+1}-1}, \{A_{v,i,j}, B_{v,i,j}\}_{v=1,j=1}^{m,n_{r+1}}, \{\mathsf{ppk}_{v,j}\}_{j=1,v=1}^{n_r,m})$ to $\mathcal{F}_{\text{BC}}$.

- Once the simulated $\mathcal{F}_{\text{BC}}$ receives $(\{H_{v,i,t}\}_{t=1}^{t_{r+1}-1}, \{A_{v,i,j}, B_{v,i,j}\}_{v=1,j=1}^{m,n_{r+1}}, \{\mathsf{ppk}_{v,j}\}_{j=1,v=1}^{n_r,m})$ from a corrupted voting committee member $\mathsf{C}^{(i)}_r \in \mathcal{C}_{\mathsf{c},r}$, the simulator S does the following:
  - Decrypt ciphertexts for honest voting committee members $\mathcal{C}_{\mathsf{h},r+1}$ and get $\{s_{v,i,j}\}_{v \in [m], j \in \mathcal{C}_{\mathsf{h},r+1}}$;
  - Interpolate to get $F_{v,i}$;
  - Compute $\{\mathsf{psk}_{v,i}\}_{v=1}^{m}$;
  - Send $(\text{EVOLVING}, sid, \{\mathsf{ppk}_{v,j}\}_{v=1,j=1}^{m,n_r}, \{\mathsf{psk}_{v,i}\}_{v=1}^{m})$ to $\mathcal{F}_{Evolving}[\mathbb{G}]$.

- Once the simulated $\mathcal{F}_{\text{BC}}$ receives $(\{\{H_{v,i,t}\}_{t=1}^{t_{r+1}-1}, \{A_{v,i,j}, B_{v,i,j}\}_{v=1,j=1}^{m,n_{r+1}}, \{\mathsf{ppk}_{v,j}\}_{j=1,v=1}^{n_r,m}\}_{\mathsf{C}^{(i)}_r \in \mathcal{C}_{\mathsf{c},r}})$ from all corrupted voting committee members, the simulator S does the following:
  - For $\mathsf{C}^{(i)}_{r+1} \in \mathcal{C}_{\mathsf{c},r+1}$:
    - $*$ Compute $s_{v,j,i} = F_{v,j}(i)$ for $\mathsf{C}^{(j)}_r \in \mathcal{C}_{\mathsf{c},r}$;
    - $*$ For $j \in [n_r]$, for $v \in [m]$, set $H_{v,j,0} = \mathsf{ppk}_{v,j}$, construct qualified set $\vec{V} \subseteq [n_r]$ where $g^{s_{v,j,i}} = \prod_{t=0}^{t_{r+1}-1} (H_{v,j,t})^{i^t}$ holds for $\mathsf{C}^{(j)}_r$, select any t_{r+1} values from $\vec{V}$ as $\vec{V}'$;
    - $*$ Compute $\mathsf{psk}_{v,i} := \sum_{\mathsf{C}^{(j)}_r \in \vec{V}'} s_{v,j,i} \cdot \gamma_{v,j}$;
  - Send $(\text{CORRUPTSHARES}, \mathsf{sid}, \{i, \{\mathsf{psk}_{v,i}\}_{v=1}^{m}\}_{\mathsf{C}^{(i)}_{r+1} \in \mathcal{C}^{[n_{r+1}]}_{\mathsf{c},r+1}})$ to $\mathcal{F}_{Evolving}[\mathbb{G}]$;

- Once $\mathcal{F}_{\text{BC}}$ gets complaint $(\text{COMPLAINT}, s_{v,j,i}, \sigma'_i)$ from $\mathsf{C}^{(i)}_{r+1}$ against $\mathsf{C}^{(j)}_r$, S will interpolate $\mathsf{C}^{(j)}_r$'s polynomial based on t_{r+1} shares sent to honest voting committee members from $\mathsf{C}^{(j)}_r$. If these shares are not in the same polynomial, S aborts;

- Upon receiving $(\text{READNEWSHARE}, \mathsf{sid})$ from the ideal functionality $\mathcal{F}_{Evolving}[\mathbb{G}]$ about an honest voting committee member $\mathsf{C}_{r+1}^{(j)} \in \mathcal{C}_{\mathsf{h},r+1}$, S follows the protocol for $\mathsf{C}_{r+1}^{(j)}$ and post $(\{\{\mathsf{ppk}'_{v,k}\}_{v=1,k=1}^{m,n_{r+1}}\})$ to $\mathcal{F}_{\text{BC}}$;
- Once the simulated $\mathcal{F}_{\text{BC}}$ receives $(\{\{\mathsf{ppk}'_{v,k}\}_{v=1,k=1}^{m,n_{r+1}}\})$ from corrupted voting committee member $\mathsf{C}_{r+1}^{(j)} \in \mathcal{C}_{\mathsf{c},r+1}$, S sends $(\text{READNEWSHARE}, \mathsf{sid})$ to $\mathcal{F}_{Evolving}[\mathbb{G}]$ on behalf of $\mathsf{C}_{r+1}^{(j)}$.

Indistinguishability.

The indistinguishability is proven through a series of hybrid worlds $\mathcal{H}_0, \ldots, \mathcal{H}_3$.

Hybrid $\mathcal{H}_0$: It is the real protocol execution $\mathsf{EXEC}^{\mathcal{F}_{\text{BC}}}_{\Pi_{Evolving}[\mathbb{G}],\mathcal{A},\mathcal{Z}}$.

Hybrid $\mathcal{H}_1$: $\mathcal{H}_1$ is the same as $\mathcal{H}_0$ except that in $\mathcal{H}_1$, $\{A_{v,i,j}, B_{v,i,j}\}_{v=1,j=1}^{m,n_{r+1}}$ sent by a honest voting committee member $\mathsf{C}_r^{(i)}$ is replaced by ciphertexts which encrypted random values.
Claim: If the lifted ElGamal encryption scheme is IND-CPA secure with adversarial advantage $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A})$, then $\mathcal{H}_1$ and $\mathcal{H}_0$ are indistinguishable with distinguishing advantage at most $(m \cdot n_{r+1}) \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A})$.
Proof: We have changed $m \cdot n_{r+1}$ ciphertexts which encrypted random strings, therefore, if any adversary $\mathcal{A}$ can distinguish $\mathcal{H}_1$ from $\mathcal{H}_0$, then we can construct an adversary $\mathcal{B}$, who can break IND-CPA game of Lifted Elgamal encryption scheme. The overall adversary advantage in $\mathcal{H}_1$ is $(m \cdot n_{r+1}) \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A})$. ■

Hybrid $\mathcal{H}_2$: $\mathcal{H}_2$ is the same as $\mathcal{H}_1$ except that in $\mathcal{H}_2$, the message $(\{H_{v,i,t}\}_{t=1}^{t_{r+1}-1})$ sent by a honest voting committee member $\mathsf{C}_r^{(i)}$ is replaced with the values computed backwards from $\{A_{v,i,j}\}_{j\in\vec{S}}$ and $\mathsf{ppk}_{v,i}$.
Claim: $\mathcal{H}_2$ and $\mathcal{H}_1$ are perfectly indistinguishable.
Proof: $(\{H_{v,i,t}\}_{t=1}^{t_{r+1}-1})$ in $\mathcal{H}_2$ and $\mathcal{H}_1$ follow the same distribution. ■

Hybrid $\mathcal{H}_3$: $\mathcal{H}_3$ is the same as $\mathcal{H}_2$ except that in $\mathcal{H}_3$, S aborts if it finds shares sent to honest member in $\mathcal{C}_{\mathsf{c},r}$ are not in the same polynomial about a $(\text{COMPLAINT}, s_{v,j,i}, \sigma'_i)$.
Claim: If Correct Decryption NIZK is computational sound with adversary advantage of $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Dec}(1^\lambda, \mathcal{A})$, $\mathcal{H}_3$ and $\mathcal{H}_2$ are indistinguishable.
Proof: If shares sent by corrupted $\mathsf{C}_r^{(j)}$ can pass NIZK but fail to lie on the same polynomial, it means adversary compromise the soundness property of NIZK, it can open NIZK to different witness. In this case, S will abort. Therefore, $\mathcal{H}_3$ and $\mathcal{H}_2$ are indistinguishable with adversary advantage of $\mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Dec}(1^\lambda, \mathcal{A})$. ■

The adversary's view of $\mathcal{H}_3$ is identical to the simulated view $\mathsf{EXEC}^{\mathcal{F}_{\text{BC}}}_{\Pi_{Evolving}[\mathbb{G}],\mathcal{A},\mathcal{Z}}$. Therefore, no $\mathsf{PPT}\,\mathcal{Z}$ can distinguish the view of the ideal execution from the view of the

real execution with more than advantage with distinguishing advantage upper bounded by

$$(m \cdot n_{r+1}) \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{Enc}}(1^\kappa, \mathcal{A}) + (t-1) \cdot \mathsf{Adv}^{\mathsf{Sound}}_{\mathsf{NIZK},Dec}(1^\kappa, \mathcal{A})$$

This concluded our proof of Theorem 17. □

7.3 Dynamic Scalable Key Management Scheme

One of the main impediments to the versatile development of blockchain-based systems is that the participating nodes require *high computational resources*. In some cases, the required computing is so complex that may take several hours or days to be completed. This not only limits the scalability of blockchain-based systems, but also raises major concerns on the sustainability of this technology ([251]). Especially, these issues are essential for Internet of Things (IoT) applications, where billions of nodes are involved and they often operate on limited computational and battery power and might frequently become offline to save energy ([252]) and extend their longevity.

When dealing with a large number of participants/nodes, the current DKG techniques are not effectively scalable. For example, [153] proposed a scaling method based on Authenticated Multipoint evaluation Tree (AMT). They also showed that for about two millions DKG participants, their proposed AMT DKG takes up to 2.2 hours and 2.7 days in the best and worst cases. This outperforms the scaling achieved by eJF-DKG ([148]) which takes about 578 days for the same number of participants. The overall efficiency of AMT DKG however is strongly dependent on the number of participants (N), as its communication and computational performance is significantly reduced by increasing N.

Particularly, upon joining new participants during DKG, the whole process of current DKG protocols needs to be restarted to adapt the new participant set. The reason behind is that these protocols only consider static participation, in which participants won't change during the execution. However, this requirement is hard to capture in practice, considering that some computation nodes might be offline or otherwise unavailable. Furthermore, due to that dynamic participation is allowed and accustomed in the nature of public blockchains, current static-participation-based model fails to meet the demand. Dynamic participation should be introduced in current privacy preserving solutions consequently to assist complex applications, where computation nodes are allowed to join in anytime and to participant in only certain phase instead of whole execution.

To address the above challenges, building upon the blockchain systems, we designed a Dynamic Scalable Key Management Scheme which has the following characteristics:

- **Fully Dynamic Participation**, so that the participants involved in distributed key generation should not be forced to remain online during the whole execution, and they are allowed to join anytime they want. Furthermore, upon joining new participants, DKG continues smoothly and no extra backward computation is required to accommodate the new participant.

- **Scalability**, so that the execution time, communication overhead, and computational complexity of DKG are not affected by increasing the number of DKG participants.

To achieve the first goal, we adopt the assembly line concept in the manufacturing processes. Intuitively, the whole DKG process can be divided into multiple interactive

epochs which are similar to workstations in the context of manufacturing process. Using this approach, in each epoch, a subset of participants are online and perform the corresponding partial DKG task. Then they transfer their results to the next subset of participants. Upon joining new participants, the process continues without the need for restarting.

To achieve the second goal, the main challenge is to avoid trading the scalability with the security. Achieving scalability, instead of engaging all participants, one may select a smaller group of them to commit. The smaller the group the higher the risk of full corruption as a powerful adversary can compromise all the participants in this small group. An alternative way is to introduce *anonymity* of participants. If the participants in each epoch remain anonymous, the adversary becomes unable to identify them before sending out messages. In the proposed method this is guaranteed by using an example of anonymous encryption, Elgamal encryption ([36]).

Using this approach, we can achieve maximum scalability with fully dynamic participation by guaranteeing the following:

- The overall corruption threshold in this setting can scale with the number of participants. This is because the evolved participants remain anonymous;
- The overall efficiency becomes independent of the number of participants as we only need smaller subset of participants to be active in the DKG process.

Based on the above, we proposed the Dynamic Scalable Key Management (*DSKM*) scheme, which enables large scale computation on a permissionless blockchain while achieving maximum scalability and fully dynamic participation.

At the heart of DSKM is a Committee-Based Assembly line (CBA line), which is inspired by Algorand's *participant-replaceability* in [253]. In our proposal, the CBA line relies on *Maintenance Committees* which include small subset of participants to perform the *Key Generation* and *Key Maintenance* tasks periodically. In the first epoch of the CBA line, the members of the first Maintenance Committee jointly generate global public and secret key. In the next epochs of the CBA line, the global secret key is maintained by different Maintenance Committees in a decentralised and privacy-preserving fashion.

The Maintenance Committee in each epoch collects the shares from the prior Maintenance Committee and compute their own partial secret keys. Except for the last epoch, similar to the relay race, the Maintenance committee then transfers the partial secret keys to the next Maintenance Committee based on the Verifiable Secret Sharing (VSS) [254].

Note that if an adversary corrupts more of the Maintenance Committee members than the VSS threshold, the secret key will be leaked to the adversary. To address this issue, we need to ensure that Maintenance Committee is secure enough (for example, honest majority) and anonymous, thus unidentifiable to the adversary. To ensure security of the Maintenance Committee against the adversary, we further introduce a *Selection Committee* with the task of endorsing and anonymising the members of Maintenance Committee. We propose the

Designation Procedure, in which a new *Selection Committee* is created to endorse and anonymise the members of Maintenance Committee.

In the *Designation Procedure*, every member of Selection Committee in each epoch selects one participant to join Maintenance Committee. To ensure the security of Maintenance Committee, each of them are assigned with a temporary alias (e.g., ephemeral key pair) by their nominator. They also remain anonymous until they send out messages including the shares of their partial secret key on the blockchain. We assume that a dishonest (honest) member of Selection Committee only chooses dishonest (honest) participants for the Maintenance Committee. Therefore, given the anonymity of Maintenance Committee, the honesty of Maintenance Committee and the security of global secret key are directly determined by the honesty of Selection Committee.

To ensure the honesty of Selection Committee, we adopted *Cryptographic Sortition* method ([253]). Firstly, we designed that, to be eligible to join the system, the participants are required to lock stakes on the blockchain. Selection Committee is self-selected through Cryptographic Sortition based on their locked stakes under the assumption that number of deposited stakes indicates participant's honesty in the system. The cryptographic sortition based on locked stakes in the past ensures the Selection Committees are honest majority. This assumption is based on the reward and penalty policy related to the deposited stakes defined by the blockchain system. If users behave dishonestly, they will lose the stakes they locked during enrollment.

The combination of Selection and Maintenance Committees enables the DSKM scheme to secure global key pair regardless of the time span of the overall execution and dynamics of the participants. Furthermore, both committees only need to be online once enabling maximum flexibility for participants to achieve secure key distribution.

7.3.1 System Model

Fig. 7.3 illustrates a schematic of the proposed *Dynamic Scalable Distributed Key Generation (DSKM)* scheme. The DSKM scheme includes two main components, Designation Procedure (run by the Selection Committee), and the CBA line (run by the Maintenance Committee). Designation Procedure is designed to create the Selection and Maintenance Committees, whereas the CBA line mainly executes the Key Generation and Key Maintenance tasks.

The system functions within iterative epochs, and for each epoch we have one or both committees online. In the Designation Procedure, Selection Committee is self-selected and each of them nominates a participant to join Maintenance Committee. Afterwards, Maintenance Committee works on CBA line similar to assemblers who are in charge of an assembly line. CBA line begins with the Key Generation task, in which members of the first Maintenance Committee jointly generate global key pair. Subsequently, the members of the later Maintenance Committees work on CBA line for Key Maintenance task by securely distributing the global secret key from one committee to another.

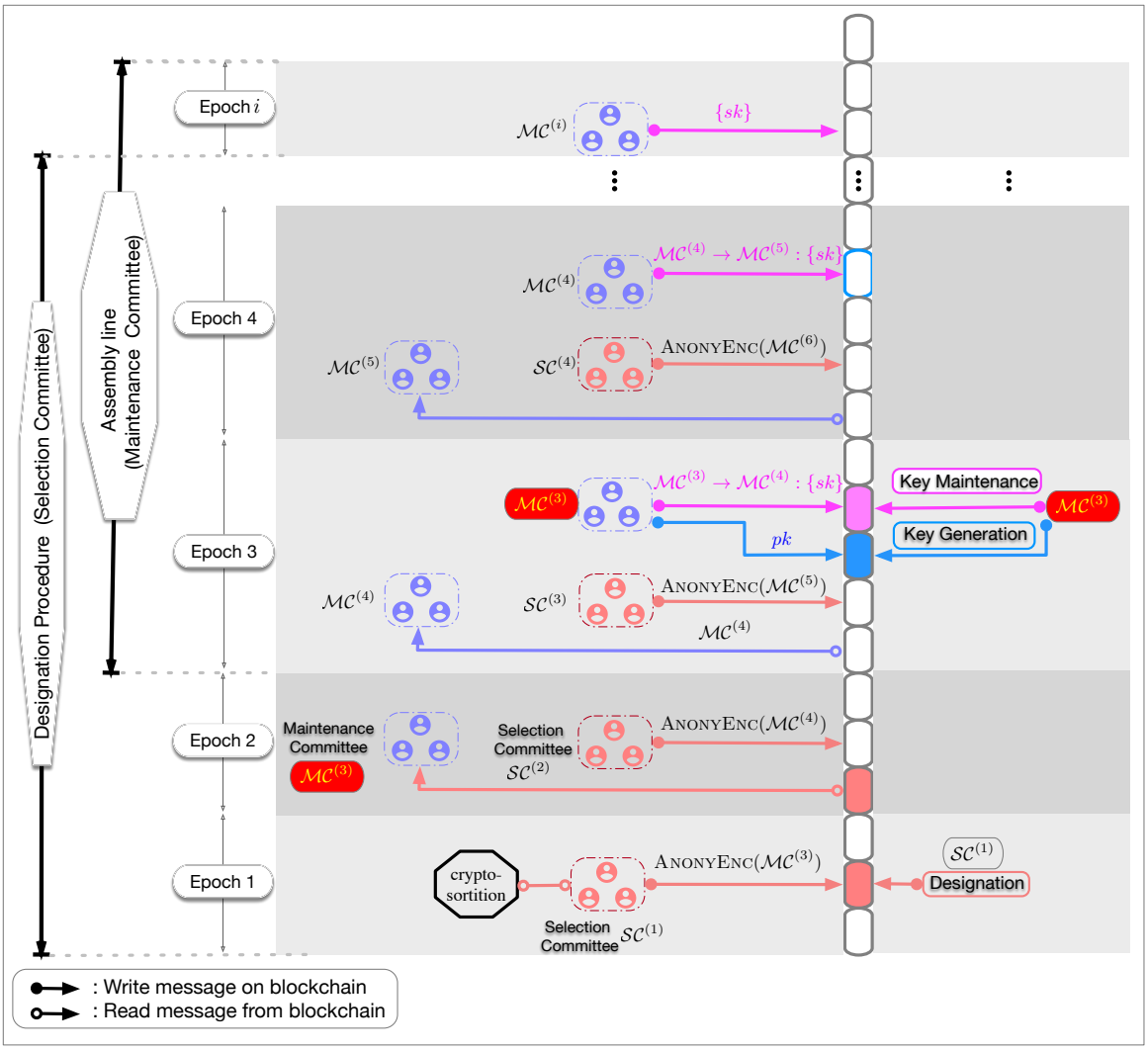

Figure 7.3: A schematic of DSKM scheme

We design DSKM scheme so that it can be used in other task-specific systems, *e.g.*, blockchain-based e-voting system, as an add-on plug-in. In the rest of this thesis, we simply refer to such systems as the "application system". We assume that a stake holder who wants to join the application system is required to lock some stakes on the blockchain during its enrollment to demonstrate their honesty. After successful enrollment, each participant, P_k, is assigned with a public key, pk_k, and a secret key, sk_k. In each epoch, a participant might play a role in the Selection Committee and/or Maintenance Committee.

We use several blocks in Fig. 7.3 to represent a blockchain. In the following sections, we first explain how to generate and maintain the key, and present Designation Procedure and CBA line in each epoch in the left part of Fig. 7.3. Then we clarify how to ensure that

the committee members only need to be online once in the right part of this figure. Next, we present the communication on a blockchain, participants registration in the system, and propose a new cryptographic sortition protocol.

7.3.1.1 Designation Procedure

Suppose that there are i epochs in DSKM scheme. Starting the first until the $(i-2)$-th epoch, *Selection Committee* is self-selected by the cryptographic sortition based on their stakes locked on the blockchain. In k-th epoch ($k \in [i-2]$), the Selection Committee is denoted by $\mathcal{SC}^{(k)} := \{\mathsf{SC}_1^{(k)}, \ldots, \mathsf{SC}_{s^{(k)}}^{(k)}\}$. Each member of Selection Committee, $\mathsf{SC}_a^{(k)}$, nominates one participant as the *Maintenance Committee* member, $\mathsf{MC}_a^{(k+2)}$, for epoch $k+2$. We denote this Maintenance Committee by $\mathcal{MC}^{(k+2)} := \{\mathsf{MC}_1^{(k+2)}, \ldots, \mathsf{MC}_{m^{(k+2)}}^{(k+2)}\}$[2].

Considering that global secret key is processed by different Maintenance Committees, it is required to ensure that the number of corrupted actions can not exceed the threshold of corresponding secret sharing scheme. Otherwise, adversary might be able to reconstruct the global secret key once it controls enough members of Maintenance Committee. This thesis addressed this problem from the following aspects:

- Selection Committee is self selected by cryptographic sortition and locking stakes on blockchain, which is ensured to be honest majority as described in Section 7.2.3;
- Selection Committee selects and announces Maintenance Committee in an anonymous way, to prevent adversary from identifying Maintenance Committee;
- Assume that the honest Selection Committee member chooses honest participant as a member of Maintenance Committee, and a member of dishonest Selection Committee chooses dishonest participant as a member of Maintenance Committee.

To keep Maintenance Committee anonymous, members of Selection Committee generate ephemeral/fake identities (namely, ephemeral public key and ephemeral secret key) for their nominees. Ephemeral public key is announced publicly on the blockchain. The ephemeral secret key is posted on the blockchain with anonymous encryption.

In this process, $\mathsf{SC}_a^{(k)}$ prepares an ephemeral key pair $(\mathsf{epk}_a, \mathsf{esk}_a)$ for $\mathsf{MC}_a^{(k+2)}$ and post $(\mathsf{ANONYENC}_{\mathsf{pk}_a}(\mathsf{esk}_a), \mathsf{epk}_a)$ on the blockchain, where $\mathsf{ANONYENC}$ is an anonymous encryption algorithm. Here in this thesis, Lifted ElGamal encryption is used as a candidate of anonymous encryption algorithm. In epoch $k+1$, the participant that wish to be selected in Maintenance Committee attempts to decrypt the cipher-texts on the blockchain with their own secret key. Once they succeed, they get a seat in the Maintenance Committee in epoch $k+2$. In epoch $k+2$, the Maintenance Committee, $\mathcal{MC}^{(k+2)}$, performs the key generation if $k=1$, otherwise they perform the key maintenance task.

[2]A member of the Selection Committee can only select one member of Maintenance Committee, i.e., $m^{(k+2)} = s^{(k)}$.

7.3.1.2 CBA line–Key Generation

In epoch 3, P_3 monitors the messages sent by $\mathcal{SC}^{(1)}$ in epoch 1, and checks if they are selected in $\mathcal{MC}^{(3)}$. Once the identities are determined, $\mathcal{MC}^{(3)}$ jointly generate global secret key and public key. Then, $\mathcal{MC}^{(3)}$ receives ephemeral public keys of $\mathcal{MC}^{(4)}$ sent by $\mathcal{SC}^{(2)}$ in epoch 2. Subsequently, $\mathcal{MC}^{(3)}$ share their partial secret keys to next Maintenance Committee, $\mathcal{MC}^{(4)}$, and post the encrypted shares on the blockchain for $\mathcal{MC}^{(4)}$ with $\mathcal{MC}^{(4)}$'s ephemeral public keys. The specific protocol has been described in Chapter 5

7.3.1.3 CBA line–Key Maintenance

For $a \in [4, i-1]$, $\mathcal{MC}^{(a)}$ reconstruct their partial secret keys by collecting the messages posted by $\mathcal{MC}^{(a-1)}$ on the blockchain. Then, $\mathcal{MC}^{(a)}$ re-distribute the shares to the next Maintenance Committee, $\mathcal{MC}^{(a+1)}$, by posting encrypted shares on the blockchain [3]. Through posting and reading messages on the blockchain, the prior Maintenance Committee handovers the secret key shares to the next Maintenance Committee as if they are playing a relay race. With the use of anonymous encryption, the real identities of Maintenance Committee members are not known by other members and they can keep the secret key shares secure from the adversary. The specific protocol has been described in Section 7.2.

7.3.1.4 How to ensure committees online only once?

As we explained in Sec. 7.3.1.2 and Sec. 7.3.1.3, in each epoch a Selection Committee member, $\mathsf{SC}_a^{(k)}$, needs to be online once to receive the self-selection result and nominate Maintenance Committee. However, a Maintenance Committee member $\mathsf{MC}_a^{(k+2)}$ needs to be online *twice*: 1) in epoch $k+1$, to know if it is chosen to be $\mathsf{MC}_a^{(k+2)}$ by fetching the messages sent by $\mathcal{SC}^{(k)}$ in epoch k, and to receive the ephemeral identities of $\mathcal{MC}^{(k+3)}$ by fetching the messages sent by $\mathcal{SC}^{(k+1)}$ in epoch $k+1$; 2) in epoch $k+2$, if $k=1$, to generate the global key pair; if $k \geq 2$, to reconstruct his secret key shares and redistribute it to $\mathcal{MC}^{(k+3)}$.

Integrity of blockchain guarantees that the record of messages sent by Selection Committee and Maintenance Committee is immutable. Therefore, Maintenance Committee can be online once and collect messages sent in previous epochs. As shown in the upper part of Fig. 7.3, $\mathsf{MC}_a^{(k+2)}$ only need to be online in epoch $k+2$, then it fetches messages in epoch $k+1$ and epoch $k+2$. By doing so, our proposed DSKM enables maximum flexibility for the participants.

[3] Owing to the traceability of blockchain, genuine identities of $\mathcal{MC}^{(a)}$ will be linked to his ephemeral identities once he sent out messages on the blockchain. We argue that this identity exposure won't affect the overall security as we consider static security here.

7.4 Summary

In this chapter, we have thoroughly investigated the construction of the evolving committee mechanism, which facilitates the dynamic changing of voting committees during a decision-making period. Specifically, we introduced the evolving ideal functionality and its corresponding protocol. Through the analysis of the honest majority probability of voting committees, we have demonstrated that the evolving protocol indeed realises the proposed evolving ideal functionality with UC framework.

In the Handover protocol, for a previous committee member, computation cost is $\mathcal{O}(n_{r+1})$, communication cost is $\mathcal{O}(\max(n_{r+1}, n_r))$. For a new committee member, the computation cost is $\mathcal{O}(\max(n_{r+1}, n_r))$, the communication cost is $\mathcal{O}(n_{r+1})$. The overall communication cost of $\Pi_{Handover}[\mathbb{G}]$ is $\mathcal{O}(\max(n_{r+1}, n_r) \cdot n_{r+1})$, and computation cost is $\mathcal{O}(\max(n_{r+1}, n_r)^2)$ for per key.

Furthermore, building upon the evolving committee construction and the key generation protocol, we explored the concept of using a much smaller subset of participants to represent the entire set of participants in distributed key generation. This led to the proposal of a novel Distributed Scalable Key Management (DSKM) scheme on a blockchain system, employing the Committee-Based Assembly line (CBA line) and the Designation Procedure. To the best of our knowledge, this is the first work that targets scalable threshold cryptosystems with fully dynamic participation on a blockchain system.

The DSKM scheme achieved optimal efficiency in scalability with fully dynamic participation, as evidenced by the following key attributes:

- The overall efficiency is independent of the number of participants, N, such as 2^{20}. Significantly smaller committee sizes, for instance, n with 10^2 members, are utilised to generate and maintain distributed key pairs.
- All committee members are only required to be online once during the entire process, avoiding repeated commitments and enhancing system efficiency.
- The DSKM scheme can stably run until the conclusion of the system, accommodating the joining of new participants in any epoch without necessitating a restart of the DKG process.

In the previous chapters, we have discussed various aspects of the privacy-preserving decision-making system, including the design of the Two Stage Voting (TSV) scheme, the construction of evolving committees, and the distributed key management scheme. However, one crucial aspect that has not been addressed yet is the incentive for voters and experts to actively participate and contribute to the decision-making process. In traditional voting systems, voters and experts may have various motivations to participate, such as the desire to influence the outcome, support their preferred proposals, or contribute to the betterment of the community.

In our proposed system, we aim to ensure active participation and meaningful contributions from voters and experts through a Reputation Management Scheme. This scheme will utilise the notion of reputation to incentivise voters and experts to behave honestly and responsibly during the decision-making process. Participants' reputation will be earned based on their actions and contributions, and it will be publicly visible within the system.

By incorporating the Reputation Management Scheme, we aim to create a self-sustaining ecosystem where participants are motivated to act in the best interest of the community and the decision-making process. The reputation earned by voters and experts will be a valuable asset that can be used to gain influence and recognition within the system. This, in turn, will foster a sense of responsibility and accountability among participants, leading to a more robust and effective decision-making process.

With the combination of evolving committees, distributed key management, and reputation management, our proposed system aims to achieve a high level of scalability, security, and fairness while incentivising active and responsible participation from voters and experts. In the forthcoming chapter, we will delve into the details of the Reputation Management Scheme and illustrate its role in fostering a thriving and reliable decision-making ecosystem.

Chapter 8

Building Block: Reputation Management Scheme

> Innovation doesn't come just from giving people incentives; it comes from creating environments where their ideas can connect.
>
> Steven Johnson

8.1 Overview

Reputation has emerged as a powerful tool in information systems to simulate and quantify trustworthiness based on participants' historical interactions ([255]). This concept has found applications in various domains, such as machine-to-machine communication systems ([256]), service provider networks ([257]), vehicular ad hoc networks ([258, 259, 260]), peer-to-peer energy trading ([261]), and even blockchain consensus mechanisms ([262, 263, 264]).

The reputation management scheme we propose in this chapter carefully analyses the behaviours and outcomes of these various participants to determine their reputation scores. Reputation serves as a powerful incentive for participants to act in the best interest of the decision-making ecosystem, as it can impact their influence and recognition within the system.

By integrating reputation management into our system, we aim to create a fair, reliable, and self-sustaining decision-making environment. Participants' reputation values will evolve over time based on their continuous interactions and contributions, reflecting their long-term reliability and trustworthiness. This will foster a sense of responsibility and accountability

among participants and enhance the overall efficiency and effectiveness of the decision-making process.

In the following sections, we will delve into the details of the reputation management scheme, its implementation, and its impact on incentivising active and honest participation from experts, voters, and proposal owners. The combination of evolving committees, distributed key management, and reputation management will contribute to the establishment of a robust, secure, and fair decision-making system that can thrive in real-world scenarios with a large number of participants.

8.2 Reputation Management Scheme

In the context of our proposed decision-making system, we use reputation management to encourage participation, significant contributions, and honesty among experts, voters, and proposal owners. Using reputation as a metric of trustworthiness allows for a more dynamic and transparent evaluation of participant actions and behaviours.

For instance, experts' reputation can be influenced by the quality and accuracy of their votes and contributions during the decision-making process. Voters, on the other hand, can earn reputation based on the validity and relevance of their choices and delegation decisions. Proposal owners may build their reputation by submitting high-quality and well-reasoned proposals.

The introduction of a reputation management system in the decision-making process is predicated on several foundational objectives:

- Enhancing Trust and Accountability: Reputation systems effectively simulate and measure trust in digital and decentralised environments. The system may measure and reward trustworthy and productive involvement by assessing their past actions and contributions.
- Improving Decision Quality: By linking reputation scores to the quality and outcomes of contributions, participants are motivated to provide thoughtful, high-quality inputs that enhance the overall decision-making process.
- Facilitating Self-Regulation: Reputation serves as a self-regulating mechanism within the community. Participants with higher reputations gain more influence, which encourages consistent and positive engagement within the system.
- Dynamic Participant Evaluation: Reputation systems allow for a nuanced assessment of participants over time, reflecting their long-term behaviours and contributions, which static evaluation methods cannot capture.

In the proposed reputation management scheme, we aim to objectively measure and quantify the activity of each participant (proposer/voter/expert) in the decision-making system by assigning them a dynamically updated reputation score specific to each field. To promote diversity and versatility in the decision-making process, we encourage participants to be active in different fields, and we calculate and compare their reputation scores based on their individual contributions in each field.

To accommodate the various roles that a participant may take on in different epochs and fields, we design the reputation score to be associated with a field label and aggregated using four reputation factors. Specifically, in the k-th treasury period, let $\mathsf{Rep}_{\mathsf{fld}}^{(k)}(u_i)$ denote the reputation score of participant u_i in the fld field. We use a weighted aggregated function,

denoted as AGG, to calculate the reputation score $\mathsf{Rep}_{\mathsf{fld}}^{(k)}(u_i)$ based on four partial reputation scores:

$$\mathsf{Rep}_{\mathsf{fld}}^{(k)}(u_i) := \mathsf{AGG}\Big(\mathsf{Rep\text{-}RW}^{(k)}(u_i), \mathsf{Rep\text{-}PC_{fld}}^{(k)}(u_i), \mathsf{Rep\text{-}FP_{fld}}^{(k)}(u_i), \mathsf{Rep\text{-}IM}_{\mathsf{fld}}^{(k)}(u_i)\Big) \tag{8.1}$$

- The reputation score $\mathsf{Rep\text{-}RW}_{u_i}^{(k)}$ serves as an indicator of a participant's general stickiness and consistency in the decision-making system. It reflects the partial reputation score gained from the participant's **regularity of work** and involvement during their participation. Whether a participant joins the decision-making system as an **expert** or a **voter**, their level of engagement and contributions should be reflected in their reputation score.

  We place a strong emphasis on participants regularly and consistently engaging in the decision-making process, regardless of the specific fields they are involved in. Participants are encouraged to contribute their knowledge, expertise, and opinions to the proposals, and their regular engagement is considered a crucial aspect of building a robust and functional decision-making system.

  By factoring in participants' regularity of work and involvement, we aim to incentivise a continuous and active presence of participants in the decision-making system. This regular engagement not only contributes to the accumulation of their reputation score but also directly impacts the usability and effectiveness of the decision-making system as a whole. A higher level of participant engagement enhances the quality of proposals, promotes a more diverse range of ideas, and ultimately leads to better-informed decisions.

  Participants are thus motivated to stay active and contribute regularly to the decision-making process to improve their reputation score and establish a positive and influential presence within the system. By recognizing and rewarding consistent engagement, the reputation management scheme fosters a thriving and vibrant decision-making ecosystem where participants are encouraged to actively participate and make meaningful contributions. This approach promotes a sustainable and efficient decision-making system that harnesses the collective expertise and involvement of its participants to reach informed and impactful decisions.

- The reputation score $\mathsf{Rep\text{-}PC_{fld}}^{(k)}(u_i)$ represents a participant's partial reputation score based on the **quality of total productive contributions** in the specific field fld. This score takes into account the constructive and valuable decisions made by the participant in their role as a voter or an expert. The decisions made by participants have a significant impact on the outcome of the decision-making process, and this influence is directly reflected in their reputation scores.

Participants are motivated to provide the decision-making system with constructive and insightful decisions that align with their areas of expertise, whether they are acting as **voters** or **experts**. Their decisions have the potential to shape the direction and outcome of proposals within the specific field they are involved in. As a result, participants are incentivised to make informed and thoughtful decisions that contribute positively to the decision-making process.

The quality of a participant's total productive contributions is a key factor in determining their reputation score. By consistently providing high-quality decisions and contributions, participants can earn a positive reputation and enhance their influence within the decision-making system. Conversely, participants who consistently provide low-quality or unconstructive decisions may see a decrease in their reputation score, impacting their level of influence.

This approach creates a dynamic and merit-based reputation system that rewards participants for their valuable contributions to the decision-making process. By aligning reputation with the quality of productive contributions, the reputation management scheme encourages participants to actively engage, share their expertise, and make meaningful contributions that positively impact the decision-making outcomes.

- The reputation score $\mathsf{Rep\text{-}FP}_{\mathsf{fld}}{}_{u_i}^{(k)}$ reflects a participant's partial reputation score related to the **winning rate** of their proposed proposals in the specific field fld. This score is applicable when the participant serves as a project proposer during their involvement in the decision-making system. The objective is to recognise and reward participants who have a track record of submitting valuable and successful proposals.

  In the decision-making system, participants have the opportunity to propose projects and initiative within their areas of expertise. These proposals are evaluated by the community and stakeholders, and some of them may receive funding or support based on their merit and alignment with the system's goals. The winning rate of a participant's proposed proposals is an indicator of their ability to come up with practical and worthwhile initiatives that gain community support.

  A participant who consistently submits successful and well-received proposals demonstrates a level of expertise and effectiveness in their field. Their ability to secure funding for their proposed projects indicates that they are not only knowledgeable but also capable of translating their ideas into tangible and impactful outcomes. Such participants are considered reputable and valuable contributors to the decision-making system.

  By incorporating the winning rate of proposed proposals into the reputation score, the reputation management scheme incentivizes participants to actively engage in proposing high-quality projects and initiatives. Participants are motivated to put forth

well-thought-out and promising proposals that align with the needs and priorities of the decision-making system. The more successful proposals a participant has, the higher their reputation score, reflecting their demonstrated track record of contributing positively to the system.

This approach encourages participants to be proactive and innovative in their role as project proposers. It fosters a culture of creativity and practicality, where participants are incentivised to propose ideas that have a high likelihood of success and positive impact. By rewarding participants who can effectively translate their expertise into successful proposals, the reputation management scheme strengthens the decision-making system's ability to identify and support valuable projects that benefit the community as a whole.

- The reputation score $\mathsf{Rep\text{-}IM}_{\mathsf{fld}}^{(k)}(u_i)$ represents the partial reputation score of **innovation management contribution** in the pre-voting epoch for experts. As described in Section 4.5, experts play a crucial role in finalising the proposal list before sending it to the next epoch for voting. This pre-voting phase is essential for ensuring the quality and relevance of the proposals that will be presented to the broader community for evaluation and decision-making.

  During the pre-voting epoch, experts collaborate to review and evaluate the proposed projects based on their expertise. Their collective goal is to select a proposal list that best aligns with the decision-making system's objectives and criteria. To ensure the legitimacy and effectiveness of the proposal list, it is required that the list is signed by a significant number of experts, whose total reputation scores account for more than 50% of all experts' reputation scores.

  The reputation score $\mathsf{Rep\text{-}IM}_{\mathsf{fld}}^{(k)}(u_i)$ is a measure of an expert's contribution to the pre-voting phase. It reflects their level of engagement and effectiveness in the process of reviewing and finalizing the proposal list. If an expert actively participates in the pre-voting epoch and contributes to the finalization of the proposal list by signing it, their $\mathsf{Rep\text{-}IM}_{\mathsf{fld}}^{(k)}(u_i)$ value will be non-zero. On the other hand, if an expert does not contribute to the proposal list generation by not signing it, their $\mathsf{Rep\text{-}IM}_{\mathsf{fld}}^{(k)}(u_i)$ value will be zero.

  By incorporating $\mathsf{Rep\text{-}IM}_{\mathsf{fld}}^{(k)}(u_i)$ into the reputation score, the reputation management scheme incentivises experts to actively engage in the pre-voting phase and take their responsibility seriously. Experts are motivated to carefully review and evaluate the proposed projects and provide their valuable input to the decision-making process. The more active and effective an expert is in contributing to the final proposal list, the higher their reputation score in the **innovation management contribution** category.

  This approach ensures that experts are actively involved in the critical phase of proposal selection, and their reputation scores reflect their level of commitment and

contribution to the decision-making system's success. By incentivising experts to take their roles seriously and make informed decisions during the pre-voting epoch, the reputation management scheme enhances the overall quality and effectiveness of the decision-making process, leading to more robust and impactful outcomes for the entire community.

The reputation management scheme employs the AGG function to combine the four partial reputation scores, $\mathsf{Rep\text{-}RW}^{(k)}u_i$, $\mathsf{Rep\text{-}PC_{fld}}^{(k)}(u_i)$, $\mathsf{Rep\text{-}FP_{fld}}^{(k)}u_i$, and $\mathsf{Rep\text{-}IM}^{(k)}_{\mathsf{fld}}(u_i)$, with appropriate weights to derive the overall reputation score $\mathsf{Rep}^{(k)}_{\mathsf{fld}}(u_i)$ for participant u_i in the specific field fld for the k-th treasury period. Each of these partial reputation scores represents a different aspect of a participant's contributions and behaviours within the decision-making system. By considering these four different aspects of participants' contributions and behaviours, the reputation management scheme provides a comprehensive and objective evaluation of their performance.

To facilitate the understanding of the reputation algorithm RepCal in Figure 8.1, we provide a summary of frequently used notations in Table 8.1. In each updating treasury period k, we calculate the reputation score $\mathsf{Rep}_{\mathsf{fld}}(u_i)^{(k)} \in [0, 1]$ of participant u_i in different fields indexed by fld.

Participants may not continuously participate in the decision-making system, so we use $(k_0, \ldots, k)$ to represent the discontinuous or continuous periods when a participant has participated, and use T to denote the total number of periods the participant has participated.

In our system, each participant may take on one or more entity roles. Therefore, we introduce the role concept $\mathsf{Role} \in \mathcal{P}\mathsf{fld}^{(k)}, \mathcal{V}\mathsf{fld}^{(k)}, \mathsf{E}^{(k)}\mathsf{fld}$ when computing the reputation score. Here, $\mathcal{P}\mathsf{fld}^{(k)}$ represents the role of a proposal proposer, $\mathcal{V}^{(k)}_{\mathsf{fld}}$ represents the role of a voter, and $\mathsf{E}^{(k)}_{\mathsf{fld}}$ represents the role of an expert. In each treasury period, the participant's role will influence how the four reputation factors are computed, as described below.

- **Regularity of work.**

  As mentioned earlier, voters who want to participate in the decision-making system are required to freeze a certain number of stakes on the blockchain, which will become their voting power. Additionally, experts will gain voting power from delegations made by voters. Thus, to assess their regularity of work, we introduce the concept of voting power ratio $\mathsf{VPR}^{(t)}(u_i)t \in (k_0, \ldots, k)$, which is computed by dividing the participant's personal voting power by the total voting power in each treasury period. This voting power ratio is illustrated in Algorithm 8.1 as $\mathsf{Rep\text{-}RW}^{(k)}u_i$.

  The voting power ratio $\mathsf{VPR}^{(t)}(u_i)$ quantifies the extent of a participant's engagement in the decision-making process over time. A higher voting power ratio indicates that the participant consistently contributes their voting power, while a lower ratio suggests sporadic or irregular engagement. Participants are incentivised to maintain a high voting power ratio by actively participating in the decision-making process and

using their voting power regularly to support proposals aligned with their interests and expertise.

In order to measure the regularity with which a participant contributes voting power to the decision-making system, we first calculate the participant's average voting power ratio $\overline{\mathsf{VPR}}$ over all T treasury periods. The average voting power ratio provides an indication of the participant's overall level of engagement and consistency in using their voting power.

Next, we assess the variability or fluctuations in the participant's voting power contributions over time, which is indicated by the standard deviation of the voting power ratio SD_{VPR}. A higher SD_{VPR} value suggests that the participant's voting power contributions are more irregular or volatile, while a lower value indicates more consistent and stable contributions.

By combining the participant's average voting power ratio $\overline{\mathsf{VPR}}$ with the standard deviation SD_{VPR}, we can compute the participant's partial reputation score $\mathsf{Rep\text{-}RW}_{u_i}^{(k)}$. This approach ensures that the reputation score reflects not only the participant's overall level of engagement (as captured by the average voting power ratio) but also the regularity and stability of their voting power contributions (as captured by the standard deviation). Participants who consistently and regularly contribute their voting power to the decision-making system will be rewarded with higher reputation scores, while those with more sporadic or irregular contributions will receive lower scores.

An important consideration in computing the reputation factor $\mathsf{Rep\text{-}RW}_{u_i}^{(k)}$ for experts is that they are trusted acquiescently due to their high reputation scores, and as such, they do not need to cast stakes as voting power to demonstrate their honesty and loyalty. Therefore, if a participant has served as an expert in a certain treasury period $h \in (k_0, \ldots, k)$, then their voting power ratio $\mathsf{VPR}^{(h)}(u_i)$ should be zero.

This distinction is essential to ensure that experts are not penalised for not contributing voting power in periods where they have served as experts. Instead, their reputation and trustworthiness are implicitly acknowledged, and they are exempt from the requirement of casting stakes as voting power during their tenure as experts.

- **Quality of total productive contributions.**

  In the decision-making system, participants are strongly encouraged to make the right decisions that reflect their own expertise and knowledge. In each treasury period, proposals may span multiple fields, and participants may be involved in decisions across different fields. Therefore, we calculate a specific part of the reputation score, denoted as $\mathsf{Rep\text{-}PC_{fld}}^{(k)}(u_i)$, based on the quality of the participant's total productive contributions in each field they have participated in.

  To promote diversity of opinions and active engagement, we incentivise participants to express their own opinions rather than consistently delegating their voting power to

experts. Thus, the reputation score $\mathsf{Rep\text{-}PC_{fld}}^{(k)}(u_i)$ reflects the outcomes of decisions that the participant has made personally, rather than decisions made through delegation.

By focusing on the participant's direct contributions and decisions in each field, the reputation management scheme fosters a more robust and diverse decision-making process. Participants are motivated to actively participate, contribute their expertise, and make well-informed decisions that align with their own knowledge and values. This approach empowers participants to have a direct impact on the system's outcomes and ensures that their reputation scores accurately reflect their individual contributions and decision-making abilities.

To quantify participants' personal contributions in decision-making, we introduce two metrics: the number of projects u_i voted **Yes** by themselves, denoted as $Y_{\mathsf{fld}}^{(k_0,\ldots,k)}(u_i)$, and the number of projects u_i voted **No** by themselves, denoted as $N_{\mathsf{fld}}^{(k_0,\ldots,k)}(u_i)$. These metrics allow us to define the participants' personal contributions sign Δ.

If a participant has not voted for any projects (either YES or NO) by themselves during the periods $(k_0, \ldots, k)$, their Δ will be 0, and they will receive a zero score in this reputation factor. On the other hand, if a participant has made at least one independent vote (either YES or NO), their Δ will be 1, indicating their active personal contributions to the decision-making process.

The use of the personal contribution sign Δ ensures that participants are incentivised to engage directly in the decision-making process and express their own opinions. By taking into account participants' individual votes, the reputation management scheme encourages a diverse range of perspectives and promotes the active participation of individuals in shaping the system's outcomes.

The accuracy rate $\{\mathsf{ACC_{fld}}^{(t)}(u_i)\}_{t\in[T]}$ for participant u_i in the t-th period is defined as the percentage of proposals supported by u_i that were included in the list of winning projects during that period. In other words, it measures the success rate of the proposals supported by the participant. The periods are indexed by $t \in [T]$, where k_0 corresponds to the first period and k corresponds to the T-th period.

However, if a participant has served as an expert in any of the periods $(k_0, \ldots, k)$, their accuracy rate in that specific period will be zero. This is because experts are already considered to have high reputation scores and their proposals are trusted without being subjected to the accuracy evaluation.

To calculate the partial reputation score $\mathsf{Rep\text{-}PC_{fld}}^{(k)}(u_i)$, we use $\mu^{(k)}$ as the average accuracy rate since the participant joined the decision-making system (during the periods $(k_0, \ldots, k)$). This $\mu^{(k)}$ represents the fraction of positive work that a participant has contributed to the entire system, reflecting the quality and impact of their proposals on the overall decision-making process.

Even reputation is calculated on participant's previous behaviours, as mentioned by Josang *et al*. [265], a participant's behaviour in the last few days is a more accurate factor of the participant's future behaviour than analysing all previous behaviour on the network. Thus, instead of computing the normal average, we calculate the exponential moving average (EMA) $\mu^{(t)}$ of accuracy rate $\mathsf{ACC}_{\mathsf{fld}}{}^{(t)}(u_i)$. Consequently, $\mu^{(t)}$ provides multiplying factors to give different weights to accuracy rate at different time during periods $(k_0, \ldots, k)$.

By using the EMA, we ensure that the reputation score $\mathsf{Rep\text{-}PC}_{\mathsf{fld}}{}^{(k)}(u_i)$ is influenced more by the participant's recent performance, rather than their entire historical behaviour. This approach aligns with the idea that recent behaviour is a better indicator of a participant's future behaviour, and it allows the reputation management scheme to adapt to changes in a participant's contributions over time. Participants who consistently make high-quality proposals and have a positive impact on the decision-making system will be rewarded with higher reputation scores, reflecting their ongoing commitment to the success and improvement of the system.

We set α as a smoothing factor that can be adjusted to make the reputation factor $\mathsf{Rep\text{-}RW}_{u_i}^{(k)}$ more or less progressive. The closer that α gets to 0, the more weight will be assigned to the initial accuracy rate in earlier projects. On the contrary, the closer that α gets to 1, the recent accuracy rate that participant gets is more important. Combining EMA of accuracy rate $\mu^{(k)}$ with standard deviation of accuracy rate $SD_{\mathsf{ACC}_{\mathsf{fld}}}$ and personal contribution sign Δ, we can get $\mathsf{Rep\text{-}PC}_{\mathsf{fld}}{}^{(k)}(u_i)$ as participant's total productive personal contribution to the decision-making system in fld field;

- **Winning rate.**

  To evaluate the reputation factor $\mathsf{Rep\text{-}FP}_{\mathsf{fld}}{}_{u_i}^{(k)}$, which is related to a participant's performance as a project proposer, we calculate the winning rate $\mathsf{PWR}_{\mathsf{fld}}{}^{(k_0,\ldots,k)}$ of proposals submitted by participant u_i during the periods $(k_0, \ldots, k)$. The winning rate is defined as the percentage of proposals submitted by the participant that have been funded or accepted.

  Specifically, let $N_{\text{total}}^{(k_0,\ldots,k)}(u_i)$ be the total number of proposals submitted by participant u_i during the periods $(k_0, \ldots, k)$, and let $N_{\text{funded}}^{(k_0,\ldots,k)}(u_i)$ be the number of proposals among those that have been funded. The winning rate is then computed as:

$$\mathsf{PWR}_{\mathsf{fld}}{}^{(k_0,\ldots,k)} = \frac{N_{\text{funded}}^{(k_0,\ldots,k)}(u_i)}{N_{\text{total}}^{(k_0,\ldots,k)}(u_i)}. \tag{8.2}$$

  The reputation factor $\mathsf{Rep\text{-}FP}_{\mathsf{fld}}{}_{u_i}^{(k)}$ is simply the winning rate $\mathsf{PWR}_{\mathsf{fld}}{}^{(k_0,\ldots,k)}$ of participant u_i as a project proposer during the periods $(k_0, \ldots, k)$. This factor reflects the participant's ability to submit valuable and winning proposals, which contributes

to the overall success and effectiveness of the decision-making system. Participants who consistently submit high-quality proposals that are accepted and funded will receive higher reputation scores, signifying their reliability and valuable contributions as project proposers.

- **Innovation management contribution.**

  When a participant served as expert, it is asked to jointly generate proposal list with other committee members for innovation management. Since the final winning projects list is heavily dependent on this proposal list, we need to regulate expert's decision by adjusting their reputation scores based on the decision made in the pre-voting epoch. As mentioned before, a group of experts should sign the proposal list, and the proposal list will only be valid if the total reputation scores of this group are more than 50% of the expert's reputation scores.

  If a participant (expert) did not sign the proposal list, it indicates that they might not agree with the decision made during the pre-voting epoch. As a result, their partial reputation score based on their innovation management contribution, denoted as $\mathsf{Rep\text{-}IM}_{\mathsf{fld}}^{(k)}(u_i)$, should be set to 0. On the other hand, if an expert did sign the proposal list, it signifies their agreement and endorsement of the decisions made during the pre-voting epoch. In this case, we directly assign a predefined score, denoted as β, to their $\mathsf{Rep\text{-}IM}_{\mathsf{fld}}^{(k)}(u_i)$.

After computing the aforementioned two reputation factors, we aggregate them based on difference weights associated with different reputation factors, which is

$$w_1 \cdot \mathsf{Rep\text{-}RW}^{(k)}(u_i) + w_2 \cdot \mathsf{Rep\text{-}PC_{fld}}^{(k)}(u_i) + w_3 \cdot \mathsf{Rep\text{-}FP_{fld}}^{(k)}(u_i) + w_4 \cdot \mathsf{Rep\text{-}IM}_{\mathsf{fld}}^{(k)}(u_i). \quad (8.3)$$

In our reputation management scheme, when weighting $\mathsf{Rep\text{-}RW}$ and $\mathsf{Rep\text{-}PC_{fld}}$, we carefully set the weight w_1 to be smaller than w_2. This choice is deliberate to ensure fairness in the evaluation of participants' reputation scores. If w_1 was greater than w_2, participants with more voting power would gain disproportionately higher reputation scores compared to experts. Such an imbalance could undermine the integrity and fairness of our reputation management scheme.

To derive the final reputation score x for a participant, we multiply the weighted aggregated result by the current treasury period number k. This design aligns with our goal of prioritising recent contributions and considering them more heavily in the computation of reputation scores. As suggested by Josang *et al*. [265], a participant's behaviour in the last few days is a more accurate predictor of their future behaviour than analysing all previous behaviour on the network. Thus, by multiplying the weighted aggregated result by k, we give higher importance to recent activities, ensuring that reputation scores gained from recent engagements are higher.

When calculating the final reputation score based on the four different reputation factors, our aim is to define precise and parameterisable social objectives for reputation management. These objectives are carefully designed to achieve specific goals within the decision-making system:

- *Careful Start*: We intend to ensure a cautious and deliberate start for participants' reputation scores. To achieve this, we configure the reputation algorithm to initiate with a slow and gradual increase in reputation scores. This approach prevents new participants from being immediately trusted and encourages them to demonstrate consistent and honest behaviors over time.

- *Potential for Quick Reward of Mature Participants*: As participants build a history of trustworthy actions and contributions, we want to reward them with the potential for rapid reputation score growth. This is accomplished by allowing the reputation scores to increase exponentially during mid-life, reflecting the participants' maturity and established trustworthiness.

- *Prevention of Over-Control*: To maintain a balanced distribution of influence and prevent excessive concentration of power, we employ a mechanism to slow down reputation score growth as participants approach the upper limit inherent in the decision-making system. This ensures that reputation scores do not grow indefinitely and help to prevent over-control by a small group of participants.

To achieve these social objectives, we utilise a sigmoid function, which allows reputation scores to grow very slowly at the start, enabling new participants to gradually build trust. As participants demonstrate continued engagement and honest behaviours, the growth rate accelerates, providing quick rewards for mature participants with established trustworthiness. However, as reputation scores approach the upper limit, the growth rate slows down and eventually reaches a plateau, preventing excessive concentration of reputation and maintaining a fair distribution of influence.

After computing all four reputation factors, we aggregate them based on different weighting parameters, carefully chosen to reflect the importance of each factor in determining participants' overall reputation scores. This parameterisation allows us to fine-tune the reputation management scheme to achieve the desired social objectives and create a balanced, dynamic, and effective decision-making ecosystem.

In detail, to regulate the progression of reputation scores and confine them within the range of $[0, 1]$, we employ a sigmoid function denoted by $f(x)$. This sigmoid function is parameterised by (a, λ), allowing us to dynamically adjust the growth rate and slope of reputation scores. By using the sigmoid function, we control the rate at which participants' reputation scores increase, ensuring that newcomers to the system are not immediately granted high reputation scores. Instead, they must demonstrate honesty and loyalty in the

decision-making system for an extended period before their reputation scores experience substantial growth.

The sigmoid function works as follows: when the participant's reputation score x is equal to the parameter a, the sigmoid function $f(x)$ reaches its inflection point. After this point, the growth rate of reputation scores starts to decline. By carefully selecting the parameters (a, λ), we can effectively control the shape and progression of the sigmoid function, providing a fair and balanced approach to reputation score growth.

By utilising the sigmoid function, we establish a reputation management scheme that ensures participants' reputation scores evolve gradually in response to their continuous and positive contributions to the decision-making system. This dynamic approach fosters an environment where newcomers can gradually build trust through consistent involvement and honest behaviours. As participants accumulate a longer history of honest and reliable actions, their reputation scores progressively increase, approaching the turning point of the sigmoid curve.

At the turning point, participants' reputation scores experience accelerated growth, incentivising them to further engage and contribute to the decision-making system. This mechanism ensures that participants who have demonstrated trustworthiness over an extended period are rewarded with a more rapid increase in their reputation scores. This encourages participants to remain actively involved and continue behaving honestly, knowing that their reputation will grow more quickly and grant them greater influence in the system.

However, as participants' reputation scores approach the plateau of the sigmoid curve, the growth rate starts to decline. This prevents reputation scores from increasing indefinitely and promotes a balanced distribution of power amongst participants. The sigmoid function creates a natural equilibrium where reputation scores stabilise at a certain level, reflecting the participants' established trustworthiness and contributions to the decision-making system.

In addition to the behaviours and contributions calculated within the decision-making system, our reputation management scheme also takes into consideration the public reputation of participants from external sources. This public reputation is denoted by $\mathsf{Rep\text{-}Pub}_{\mathsf{fld}} \in [0, 1]$ and reflects the fundamental credibility of participants. It is independent of the decision-making system and is participant-facing, implying their general trustworthiness and reputation in other contexts.

When a knowledgeable and renowned expert joins the voting system, it is reasonable to assume that they should have a higher initial reputation than other ordinary new participants. This public reputation score serves as an initial reputation level for each participant, representing their pre-existing reputation before participating in the decision-making system. Participants with higher public reputation scores are deemed to have a higher level of initial trust within the system.

We use a binary rating scheme to represent the credibility of participants, denoted by $\mathsf{CR}u_i \in 0, 1$. A value of 1 indicates that the participant is considered honest and trustworthy, while a value of 0 indicates that the participant has misbehaved or been involved in dishonest

activities. If a participant's $\mathsf{CR}u_i$ is set to 0, their reputation score will be permanently set to 0 as well. In other words, they will be blacklisted by the reputation management scheme due to their lack of trustworthiness.

By incorporating both the public reputation score $\mathsf{Rep\text{-}Pub}_{\mathsf{fld}}$ and the credibility rating CR_{u_i} into the reputation management scheme, we ensure that participants with a strong pre-existing reputation are given appropriate initial trust within the decision-making system. At the same time, participants who have violated trust in the past are not allowed to gain reputation and influence within the system, promoting a trustworthy and reliable decision-making environment.

Table 8.1: Notations in RepCal

Variables	Explanation
k_0	the first period when participant joined
k	current treasury period
T	total treasury periods u_i joined including the k-th period
fld	field of reputation score
$\mathsf{Rep}_{\mathsf{fld}}(u_i)^{(k)}$	reputation score of participant u_i in fld field in k-th treasury period
$\mathsf{Role} \in \{\mathcal{P}_{\mathsf{fld}}^{(k)} \cup \mathcal{V}_{\mathsf{fld}}^{(k)} \cup \mathsf{E}_{\mathsf{fld}}^{(k)}\}$	The role of participant in k-th period
$\{\mathsf{VPR}^{(t)}(u_i)\}_{t\in(k_0,\ldots,k)}$	the voting power ratio casted by u_i in t-th period
$\mathsf{ConsR}(u_i)^{(k)}$	Consensus ratio, if u_i made same decision as other experts in Pre-voting epoch, then $\mathsf{ConsR}(u_i)^{(k)} = 1$
ω	Consensus ratio smoothing factor
$Y_{\mathsf{fld}}^{[k_0,\ldots,k]}(u_i)$	the number of projects u_i voted YES by himself in filed fld from k_0-th period to k-th period
$N_{\mathsf{fld}}^{[k_0,\ldots,k]}(u_i)$	the number of projects u_i voted NO by himself in filed fld from k_0-th period to k-th period
$\mathsf{Sign}_{\mathsf{fld}}^{(k)}(u_i) \in \{0,1\}$	if u_i is an expert in period k and it signed the proposal list in filed fld, then $\mathsf{Sign}_{\mathsf{fld}}^{(k)}(u_i) = 1$; otherwise, $\mathsf{Sign}_{\mathsf{fld}}^{(k)}(u_i) = 0$.
$\{\mathsf{ACC}_{\mathsf{fld}}{}^{(t)}(u_i)\}_{t\in[T]}$	the percentage of proposals supported by u_i in filed fld entered the list of winning projects when he joined the t-th time
$\mathsf{PWR}_{\mathsf{fld}}{}^{(k_0,\ldots,k)}$	winning rate of proposals proposed by u_i as a proposer from k_0-th period to k-th period in fld field
$\alpha \in (0,1)$	a constant smoothing factor
(w_1, w_2, w_3, w_4)	reputation weighting parameters where w_2 should be superior than w_1
(a, λ)	reputation system parameters
CR_{u_i}	participant's credibility, if u_i is honest, $\mathsf{CR}_{u_i} = 1$; otherwise $\mathsf{CR}_{u_i} = 0$
$\mathsf{Rep\text{-}Pub}_{\mathsf{fld}}$	the optional external source of reputation for u_i

Algorithm RepCal

Input: fld; k_0; k; α; (w_1, w_2, w_3, w_4); (a, λ); CR_{u_i}; $\mathsf{Rep\text{-}Pub}_{\mathsf{fld}}$. $\{\mathsf{VPR}^{(t)}(u_i)\}_{t\in(k_0,\ldots,k)}$; T; Role; ω;

Output: $\mathsf{Rep}_{\mathsf{fld}}(u_i)^{(k)} \in [0, 1]$

Regularity of Work:

- $\overline{\mathsf{VPR}} := \frac{1}{T}\sum_{t\in(k_0,\ldots,k)} \mathsf{VPR}^{(t)}(u_i)$
- $SD_{\mathsf{VPR}} := \sqrt{\frac{\sum_{t\in(k_0,\ldots,k)}(\mathsf{VPR}^{(t)}(u_i)-\overline{\mathsf{VPR}})^2}{T}}$
- $\mathsf{Rep\text{-}RW}_{u_i}^{(k)} := \frac{\overline{\mathsf{VPR}}}{1+SD_{\mathsf{VPR}}}$ if Role $\in \{\mathcal{P}_{\mathsf{fld}}^{(k)} \cup \mathsf{E}_{\mathsf{fld}}^{(k)}\}$, 0 otherwise;

Quality of total productive contributions:

- If $Y_{\mathsf{fld}}^{[k_0,\ldots,k]}(u_i) + N_{\mathsf{fld}}^{[k_0,\ldots,k]}(u_i) \geq 1$
  - $\Delta_{\mathsf{fld}} := 1$;
  - else $\Delta_{\mathsf{fld}} := 0$.
- $\mu^{(1)} := \mathsf{ACC}_{\mathsf{fld}\,1}(u_i)$, $S^{(1)} := 0$;
- for $t = [2, T]$
  - $\mu^{(t)} := (1-\alpha)\mu^{(t-1)} + \alpha \cdot \mathsf{ACC}_{\mathsf{fld}}{}^{(t)}(u_i)$;
  - $S^{(t)} := (1-\alpha)S^{(t-1)} + \alpha(\mathsf{ACC}_{\mathsf{fld}}{}^{(t)}(u_i) - \mu^{(t-1)})^2)$;
- $SD_{\mathsf{ACC}_{\mathsf{fld}}} := \sqrt{S^{(t)}}$, $\mathsf{Rep\text{-}PC}_{\mathsf{fld}}{}^{(k)}(u_i) := \Delta\frac{\mu^{(k)}}{1+SD_{\mathsf{ACC}_{\mathsf{fld}}}}$ if Role $\in \mathcal{P}_{\mathsf{fld}}^{(k)}$, 0 otherwise;;

winning rate:

- $\mathsf{Rep\text{-}FP}_{\mathsf{fld}}{}_{u_i}^{(k)} := \mathsf{PWR}_{\mathsf{fld}}{}^{(k_0,\ldots,k)}$ if Role $\in \mathcal{P}_{\mathsf{fld}}^{(k)}$, 0 otherwise;

Innovation Management Contribution:

- $\mathsf{Rep\text{-}IM}_{\mathsf{fld}}^{(k)}(u_i) = \omega \cdot log_{10}(\mathsf{ConsR}(u_i)^{(k)}) + 1$, if Role $\in \mathsf{E}_{\mathsf{fld}}^{(k)}$, $\mathsf{Sign}^{(k)}(u_i) = 1$;

Reputation Factors Aggregation:

- $x = (w_1 \cdot \mathsf{Rep\text{-}RW}^{(k)}(u_i) + w_2 \cdot \mathsf{Rep\text{-}PC}_{\mathsf{fld}}{}^{(k)}(u_i) + w_3 \cdot \mathsf{Rep\text{-}FP}_{\mathsf{fld}}{}^{(k)}(u_i) + w_4 \cdot \mathsf{Rep\text{-}IM}_{\mathsf{fld}}^{(k)}(u_i)) \cdot k$;
- $f(x) = \frac{1}{2} \cdot (1 + \frac{x-a}{\lambda+|x-a|})$;

Output:

- $\mathsf{Rep}_{\mathsf{fld}}(u_i)^{(k)} = min(1, \mathsf{CR}_{u_i} \cdot (\mathsf{Rep\text{-}Pub}_{\mathsf{fld}}{}_{u_i}^{(k_0)} + f(x)))$;

Figure 8.1: Reputation Computation Algorithm RepCal

8.3 Summary

In this chapter, we have presented a comprehensive and innovative reputation management scheme that dynamically updates the reputation values of proposers, voters, and experts based on their behaviours and contributions within each decision-making period. To achieve this, we divided the overall reputation value into four distinct reputation factors, each capturing the participant's performance and involvement in different roles and fields. These factors include the regularity of voting power contributions, the quality of total productive contributions, the winning rate of proposed projects, and the innovation management contribution for experts.

By carefully designing the reputation factors and incorporating them into our reputation management scheme, we ensure a fair and objective evaluation of participants' contributions, behaviours, and expertise in the decision-making system. This dynamic approach allows newcomers to gradually build trust through consistent involvement and honest behaviours, while also rewarding long-standing participants for their sustained engagement and dedication to the system. The use of a Sigmoid function further regulates the reputation score growth rate, promoting a balanced and gradual progression of reputation values.

Our reputation management system presents a broader and more versatile application of reputation metrics compared to the reputation-based consensus mechanisms described in [266, 267], and others such as ZkRep [268] and PoRX [264]. Unlike these approaches, which primarily utilise reputation to select nodes for block proposal and influence blockchain consensus, our system integrates reputation into various facets of decentralised decision-making, extending beyond blockchain operations. This includes influencing proposal submissions, voting behaviours, and participant engagement across different domains. This multifaceted approach not only fosters sustained engagement and incentives high-quality, honest contributions but also enhances the robustness of the system against malicious behaviour through continuous, dynamic updates of reputation scores based on diverse participant activities.

The integration of a reputation management system within our decision-making framework introduces notable benefits by incentivising participant engagement and ensuring reliability. However, this approach also presents substantial privacy challenges that are critical to address. Key privacy risks include the potential for inference attacks, where adversaries could deduce individual behaviours from reputation changes, and data correlation risks that could compromise participant anonymity.

To mitigate these initial concerns, this chapter has proposed the preliminary adoption of pseudonymization techniques to protect participant identities and suggested the potential for differential privacy to obscure exact reputation scores. These measures are intended to provide a foundational layer of privacy protection, ensuring that the reputation system enhances decision-making without compromising the confidentiality of participant actions.

Throughout the previous chapters, we have meticulously constructed the fundamental building blocks of our proposed decision-making system. The Distributed Key Generation

(DKG) protocol, the Two-Stage Voting (TSV) scheme, the construction of evolving committees, the distributed key management scheme and the Reputation Management scheme collectively form the backbone of our innovative approach. Each component has been designed to ensure privacy, verifiability, scalability, and trustworthiness, creating a holistic decision-making system that addresses the challenges faced by traditional centralised and non-private methods.

In the upcoming chapter, we will delve into the comprehensive implementation details of our proposed decision-making system. This chapter will serve as a bridge between theoretical design and practical realisation, providing readers with a hands-on understanding of how our system functions in a real-world setting.

Our focus will be on presenting benchmark tests and conducting rigorous performance evaluations of prototype implementations of the decision-making system. These practical tests will offer valuable insights into the effectiveness and efficiency of our proposed scheme, validating its functionality and robustness in diverse real-world scenarios. By subjecting the system to various testing scenarios, we aim to demonstrate its adaptability, resilience, and scalability, reinforcing its applicability in complex decision-making processes.

In the course of the next chapter, we will present detailed methodologies for carrying out the benchmark tests, illustrating how each building block of the decision-making system performs. We will provide concrete metrics and measurements to quantify the system's performance.

Additionally, the next chapter will explore the impact of different parameter settings on the system's performance, offering insights into optimisation strategies for achieving the best possible results. We will also assess the system's security aspects, ensuring that it remains robust against potential attacks or adversarial behaviours.

Our ultimate goal with these implementation details and benchmark tests is to demonstrate the viability of our decision-making system in real-world applications. We seek to provide stakeholders, developers, and decision-makers with tangible evidence of its efficacy, offering them a compelling incentive to adopt this innovative solution for their decision-making needs.

By merging theoretical foundations with practical demonstrations, we aspire to establish a seamless and trustworthy decision-making system that revolutionises governance processes across various domains. Through empirical evidence and rigorous evaluations, we aim to cement the position of our proposed scheme as a leading-edge solution, redefining the standards of accuracy, fairness, and security in decision-making systems.

Chapter 9

Implementation and Performance

Security is a process, not a product.

Bruce Schneier

9.1 Overview

In this chapter, we present the successful implementation of a prototype of our proposed decision-making system on the blockchain, as outlined in Chapter 4. The implementation of the system involved careful consideration of various components, including the pre-voting epoch and voting epoch, to ensure seamless and efficient operation. We conducted a comprehensive evaluation of the core protocols used in the system to validate its performance and efficiency.

In Section 9.2, we provide an in-depth explanation of how we translated the theoretical design of the decision-making system into a working prototype. This section outlines the technical details of the implementation, covering the various stages of the decision-making process, from the initial pre-voting epoch, where proposals are generated and curated, to the voting epoch, where participants cast their votes on the selected proposals, and finally, the post-voting epoch, where the winning proposals are implemented, and participants receive penalties or rewards.

Throughout the implementation process, we ensured that the system's key components, such as the distributed batched key generation protocol from Chapter 5, the two-stage voting scheme from Chapter 6, and the evolving committee mechanism from Chapter 7, were seamlessly integrated to create a cohesive and efficient decision-making system.

In Section 9.3, we conduct a detailed evaluation of the proposed protocols in the system. This evaluation process involves rigorous testing and performance analysis of each protocol to assess its effectiveness and efficiency. We focus on key metrics, such as response times,

computational resource consumption, and scalability, to gauge the system's ability to support a large number of participants without compromising performance.

9.2 Implementation

To fully demonstrate the practicality and versatility of our proposed decision-making system over blockchain, we chose to apply it to the context of blockchain development funding decisions[269] as a prototype. By implementing all the required functions in the pre-voting epoch, voting epoch, and post-voting epoch, as presented in Figure 4.1, we showcase the system's capabilities for transparent, secure, and decentralised decision-making. Furthermore, our proposed decision-making system is not limited to a specific domain; it can be adapted and utilised for distributed decision-making in various scenarios, including medical decision-making, IoT applications, financial services, and more.

In the course of implementing our system, a major challenge we tackled was achieving seamless interoperability between our proposed decentralised decision-making system and the broader blockchain ecosystem. Unlike traditional interoperability solutions, which often rely on bridge nodes to connect disparate blockchain systems and synchronise their states through mechanisms like multi-signature voting, our approach fundamentally rethinks interoperability.

Our architecture is designed to integrate directly with existing blockchain infrastructures without the need for intermediary protocols or bridge nodes. This direct integration is facilitated by the use of universal cryptographic standards and adapting our protocols to be compatible with multiple blockchain platforms. By doing so, we enable secure, and efficient cross-chain interactions which are essential for the broad applicability of decentralised decision-making systems.

In the pre-voting epoch, we successfully implemented crucial procedures such as proposal registration, external expert registration, and the registration of experts and voters with locked stakes using unspent transaction outputs (UTXOs) on the blockchain. Additionally, we implemented innovation management for experts to generate proposal lists, enhancing the expertise-driven nature of the system. All the information and metadata provided by experts, voters, and other participants are recorded on the blockchain, ensuring transparency and validation for everyone.

Furthermore, we developed the selection mechanism for voting committee members using verifiable random functions, along with the proposed Distributed Batched Key Generation (DBKG) protocol from Chapter 5 and its associated NIZK proofs. Our implementation of the DBKG protocol enables voting committee members to generate distributed key pairs, which are used to encrypt voters' and experts' ballots during the voting epoch. This threshold encryption ensures the privacy and security of ballots. Additionally, we implemented the evolving protocol from Chapter 7 with its corresponding NIZK proofs, allowing for a seamless transition of voting committee members during the Distributed Key Generation (DKG) process. Both the DBKG protocol and the evolving protocol support fault tolerance, ensuring the security of the system even in the presence of a minority of dishonest participants. Moreover, the behaviours of voting committee members in these protocols are

recorded online, enabling the detection and banning of any malicious activities.

In the voting epoch, we successfully implemented the proposed Two Stage Voting scheme from Chapter 6, including the Preferential Voting protocol and the Threshold Voting protocol. In both protocols, we incorporated delegation voting, allowing experts and voters to cast their ballots with Lifted Elgamal encryption, along with ballot validation NIZK proofs. All the encrypted ballots are submitted on the blockchain, and any participant submitting invalid ballots gets banned from further participation in the decision-making process, thereby safeguarding the integrity of the voting phase. Moreover, during the tally phase of both protocols, all ballots are thoroughly checked by the voting committee, ensuring that only valid ballots are included in the final tally results.

In the post-voting epoch, we implemented the penalty and reward mechanisms for voters, experts, and voting committee members, reinforcing the importance of honesty and responsible behaviour throughout the decision-making process. Additionally, project owners whose proposals are included in the final winning list receive funds on the blockchain, delivered to the address they provided during the pre-voting epoch.

By accomplishing these implementations, we have demonstrated the practicality and effectiveness of our proposed decision-making system. The successful application of the prototype in the blockchain development funding context illustrates its potential in various real-world scenarios. Our decision-making system fosters transparent and accountable governance, enabling participants to make collective and informed decisions securely and efficiently. In the next chapter, we conclude our work by summarising the key contributions and findings and discussing potential future directions for further improvement and expansion of the system.

9.3 Evaluation

To assess the realistic performance of our proposed decision-making system and its cryptographic protocols, we conducted a series of comprehensive tests. These tests were specifically designed to evaluate the efficiency of the cryptographic operations involved in our system. The configuration of the workstation used for testing comprised an Intel Core i7-6500U CPU running at 2.50GHz, 16GB RAM, and Linux Ubuntu 16.04 64-bit operating system. The cryptographic library used in our tests was implemented in Scala version 2.12.3, with the OpenJDK Runtime Environment (build 1.8.0_131-8u131-b11-2ubuntu1.16.04.3-b11) as the Java runtime environment. Additionally, we employed the org.bouncycastle library version 1.58 as the Elliptic Curve Math Library, and the parameters of the elliptic curve were set to Secp256k1, commonly used in blockchain applications.

The tests were conducted with a specific focus on cryptographic operations, as these operations are crucial in maintaining the security and privacy of the proposed decision-making system. By evaluating the performance of these operations, we gained insights into the system's efficiency and potential scalability in real-world scenarios. The test suite covered a range of cryptographic protocols, including key generation, encryption, decryption, and cryptographic proofs used in our proposed protocols.

To ensure the validity of our performance evaluations, we conducted multiple iterations of each test and recorded the average execution time and resource utilisation. These benchmarks were instrumental in understanding how the cryptographic operations performed under different scenarios and loads. By analysing the results, we could identify any potential bottlenecks or areas that might require optimisation.

The choice of the Scala programming language and the Bouncy Castle library for cryptographic operations enabled us to implement our cryptographic protocols efficiently and effectively. Furthermore, the usage of Secp256k1 as the elliptic curve parameter aligned with industry standards and blockchain applications, enhancing the compatibility and trustworthiness of our tests.

The performance evaluations and benchmarks provided valuable insights into the feasibility and practicality of our proposed decision-making system. By using real-world configurations and conducting rigorous testing, we aimed to ensure the reliability and robustness of our cryptographic protocols. In the following sections, we present the detailed results of these performance evaluations and discuss their implications for the overall system design and implementation. Additionally, we provide a comprehensive analysis of the cryptographic protocols' efficiency and scalability, paving the way for further optimisations and improvements to our decision-making system.

During the tests, we placed special emphasis on studying the performance of our decision-making system and its cryptographic protocols in various key aspects, including:

- Distributed Batched Key Generation (DBKG) protocol in Chapter 5:

– During the performance testing of the DBKG protocol, we evaluated its efficiency and resilience under various scenarios. We tested the protocol with a varying number of participation nodes, ranging from a few participants to as many as 100. For each case, we examined the protocol's behaviour when the corruption rate was set from 0 to $50\% - 1$, representing different levels of malicious behaviour within the system.

  In the DBKG protocol, the number of generated key pairs is set to half of the total number of participants in the Distributed Key Generation (DKG) protocol, which ensures a secure distribution of key pairs among the committee members. This design allows for a balance between efficiency and security, as generating a key pair for each participant could become computationally expensive and impractical for large-scale systems.

  By conducting these tests, we gained insights into the scalability of the DBKG protocol and its ability to handle different numbers of participants and corruption rates. The performance data provided valuable information about the protocol's resource consumption, execution times, and its ability to withstand malicious attempts to compromise the key generation process.

  The results of these tests allowed us to verify the effectiveness and efficiency of the DBKG protocol in generating secure and distributed key pairs for the decision-making system. It demonstrated the protocol's capability to support a large number of participants while maintaining the integrity and security of the key generation process, even in the presence of malicious actors.

– During our evaluation of the Lifted Elgamal encryption in the protocol, we performed tests using different segment sizes, such as 8-bit and 16-bit segments. We found that using smaller segment sizes, such as 8-bit, minimized the overall time of protocol execution. This is because DLOG bruteforcing during ciphertext decryption takes significant time for larger segments. For instance, with 5 members in the protocol, the overall time was 29 seconds for 16-bit segments, while it reduced to 0.5 seconds for 8-bit segments. Additionally, using larger segments decreased overall traffic about linearly. For the case with 5 members, the overall traffic was 32 KB for 16-bit segments instead of 61 KB for 8-bit segments.

  In our evaluation, we also considered the communication cost (overall traffic, KB) for all participants and the computation cost (seconds) for each participant in the protocol to generate one key pair. The results were compared with Gennaro et al.'s DKG protocol used in other decision-making systems, such as [13].

  When there were 10 participants, our proposed DBKG protocol could generate 5 key pairs, and the execution time for one participant, even with corruption, averaged only 0.04 seconds. Meanwhile, the communication cost for one key

pair was around 17 KB. We observed that our DBKG protocol demonstrated a linear growth in execution time and overall traffic, outperforming Gennaro et al.'s DKG protocol, which exhibited a square growth.

For larger participant sizes, such as 100, our DBKG protocol still excelled, with an execution time of less than 2 seconds even with minority dishonesty. The average communication cost for one key pair was around 169 KB. These results demonstrated the efficiency and scalability of our proposed DBKG protocol, showcasing its ability to handle a large number of participants while maintaining low computation and communication costs.

Overall, the evaluation of the Lifted Elgamal encryption and the DBKG protocol provided essential insights into their performance characteristics, enabling us to make informed decisions and optimizations for the practical implementation of the decision-making system on the blockchain. The results also highlighted the advantages of our proposed protocol over existing solutions, reinforcing its suitability for real-world deployment.

– During our evaluation of the NIZK proofs used in the DBKG protocol, specifically the Correct Sharing protocol and Correct Decryption protocol described in Section 5.5, we analyzed both the proof size and the execution time for the prover and verifier.

  For the Correct Sharing NIZK proof, we found that the proof size was small, which is crucial for efficiency in blockchain applications where storage space is limited. For the Correct Decryption NIZK proof, which is also used in the DBKG protocol and the Evolving protocol described in Section 7.2, we found that the proof size was only 102 bytes. This small proof size is advantageous as it reduces the overhead in communication and storage on the blockchain. Additionally, the prover's running time for this proof was 0.889 milliseconds, and the verifier's running time was 0.924 milliseconds.

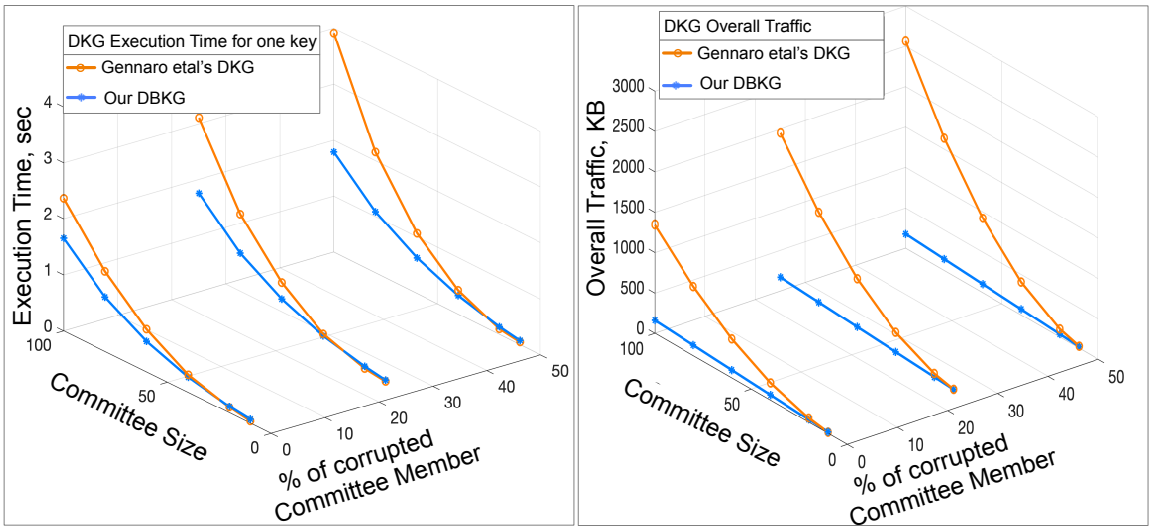

Figure 9.1: Execution Time and Overall Traffic of DKG protocols: the proposed DBKG protocol in Chapter 5 v.s. Gennaro *et al.*'s DKG [38].

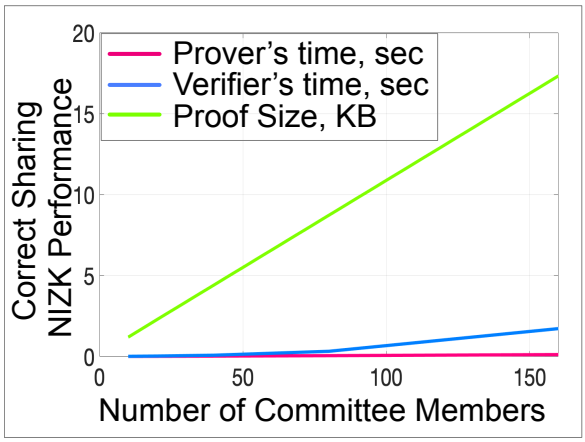

Figure 9.2: The prover's running time, verifier's running time and the proof size for Correct Sharing NIZK proof.

- Two Stage Voting protocols in Chapter 6:
  - During our evaluation of the voting stages in the proposed decision-making system in Figure 9.3, we focused on the ballot creation time and size for both voters and experts in the Preferential Voting Stage. The size of the shortlist, which represents the number of projects being considered for voting, was varied from 4 to 16 during the tests.

In the Threshold Voting Stage, we observed that each voter and expert spent less than 1 second to create their respective ballots. This minimal time overhead ensures that the overall voting performance is not significantly affected by the ballot creation process. Additionally, with 5000 voters and 50 experts participating, the overall communication overhead in the Threshold Voting Stage was approximately 20 MB per project.

It is important to note that in practical scenarios, one decision-making period is sufficiently long (*e.g.*, around 30 days or approximately 4320 blocks for the Bitcoin blockchain). As a result, the blockchain space overhead for deploying our proposed decision-making system on a cryptocurrency blockchain is negligible. The low space requirement ensures that the decision-making process can be effectively conducted on the blockchain without significantly impacting other transactions and data stored on the blockchain.

– In our evaluation of the NIZK proofs used in the Preferential Voting protocol and the Threshold Voting protocol, we examined the Batched 0 or 1 NIZK proof and the Unit Vector NIZK proof.

  For the Batched 0 or 1 NIZK proof, as shown in Figure 9.4, we analysed the proof size and execution time for both the prover and the verifier. The results demonstrated that the proof size was relatively small, and the execution times for both the prover and the verifier were quite fast. This efficiency is crucial for maintaining the overall performance of the decision-making system, especially in scenarios with a large number of participants and projects.

  In the case of the Unit Vector NIZK proof, used in the ballot casting process during the two-stage voting, we found that the prover's execution time and verifier's execution time was less than 0.5 seconds. Additionally, the proof size was less than 2.5 KB, further highlighting the efficiency of the NIZK proof used in our proposed voting protocols.

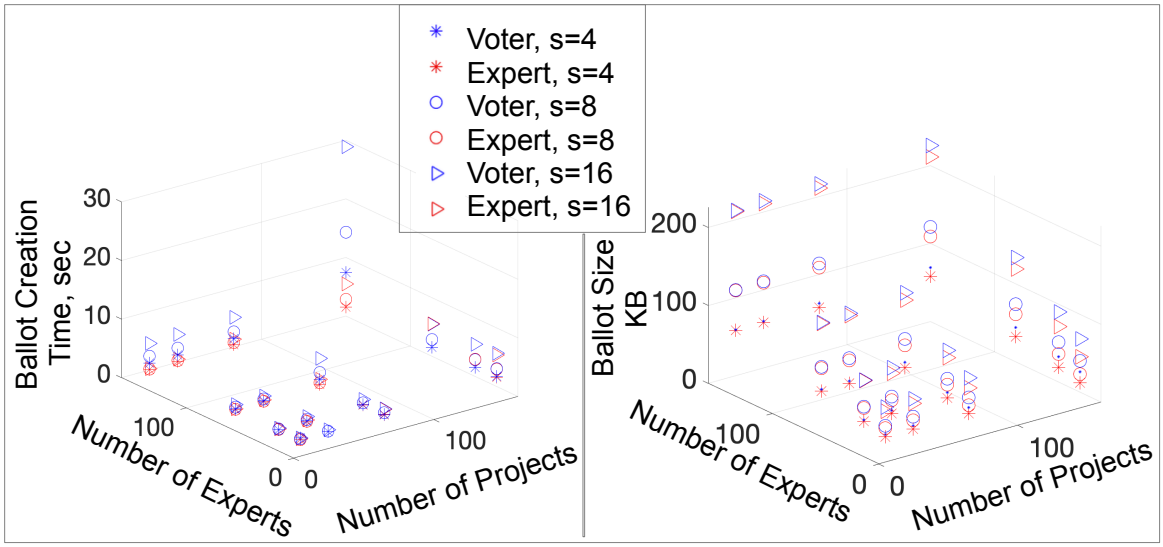

Figure 9.3: Ballot size and creation time for each voter and expert in Preferential Voting Stage.

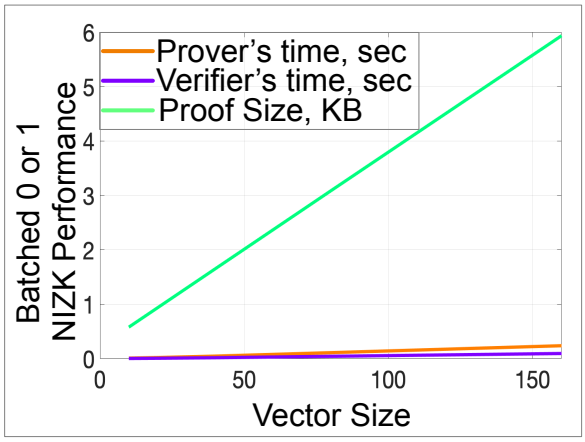

Figure 9.4: The prover's running time, verifier's running time and the proof size for Batched 0 or 1 NIZK proof.

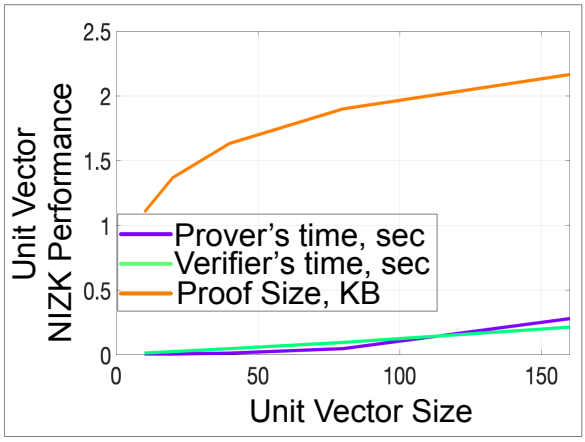

Figure 9.5: The prover's running time, verifier's running time and the proof size for Unit Vector NIZK proof.

- Evolving protocol in Chapter 7: In our evaluation of the Handover protocol, we analysed the execution time and overall traffic for different numbers of committee members (n) and different numbers of corrupted parties. The segment size used in the evaluation was set to 32 bits.

  As shown in Figure 9.6, we observed the performance of the Handover protocol under various scenarios. The results indicate that the execution time of the protocol increases with the number of committee members and the number of corrupted parties. This is expected, as more committee members or more corruption introduce additional computational overhead and communication requirements.

  However, it is worth noting that even with a larger number of committee members (up to 100) and a significant number of corrupted parties (up to $n/2 - 1$), the Handover protocol still performs efficiently. The execution time remains within a reasonable range, and the overall traffic generated during the protocol execution is manageable. This demonstrates the robustness of the Handover protocol and its ability to handle a diverse range of decision-making scenarios.

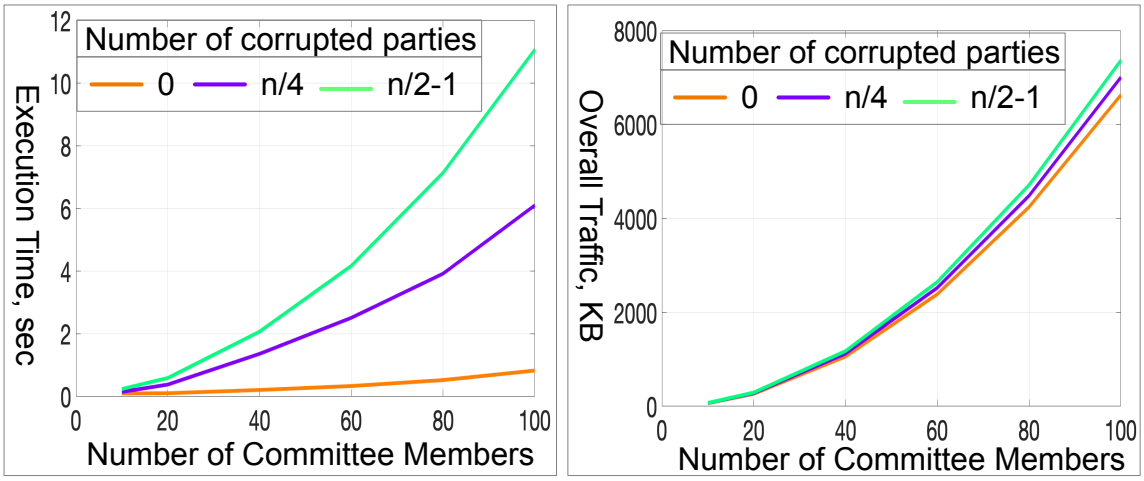

Figure 9.6: Handover Execution Time and Overall Traffic.

9.4 Summary

In this chapter, we provided a comprehensive overview of the implementation of the prototype for our proposed decision-making system and conducted a thorough evaluation of the performance of its main protocols. The prototype was developed to demonstrate the practicality and effectiveness of our proposed system. We chose blockchain development funding decision-making as an example to showcase the functionalities of the system, but it can be applied to various other scenarios such as medical decision-making, IoT applications, financial services, and more.

In the pre-voting epoch, we successfully implemented all the required functions, including proposal registration, registration of external experts, registration of experts and voters with locked stakes, and innovation management. All relevant information, such as proposal metadata, voter and expert information, and proposal list, was recorded on the blockchain to ensure transparency and validation.

The Distributed Batched Key Generation (DBKG) protocol, which enables voting committee members to generate distributed key pairs for encrypting voters' and experts' ballots, was also implemented. Additionally, we successfully integrated the evolving protocol to change voting committee members during the Distributed Key Generation process. Both protocols demonstrated fault tolerance, security, and the ability to detect and ban malicious behaviors.

Furthermore, the Two Stage Voting scheme, consisting of the Preferential Voting protocol and Threshold Voting protocol, was successfully implemented. We integrated delegation voting and ballot casting using Lifted Elgamal encryption with non-interactive zero-knowledge (NIZK) proofs. Invalid ballots were detected and removed during the tally phase, ensuring the integrity of the voting process.

Throughout our evaluation, we assessed the performance of various protocols in terms of execution time, communication cost, and scalability. We observed that the system could efficiently handle a large number of participants, demonstrating its practicality and robustness. Our decision-making system achieved low communication overhead and reasonable execution times, even in scenarios with a large number of participants and potential malicious behaviour.

In conclusion, this chapter highlighted the successful implementation and evaluation of the prototype for our proposed decision-making system. The results obtained from our performance evaluation further validate the efficiency and effectiveness of the system's protocols. In the next chapter, we conclude our work by summarising the key contributions and findings and discussing potential future directions for further improvement and expansion of the system.

Chapter 10

Conclusion and Future Works

> As our eyes grow accustomed to sight,
> they armour themselves against wonder.
>
> Leonard Cohen

10.1 Overview

In this thesis, we have presented a novel and comprehensive approach to decentralised decision-making on blockchain systems. Our work addresses the challenges of scalability, dynamic participation, and fair representation in the decision-making process. We have designed and implemented a fully functional prototype of our proposed decision-making system, which leverages cryptographic protocols and blockchain technology to enable secure and efficient voting and proposal evaluation. Through extensive evaluations, we have demonstrated the practicality, efficiency, and effectiveness of our system in real-world scenarios.

In this chapter, we provide a review of the aims of this thesis and a summary of the contributions. Then we summarise the discussion and implications from previous chapters, presenting next steps aiming at further advancing the research in decentralised decision-making systems on blockchain.

10.2 Contributions

The primary goal of this thesis was to design and develop a decentralised decision-making system on the blockchain that addresses the challenges of scalability, dynamic participation, and fair representation. To achieve this aim, we proposed a novel systemic design for decision-making, encompassing three main phases: pre-voting, voting, and post-voting epochs. We introduced an evolving committee mechanism to adapt to changing participation and enhance fault tolerance and security. Additionally, we developed a reputation management scheme to objectively measure participants' contributions and behaviours across different roles and fields.

The contributions of this thesis can be summarised as follows:

1. **Systemic Design:** We proposed a comprehensive systemic design for decentralised decision-making on the blockchain, covering proposal registration, voting, and rewards distribution, to ensure a fair and efficient decision-making process.

2. **Evolving Committee Mechanism:** We introduced a mechanism to enable the voting committee to evolve during the distributed key generation process, improving adaptability and security in the face of changing participation and potential malicious behaviours.

3. **Reputation Management Scheme:** We designed a reputation management scheme that objectively evaluates participants' contributions and behaviours, encouraging diverse and active engagement in the decision-making ecosystem.

4. **Efficient Cryptographic Protocols:** We developed efficient cryptographic protocols, such as the Distributed Batched Key Generation (DBKG) protocol and Two-Stage Voting scheme, to ensure privacy, security, and verifiability while minimising communication overhead and execution times.

5. **Prototype Implementation and Evaluation:** We created a fully functional prototype of the proposed decision-making system and conducted extensive evaluations to demonstrate its practicality, efficiency, and scalability in real-world scenarios.

10.3 Future Works

Although this thesis has made great progress towards distributed decision-making systems on the blockchain, various exciting future directions for more study and development abound:

- **Coercion-Resistance and Forgiveness:** Coercion-resistance ensures that coerced voters cannot be identified from those voting voluntarily. Future studies can look at other ways to acquire this quality or tackle lesser ideas as forgiveness. Forgiveness gives forced voters a chance to later on change their votes after casting them, therefore reflecting their actual sentiments.

- **Self-tallying Voting:** Self-tallying voting is a fascinating path for more investigation on decentralised systems of decision-making. This strategy can improve the system's openness and verifiability by enabling each voter to independently calculate the voting outcomes. Self-tallying systems let voters confirm the accuracy of the last count without depending on a central government or voting body. Since voters can independently confirm their own votes and help to total the system, this function not only increases its dependability but also lowers the communication overhead. Investigating and putting self-tallying mechanisms into use in our suggested system will help to improve decentralisation and democratise the decision-making process.

- **Adversary Models:** In this thesis, we considered a stationary corruption adversary model, where the adversary chooses which parties to corrupt before the protocol's execution. This will provide fresh insights on scalability and efficiency. Future studies could investigate various adversary models, such transient (mobile) corruption, whereby participants could be infected and uncorrupted dynamically and the adversary's activities can vary during the procedure. Another fascinating topic to investigate is proactive security, in which the system acts to minimise the influence of possible enemies.

- **Communication Modes:** We examined a synchronous communication paradigm for this thesis. Future studies could concentrate on creating asynchronous versions of the cryptographic systems to guarantee their security and efficiency in situations including different communication delays.

- **Scalability and Optimisation:** As decentralised decision-making systems grow to accommodate more participants and proposals, scalability becomes a critical concern. Future research could explore optimisation techniques for cryptographic protocols and algorithms to reduce communication and computation overhead, enabling the system to scale more efficiently.

- **Privacy and Confidentiality Enhancements:** Future studies could investigate methods of optimising algorithms and cryptographic protocols to lower communication

and computing overhead, therefore allowing the system to scale more effectively. Although our cryptographic systems offer a fair degree of privacy and confidentiality, more research on advanced privacy-enhancing technologies including zero-knowledge proofs and secure multi-party computation will help to strengthen the anonymity and confidence of the decision-making process

- **Governance Models:** System dynamics and decision-making outcomes can be much changed by different governance models. To find their effects on decision outcomes and community dynamics, future studies could investigate various governance models as futarchy or quadratic voting.

- **Enhancing Privacy in Reputation-Based Systems:** : Future work can focus on the integration of advanced privacy-preserving technologies to safeguard participant data as we continue to refine our reputation management framework. In particular, we can investigate the use of differential privacy to obscure reputation scores, implement Zero-Knowledge Proofs (ZKPs) to validate qualifications without disclosing confidential information, and employ decentralised architectures to distribute the computation of reputation scores. These initiatives will be complemented by rigorous access controls and enhanced encryption protocols, which will provide comprehensive protection against inference assaults and unauthorised data access. Furthermore, in order to enhance the system's accountability and privacy, the integration of decentralised identity technologies will be examined. The collective objective of these developments is to create a secure, private, and efficient environment for decentralised decision-making.

In conclusion, this thesis has contributed to the advancement of decentralised decision-making systems on the blockchain by proposing a systemic design, efficient cryptographic protocols, and a reputation management scheme. Through a fully functional prototype and extensive evaluations, we have demonstrated the practicality, efficiency, and scalability of our approach. Future research can further enhance and expand upon these contributions to make decentralised decision-making a reality across various domains and applications.

Appendix A

Honest Majority Analysis Results

A.1 Adversary's Corruption Probability

Table A.1: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 10$.

$\mathbb{P}_m$ \ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.651322	0.892626	0.971752	0.993953	0.997467
0.15	0.263901	0.62419	0.850692	0.953643	0.976743
0.2	0.263901	0.62419	0.850692	0.953643	0.976743
0.25	0.070191	0.3222	0.617217	0.83271	0.90044
0.3	0.070191	0.3222	0.617217	0.83271	0.90044
0.35	0.012795	0.120874	0.350389	0.617719	0.733962
0.4	0.012795	0.120874	0.350389	0.617719	0.733962
0.45	0.001635	0.032793	0.150268	0.366897	0.495595
0.5	0.001635	0.032793	0.150268	0.366897	0.495595

Table A.2: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 20$.

$\mathbb{P}_m$ \ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.608253	0.930825	0.992363	0.999476	0.999889
0.15	0.323073	0.793915	0.964517	0.996389	0.999073
0.2	0.132953	0.588551	0.892913	0.984039	0.995067
0.25	0.043174	0.370352	0.762492	0.949048	0.981137
0.3	0.011253	0.195792	0.583629	0.874401	0.944666
0.35	0.002386	0.086693	0.39199	0.749989	0.870066
0.4	0.000416	0.032143	0.227728	0.584107	0.747994
0.45	0.00006	0.009982	0.113331	0.404401	0.585694
0.5	0.000007	0.002595	0.047962	0.244663	0.408639

Table A.3: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 30$.

$\mathbb{P}_m$ \ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.588649	0.955821	0.997887	0.999953	0.999995
0.15	0.175495	0.744767	0.969845	0.99849	0.999759
0.2	0.07319	0.572488	0.923405	0.994341	0.99891
0.25	0.007784	0.239209	0.718623	0.956476	0.987895
0.3	0.00202	0.128651	0.568482	0.905989	0.968794
0.35	0.000089	0.025616	0.26963	0.708528	0.864955
0.4	0.000015	0.009493	0.159322	0.56891	0.767313
0.45	0	0.000902	0.040053	0.285496	0.497524
0.5	0	0.000231	0.016937	0.175369	0.355156

Table A.4: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 40$.

$\mathbb{P}_m$ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.576869	0.971538	0.999402	0.999996	1
0.15	0.206273	0.838671	0.991382	0.999856	0.999988
0.2	0.041902	0.562854	0.944717	0.997947	0.999752
0.25	0.005063	0.268223	0.804075	0.984427	0.997267
0.3	0.000381	0.087505	0.559393	0.929051	0.98211
0.35	0.000018	0.019407	0.296751	0.78884	0.924945
0.4	0.000001	0.002936	0.115147	0.55978	0.785786
0.45	0	0.000304	0.031951	0.311481	0.560938
0.5	0	0.000022	0.006255	0.129766	0.315586

Table A.5: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 50$.

$\mathbb{P}_m$ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.568802	0.981504	0.999828	1	1
0.15	0.122145	0.80959	0.992736	0.999939	0.999997
0.2	0.024538	0.55626	0.959768	0.999243	0.999943
0.25	0.001005	0.186057	0.777134	0.986749	0.998231
0.3	0.000074	0.060722	0.553168	0.946045	0.989616
0.35	0.000001	0.006261	0.217807	0.763124	0.92347
0.4	0	0.000932	0.084803	0.553524	0.802632
0.45	0	0.00003	0.012276	0.233983	0.498093
0.5	0	0.000002	0.00237	0.097807	0.283961

Table A.6: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 60$.

$\mathbb{P}_m$ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.562834	0.987894	0.99995	1	1
0.15	0.141636	0.873208	0.997792	0.999994	1
0.2	0.014585	0.551383	0.970525	0.999717	0.999987
0.25	0.000666	0.206542	0.837892	0.994989	0.999577
0.3	0.000015	0.042697	0.548564	0.958712	0.993917
0.35	0	0.004826	0.237783	0.82143	0.955388
0.4	0	0.000301	0.063238	0.548894	0.817859
0.45	0	0.000011	0.009961	0.253565	0.549849
0.5	0	0	0.000913	0.074624	0.257606

Table A.7: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 70$.

$\mathbb{P}_m$ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.55819	0.99203	0.999985	1	1
0.15	0.087269	0.853211	0.99814	0.999997	1
0.2	0.008761	0.547587	0.978288	0.999894	0.999997
0.25	0.000138	0.148083	0.81858	0.995732	0.999724
0.3	0.000003	0.030308	0.54498	0.968244	0.996411
0.35	0	0.001617	0.17985	0.80273	0.954781
0.4	0	0.000099	0.04758	0.545289	0.83162
0.45	0	0.000001	0.003968	0.196024	0.498392
0.5	0	0	0.000356	0.057434	0.235063

Table A.8: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 80$.

$\mathbb{P}_m$ \ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.554444	0.994728	0.999996	1	1
0.15	0.100439	0.899402	0.999416	1	1
0.2	0.005304	0.544525	0.983935	0.99996	0.999999
0.25	0.000092	0.163415	0.864777	0.998338	0.999932
0.3	0.000001	0.021668	0.542087	0.975474	0.99787
0.35	0	0.001257	0.195358	0.847911	0.972947
0.4	0	0.000033	0.036041	0.542379	0.844089
0.45	0	0	0.003247	0.211489	0.543211
0.5	0	0	0.00014	0.044497	0.215438

Table A.9: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 90$.

$\mathbb{P}_m$ \ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.551338	0.996498	0.999999	1	1
0.15	0.063366	0.884789	0.999508	1	1
0.2	0.003232	0.541987	0.98807	0.999985	1
0.25	0.000019	0.119508	0.850046	0.998583	0.999956
0.3	0	0.015577	0.53969	0.980992	0.99873
0.35	0	0.00043	0.150457	0.833421	0.972657
0.4	0	0.000011	0.027445	0.539967	0.855419
0.45	0	0	0.00132	0.166345	0.498584
0.5	0	0	0.000055	0.034653	0.198134

Table A.10: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 100$.

$\mathbb{P}_m$ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.54871	0.997666	1	1	1
0.15	0.072573	0.919556	0.999843	1	1
0.2	0.001979	0.539839	0.991113	0.999994	1
0.25	0.000013	0.131353	0.88643	0.999438	0.999989
0.3	0	0.011249	0.53766	0.985225	0.99924
0.35	0	0.000336	0.162858	0.869663	0.983367
0.4	0	0.000004	0.020989	0.537925	0.865746
0.45	0	0	0.001086	0.178902	0.538671
0.5	0	0	0.000022	0.027099	0.182728

Table A.11: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 300$.

$\mathbb{P}_m$ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.528142	0.999999	1	1	1
0.15	0.004055	0.989395	1	1	1
0.2	0	0.523022	0.999967	1	1
0.25	0	0.020198	0.976115	1	1
0.3	0	0.000025	0.521768	0.999872	1
0.35	0	0	0.035212	0.96692	0.999824
0.4	0	0	0.000143	0.521926	0.96441
0.45	0	0	0	0.044346	0.522361
0.5	0	0	0	0.000286	0.04647

Table A.12: The probability that adversary corrupts at least R_{mv} of the n voting committee members if it takes over R_{ms} of the whole stakes when $n = 500$.

$\mathbb{P}_m$ \ R_{ms} / R_{mv}	0.1	0.2	0.3	0.4	0.45
0.1	0.521802	1	1	1	1
0.15	0.00028	0.998373	1	1	1
0.2	0	0.517836	1	1	1
0.25	0	0.003742	0.994237	1	1
0.3	0	0	0.516865	0.999999	1
0.35	0	0	0.009058	0.990444	0.999998
0.4	0	0	0.000001	0.516988	0.989287
0.45	0	0	0	0.013006	0.517326
0.5	0	0	0	0.000004	0.013972

A.2 Honest Committee's Probability

Table A.13: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 10$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	0.972608	0.987705	0.99841	0.999922	1
0.4	0.898005	0.945238	0.989408	0.999136	0.999991
0.45	0.898005	0.945238	0.989408	0.999136	0.999991
0.5	0.738437	0.833761	0.952651	0.993631	0.999853
0.6	0.504405	0.633103	0.849732	0.967207	0.998365
0.7	0.266038	0.382281	0.649611	0.879126	0.987205
0.8	0.09956	0.16729	0.382783	0.6778	0.929809
0.9	0.023257	0.046357	0.149308	0.37581	0.736099
0.95	0.023257	0.046357	0.149308	0.37581	0.736099

Table A.14: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 20$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	0.978586	0.993534	0.999739	0.999998	1
0.4	0.941966	0.978971	0.998721	0.999985	1
0.45	0.869235	0.943474	0.994862	0.999898	1
0.5	0.750711	0.872479	0.982855	0.999437	0.999999
0.6	0.414306	0.595599	0.886669	0.990018	0.99994
0.7	0.129934	0.250011	0.60801	0.913307	0.997614
0.8	0.018863	0.050952	0.237508	0.629648	0.956826
0.9	0.000927	0.003611	0.035483	0.206085	0.676927
0.95	0.000111	0.000524	0.007637	0.069175	0.391747

Table A.15: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 30$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	0.994988	0.999144	0.999993	1	1
0.4	0.966556	0.991698	0.999838	1	1
0.45	0.928611	0.97876	0.999374	0.999998	1
0.5	0.769091	0.902943	0.99363	0.999948	1
0.6	0.359178	0.578466	0.91553	0.996889	0.999998
0.7	0.069407	0.176286	0.588809	0.938913	0.999546
0.8	0.003985	0.017183	0.159523	0.60697	0.974173
0.9	0.000041	0.000313	0.009317	0.122711	0.647439
0.95	0.000005	0.000047	0.002113	0.044179	0.411351

Table A.16: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 40$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	0.996572	0.999598	0.999999	1	1
0.4	0.980422	0.996649	0.999979	1	1
0.45	0.923318	0.981089	0.999725	1	1
0.5	0.786956	0.925648	0.997581	0.999995	1
0.6	0.318548	0.568132	0.936687	0.999009	1
0.7	0.038585	0.12851	0.577181	0.956758	0.999912
0.8	0.000883	0.006065	0.111009	0.593127	0.984505
0.9	0.000002	0.000029	0.002561	0.075914	0.629018
0.95	0	0.000001	0.000103	0.007942	0.222808

Table A.17: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 50$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	0.999131	0.999942	1	1	1
0.4	0.988402	0.998626	0.999997	1	1
0.45	0.955621	0.992383	0.999964	1	1
0.5	0.803369	0.942656	0.999067	1	1
0.6	0.28617	0.561035	0.952236	0.999679	1
0.7	0.021951	0.095502	0.569178	0.969197	0.999983
0.8	0.000201	0.002197	0.078851	0.583559	0.990645
0.9	0	0.000003	0.000723	0.048027	0.616123
0.95	0	0	0.000032	0.005656	0.250294

Table A.18: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 60$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	0.999414	0.999973	1	1	1
0.4	0.993067	0.99943	1	1	1
0.45	0.953822	0.993363	0.999984	1	1
0.5	0.818271	0.95552	0.999636	1	1
0.6	0.259248	0.555776	0.963762	0.999895	1
0.7	0.012678	0.071879	0.56324	0.977932	0.999996
0.8	0.000047	0.00081	0.056771	0.57644	0.994319
0.9	0	0	0.000208	0.030837	0.606451
0.95	0	0	0.000002	0.001013	0.137399

Table A.19: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 70$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	0.999846	0.999996	1	1	1
0.4	0.995826	0.999762	1	1	1
0.45	0.972406	0.997229	0.999998	1	1
0.5	0.831782	0.965334	0.999857	1	1
0.6	0.236264	0.551677	0.972377	0.999965	1
0.7	0.007402	0.054592	0.558608	0.984113	0.999999
0.8	0.000011	0.000302	0.04127	0.570877	0.996532
0.9	0	0	0.00006	0.02001	0.598852
0.95	0	0	0.000001	0.000758	0.158794

Table A.20: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 80$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	0.999896	0.999998	1	1	1
0.4	0.997473	0.9999	1	1	1
0.45	0.971649	0.997603	0.999999	1	1
0.5	0.844052	0.972876	0.999943	1	1
0.6	0.216286	0.548367	0.978861	0.999988	1
0.7	0.004356	0.041747	0.554864	0.988516	1
0.8	0.000003	0.000114	0.03022	0.566375	0.997874
0.9	0	0	0.000018	0.013088	0.592676
0.95	0	0	0	0.000136	0.087971

Table A.21: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 90$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	0.999972	1	1	1	1
0.4	0.998463	0.999958	1	1	1
0.45	0.982715	0.998977	1	1	1
0.5	0.855224	0.978708	0.999977	1	1
0.6	0.198696	0.54562	0.983769	0.999996	1
0.7	0.00258	0.032096	0.551757	0.991671	1
0.8	0.000001	0.000043	0.022254	0.562634	0.998692
0.9	0	0	0.000005	0.008613	0.58753
0.95	0	0	0	0.000104	0.103246

Table A.22: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 100$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	0.999981	1	1	1	1
0.4	0.999062	0.999982	1	1	1
0.45	0.982359	0.999118	1	1	1
0.5	0.865424	0.983238	0.999991	1	1
0.6	0.183057	0.543294	0.987502	0.999999	1
0.7	0.001536	0.024783	0.549124	0.993941	1
0.8	0	0.000016	0.016463	0.559462	0.999192
0.9	0	0	0.000002	0.005696	0.583156
0.95	0	0	0	0.000019	0.057577

Table A.23: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 300$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	1	1	1	1	1
0.4	1	1	1	1	1
0.45	0.999791	1	1	1	1
0.5	0.963722	0.999815	1	1	1
0.6	0.045782	0.525049	0.999911	1	1
0.7	0	0.000205	0.528442	0.999986	1
0.8	0	0	0.000059	0.534476	1
0.9	0	0	0	0.000002	0.548419
0.95	0	0	0	0	0.001267

Table A.24: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 500$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	1	1	1	1	1
0.4	1	1	1	1	1
0.45	0.999997	1	1	1	1
0.5	0.988918	0.999998	1	1	1
0.6	0.013564	0.519411	0.999999	1	1
0.7	0	0.000002	0.522043	1	1
0.8	0	0	0	0.526728	1
0.9	0	0	0	0	0.537569
0.95	0	0	0	0	0.000035

Table A.25: The probability that at least R_{hv} of the n voting committee members are honest if R_{hs} of the whole stakes are honest when $n = 1000$.

$\mathbb{P}_h$ \ R_{hs} / R_{hv}	0.55	0.6	0.7	0.8	0.9
0.35	1	1	1	1	1
0.4	1	1	1	1	1
0.45	1	1	1	1	1
0.5	0.999319	1	1	1	1
0.6	0.000793	0.51373	1	1	1
0.7	0	0	0.515594	1	1
0.8	0	0	0	0.518911	1
0.9	0	0	0	0	0.526599
0.95	0	0	0	0	0

Bibliography

[1] Mohammad Syaiful Aris et al. “The Blockchain-Based on E-Voting in The Local Elections System: An Effort to Realize E-Democracy”. In: *Jurnal Pembaharuan Hukum* 10.1 (2023), pp. 27–42.

[2] Yizhi Liu et al. “A blockchain-based decentralized, fair and authenticated information sharing scheme in zero trust internet-of-things”. In: *IEEE Transactions on Computers* 72.2 (2022), pp. 501–512.

[3] Christoph Mueller-Bloch et al. “Understanding decentralization of decision-making power in proof-of-stake blockchains: an agent-based simulation approach”. In: *European Journal of Information Systems* (2022), pp. 1–20.

[4] Saban Adana et al. “Linking decentralization in decision-making to resilience outcomes: a supply chain orientation perspective”. In: *The International Journal of Logistics Management* (2023).

[5] Dylan Weiss, Jacob Wolmer, and Avimanyou Vatsa. “Blockchain-based Electronic Voting System for Modern Democracy: A Review”. In: *2022 IEEE Integrated STEM Education Conference (ISEC)*. IEEE. 2022, pp. 162–166.

[6] Michele Scarlato, Reza Tourani, and Moongu Jeon. “Sancus: an Anonymous and Trustworthy Blockchain-based Electronic Voting Architecture”. In: *2022 Fourth International Conference on Blockchain Computing and Applications (BCCA)*. IEEE. 2022, pp. 93–98.

[7] Sachi Chaudhary et al. “Blockchain-based Secure Voting Mechanism Underlying 5G Network: A Smart Contract Approach”. In: *IEEE Access* (2023).

[8] LM Goodman. “TezosÔÇöa self-amending crypto-ledger White paper”. In: *URL: https://www. tezos. com/static/papers/white paper. pdf* (2014).

[9] *Tezos*. `http://tezos.gitlab.io/`. Accessed: 2022-06-01. 2017.

[10] *Polkadot*. `https://wiki.polkadot.network/docs/learn-governance/`. Accessed: 2022-06-11. 2017.

[11] *Cardano Governance*. `https://cardano.org/governance`. Accessed: 2022-06-01. 2018.

[12] *Bitcoin Improvement Proposals*. `https://github.com/bitcoin/bips/`. Accessed: 2022-06-01. 2013.

[13] Bingsheng Zhang, Roman Oliynykov, and Hamed Balogun. "A Treasury System for Cryptocurrencies: Enabling Better Collaborative Intelligence". In: *The Network and Distributed System Security Symposium (NDSS '19)*. 2019.

[14] Xuechao Yang et al. "Blockchain voting: Publicly verifiable online voting protocol without trusted tallying authorities". In: *Future Generation Computer Systems* 112 (2020), pp. 859–874.

[15] Saad Khan et al. "Implementation of decentralized blockchain E-voting". In: *EAI Endorsed Transactions on Smart Cities* 4.10 (2020).

[16] Hamed Olanrewwaju Balogun. *Towards sustainable blockchains: cryptocurrency treasury and general decision-making systems with provably secure delegable blockchain-based voting*. Lancaster University (United Kingdom), 2021.

[17] D. Kaidalov, A. Nastenko, et al. *Dash Governance System: Analysis and Suggestions for Improvements*. `https://iohk.io/en/research`. Accessed: 2021-12-27.

[18] Fabrice Benhamouda et al. "Can a Public Blockchain Keep a Secret?" In: *Theory of Cryptography Conference*. Springer. 2020, pp. 260–290.

[19] Harshitha U Kumar and Raghavendra Prasad SG. "Algorand: A Better Distributed Ledger". In: *2019 1st International Conference on Advances in Information Technology (ICAIT)*. IEEE. 2019, pp. 496–499.

[20] Ran Canetti. "Universally Composable Security: A New Paradigm for Cryptographic Protocols". In: *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*. IEEE Computer Society, 2001, pp. 136–145.

[21] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020.

[22] Manuel Blum. "Coin Flipping by Telephone". In: *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA, August 24-26, 1981*. Ed. by Allen Gersho. U. C. Santa Barbara, Dept. of Elec. and Computer Eng., ECE Report No 82-04, 1981, pp. 11–15.

[23] Adi Shamir. "How to Share a Secret". In: *Commun. ACM* 22.11 (1979), pp. 612–613.

[24] Benny Chor et al. "Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract)". In: *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*. IEEE Computer Society, 1985, pp. 383–395.

[25] Zuzana Beerliová-Trubíniová and Martin Hirt. "Perfectly-Secure MPC with Linear Communication Complexity". In: *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008*. Ed. by Ran Canetti. Vol. 4948. Lecture Notes in Computer Science. Springer, 2008, pp. 213–230.

[26] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract)". In: *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*. Ed. by Robert Sedgewick. ACM, 1985, pp. 291–304.

[27] Shafi Goldwasser and Silvio Micali. "Probabilistic Encryption". In: *J. Comput. Syst. Sci.* 28.2 (1984), pp. 270–299.

[28] Moni Naor and Moti Yung. "Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks". In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*. Ed. by Harriet Ortiz. ACM, 1990, pp. 427–437.

[29] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof Systems". In: *SIAM J. Comput.* 18.1 (1989), pp. 186–208.

[30] Boaz Barak et al. "Resettably-Sound Zero-Knowledge and its Applications". In: *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*. IEEE Computer Society, 2001, pp. 116–125.

[31] Roger M. Needham and Michael D. Schroeder. "Authentication Revisited". In: *ACM SIGOPS Oper. Syst. Rev.* 21.1 (1987), p. 7.

[32] William M. Springer II. "Review of Cryptography: theory and practice, second edition by Douglas R. Stinson. CRC Press". In: *SIGACT News* 34.4 (2003), pp. 22–25.

[33] Claude E. Shannon. "Communication theory of secrecy systems". In: *Bell Syst. Tech. J.* 28.4 (1949), pp. 656–715.

[34] Benoit Chevallier-Mames et al. "Secure delegation of elliptic-curve pairing". In: *Smart Card Research and Advanced Application: 9th IFIP WG 8.8/11.2 International Conference, CARDIS 2010, Passau, Germany, April 14-16, 2010. Proceedings 9*. Springer. 2010, pp. 24–35.

[35] Taher El Gamal. "A public key cryptosystem and a signature scheme based on discrete logarithms". In: *IEEE Trans. Inf. Theory* 31.4 (1985), pp. 469–472.

[36] Mihir Bellare et al. “Key-Privacy in Public-Key Encryption”. In: *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*. Ed. by Colin Boyd. Vol. 2248. Lecture Notes in Computer Science. Springer, 2001, pp. 566–582.

[37] Torben Pedersen and Bent Petersen. “Explaining gradually increasing resource commitment to a foreign market”. In: *International business review* 7.5 (1998), pp. 483–501.

[38] Rosario Gennaro et al. “Secure Distributed Key Generation for Discrete-Log Based Cryptosystems”. In: *J. Cryptol.* 20.1 (2007), pp. 51–83.

[39] George Robert Blakley. “Safeguarding cryptographic keys”. In: *Managing Requirements Knowledge, International Workshop on*. IEEE Computer Society. 1979, pp. 313–313.

[40] Ivan Damgård. *On Σ-Protocols*. `https://cs.au.dk/~ivan/Sigma.pdf`. Accessed: 2022-06-01. 2010.

[41] Jens Groth. “Linear Algebra with Sub-linear Zero-Knowledge Arguments”. In: *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*. Ed. by Shai Halevi. Vol. 5677. Lecture Notes in Computer Science. Springer, 2009, pp. 192–208.

[42] Jens Groth. “A Verifiable Secret Shuffle of Homomorphic Encryptions”. In: *J. Cryptol.* 23.4 (2010), pp. 546–579.

[43] Stephanie Bayer and Jens Groth. “Efficient Zero-Knowledge Argument for Correctness of a Shuffle”. In: *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 263–280.

[44] Amos Fiat and Adi Shamir. “How to Prove Yourself: Practical Solutions to Identification and Signature Problems”. In: *Advances in Cryptology - CRYPTO ’86, Santa Barbara, California, USA, 1986, Proceedings*. Ed. by Andrew M. Odlyzko. Vol. 263. Lecture Notes in Computer Science. Springer, 1986, pp. 186–194.

[45] Charles Rackoff and Daniel R. Simon. “Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack”. In: *Advances in Cryptology - CRYPTO ’91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*. Ed. by Joan Feigenbaum. Vol. 576. Lecture Notes in Computer Science. Springer, 1991, pp. 433–444.

[46] J. T. Schwartz. "Fast Probabilistic Algorithms for Verification of Polynomial Identities". In: *J. ACM* 27.4 (Oct. 1980), pp. 701–717. ISSN: 0004-5411.

[47] Raymond Cheng et al. "Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contracts". In: *IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019*. IEEE, 2019, pp. 185–200.

[48] Rosario Gennaro et al. "Secure Distributed Key Generation for Discrete-Log Based Cryptosystems". In: *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*. Ed. by Jacques Stern. Vol. 1592. Lecture Notes in Computer Science. Springer, 1999, pp. 295–310.

[49] David Chaum and Torben Pryds Pedersen. "Wallet databases with observers". In: *Annual international cryptology conference*. Springer. 1992, pp. 89–105.

[50] Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. "Verifiable Random Functions". In: *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*. IEEE Computer Society, 1999, pp. 120–130.

[51] Yevgeniy Dodis and Aleksandr Yampolskiy. "A verifiable random function with short proofs and keys". In: *International Workshop on Public Key Cryptography*. Springer. 2005, pp. 416–431.

[52] Larry K Michaelsen, Warren E Watson, and Robert H Black. "A realistic test of individual versus group consensus decision making." In: *Journal of Applied Psychology* 74.5 (1989), p. 834.

[53] Eric Bonabeau. "Decisions 2.0: The power of collective intelligence". In: *MIT Sloan management review* 50.2 (2009), p. 45.

[54] Joshua Becker, Devon Brackbill, and Damon Centola. "Network dynamics of social influence in the wisdom of crowds". In: *Proceedings of the national academy of sciences* 114.26 (2017), E5070–E5076.

[55] Gianluca Elia, Alessandro Margherita, and Giuseppina Passiante. "Digital entrepreneurship ecosystem: How digital technologies and collective intelligence are reshaping the entrepreneurial process". In: *Technological Forecasting and Social Change* 150 (2020), p. 119791.

[56] Peide Liu, Xiaohong Zhang, and Witold Pedrycz. "A consensus model for hesitant fuzzy linguistic group decision-making in the framework of Dempster–Shafer evidence theory". In: *Knowledge-Based Systems* 212 (2021), p. 106559.

[57] Jinhyo Joseph Yun et al. "Collective intelligence: The creative way from knowledge to open innovation". In: *Science, Technology and Society* 26.2 (2021), pp. 201–222.

[58] Ming Zeng et al. “Primarily research for multi module cooperative autonomous mode of energy internet under blockchain framework”. In: *Proceedings of the CSEE* 37.13 (2017), pp. 3672–3681.

[59] Desmond Johnson. “Blockchain-based voting in the US and EU constitutional orders: a digital technology to secure democratic values?” In: *European Journal of Risk Regulation* 10.2 (2019), pp. 330–358.

[60] Kangning Zheng et al. “Blockchain adoption for information sharing: risk decision-making in spacecraft supply chain”. In: *Enterprise Information Systems* 15.8 (2021), pp. 1070–1091.

[61] Sathishkumar Ranganathan, Muralindran Mariappan, and Karthigayan Muthukaruppan PG. “Design Methodology For Using Blockchain In Swarm Robotics”. In: *2021 IEEE 19th Student Conference on Research and Development (SCOReD)*. IEEE. 2021, pp. 76–81.

[62] Lawrence Lessig. “Code is law”. In: *Harvard magazine* 1 (2000), p. 2000.

[63] Nick Webb. “A fork in the blockchain: income tax and the bitcoin/bitcoin cash hard fork”. In: *North Carolina Journal of Law & Technology* 19.4 (2018), p. 283.

[64] Rowan van Pelt et al. “Defining blockchain governance: a framework for analysis and comparison”. In: *Information Systems Management* 38.1 (2021), pp. 21–41.

[65] Aggelos Kiayias and Philip Lazos. “SoK: Blockchain Governance”. In: *arXiv preprint arXiv:2201.07188* (2022).

[66] Daniel Ferreira, Jin Li, and Radoslawa Nikolowa. “Corporate capture of blockchain governance”. In: *The Review of Financial Studies* 36.4 (2023), pp. 1364–1407.

[67] *Ethereum Improvement Proposals*. `https://eips.ethereum.org/`. Accessed: 2022-06-01. 2015.

[68] Geoffrey Hayes Robert Leshner. *Compound: The Money Market Protocol*. `https://compound.finance/documents/Compound.Whitepaper.pdf`. Accessed: 2022-06-01. 2019.

[69] *Zcash development and governance*. `https://z.cash/zcash-development-and-governance/`. Accessed: 2022-06-01. 2016.

[70] *Zcash Protocol Specification*. `https://zips.z.cash/protocol/protocol.pdf/`. Accessed: 2022-06-01. 2022.

[71] *Dash Documentation*. `https://docs.dash.org/en/stable/`. Accessed: 2022-06-01. 2018.

[72] *Decred Documentation*. `https://docs.decred.org/`. Accessed: 2022-06-01. 2016.

[73] *The Maker Protocol: MakerDAO's Multi-Collateral Dai (MCD) System*. `https://makerdao.com/en/whitepaper/`. Accessed: 2022-06-01. 2017.

[74] Nida Khan et al. "Blockchain Governance: An Overview and Prediction of Optimal Strategies using Nash Equilibrium". In: *arXiv preprint arXiv:2003.09241* (2020).

[75] Svante Janson. *PhragménÔÇÖs and ThieleÔÇÖs election methods*. Tech. rep. Technical report, 2016.

[76] Friðrik Þ Hjálmarsson et al. "Blockchain-based e-voting system". In: *2018 IEEE 11th international conference on cloud computing (CLOUD)*. IEEE. 2018, pp. 983–986.

[77] Saggi Nevo and Henry Kim. "How to compare and analyse risks of internet voting versus other modes of voting". In: *Electronic Government* 3.1 (2006), pp. 105–112.

[78] Anthony J Perez and Ebrima N Ceesay. "Improving end-to-end verifiable voting systems with blockchain technologies". In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE. 2018, pp. 1108–1115.

[79] Olawande Daramola and Darren Thebus. "Architecture-centric evaluation of blockchain-based smart contract e-voting for national elections". In: *Informatics*. Vol. 7. 2. MDPI. 2020, p. 16.

[80] Ruhi Taş and Ömer Özgür Tanrıöver. "A systematic review of challenges and opportunities of blockchain for E-voting". In: *Symmetry* 12.8 (2020), p. 1328.

[81] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. "A practical secret voting scheme for large scale elections". In: *International Workshop on the Theory and Application of Cryptographic Techniques*. Springer. 1992, pp. 244–251.

[82] Rachid Anane, Richard Freeland, and Georgios Theodoropoulos. "E-voting requirements and implementation". In: *The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007)*. IEEE. 2007, pp. 382–392.

[83] Véronique Cortier et al. "Sok: Verifiability notions for e-voting protocols". In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2016, pp. 779–798.

[84] Wenbin Zhang et al. "A privacy-preserving voting protocol on blockchain". In: *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*. IEEE. 2018, pp. 401–408.

[85] Nir Kshetri and Jeffrey Voas. "Blockchain-enabled e-voting". In: *Ieee Software* 35.4 (2018), pp. 95–99.

[86] Freya Sheer Hardwick et al. "E-voting with blockchain: An e-voting protocol with decentralisation and voter privacy". In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE. 2018, pp. 1561–1567.

[87] Zhichao Zhao and T-H Hubert Chan. "How to vote privately using bitcoin". In: *International Conference on Information and Communications Security*. Springer. 2015, pp. 82–96.

[88] Silvia Bartolucci, Pauline Bernat, and Daniel Joseph. "SHARVOT: secret SHARe-based VOTing on the blockchain". In: *Proceedings of the 1st international workshop on emerging trends in software engineering for blockchain*. 2018, pp. 30–34.

[89] Sarad Venugopalan and Ivan Homoliak. "Always on Voting: A Framework for Repetitive Voting on the Blockchain". In: *arXiv preprint arXiv:2107.10571* (2021).

[90] Yang Yang et al. "Priscore: blockchain-based self-tallying election system supporting score voting". In: *IEEE Transactions on Information Forensics and Security* 16 (2021), pp. 4705–4720.

[91] Truc Nguyen and My T Thai. "zVote: A Blockchain-based Privacy-preserving Platform for Remote E-voting". In: *ICC 2022-IEEE International Conference on Communications*. IEEE. 2022, pp. 4745–4750.

[92] Syada Tasmia Alvi et al. "DVTChain: A blockchain-based decentralized mechanism to ensure the security of digital voting system voting system". In: *Journal of King Saud University-Computer and Information Sciences* 34.9 (2022), pp. 6855–6871.

[93] Chenchen Li et al. "AMVchain: authority management mechanism on blockchain-based voting systems". In: *Peer-to-peer Networking and Applications* 14 (2021), pp. 2801–2812.

[94] Htet Ne Oo and AM Aung. "A survey of different electronic voting systems". In: *International Journal of Scientific Engineering and Technology Research* 3.16 (2014), pp. 3460–3464.

[95] Yinghui Luo et al. "A new election algorithm for DPos consensus mechanism in blockchain". In: *2018 7th international conference on digital home (ICDH)*. IEEE. 2018, pp. 116–120.

[96] Drew Springall et al. "Security analysis of the Estonian internet voting system". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014, pp. 703–715.

[97] SJ Lewis, O Pereira, and V Teague. *The use of trapdoor commitments in bayer-groth proofs and the implications for the verifiabilty of the scytl-swisspost internet voting system*. 2019.

[98] Mourine Achieng and Ephias Ruhode. "The adoption and challenges of electronic voting technologies within the South African context". In: *arXiv preprint arXiv:1312.2406* (2013).

[99] Elham Akbari et al. "From blockchain to internet-based voting". In: *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2017, pp. 218–221.

[100] Rifa Hanifatunnisa and Budi Rahardjo. "Blockchain based e-voting recording system design". In: *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*. IEEE. 2017, pp. 1–6.

[101] Sanjeev Verma and Ashutosh Sheel. "Blockchain for government organizations: past, present and future". In: *Journal of Global Operations and Strategic Sourcing* (2022).

[102] Bin Yu et al. "Platform-independent secure blockchain-based voting system". In: *International Conference on Information Security*. Springer. 2018, pp. 369–386.

[103] Peng Li and Junzuo Lai. "LaT-Voting: Traceable anonymous E-voting on blockchain". In: *Network and System Security: 13th International Conference, NSS 2019, Sapporo, Japan, December 15–18, 2019, Proceedings 13*. Springer. 2019, pp. 234–254.

[104] Sebastian Gajek and Marco Lewandowsky. "Trustless, censorship-resilient and scalable votings in the permission-based blockchain model". In: *European Conference on Parallel Processing*. Springer. 2020, pp. 54–65.

[105] Anwar Shah. *Participatory budgeting*. World Bank Publications, 2007.

[106] Yves Cabannes. "Participatory budgeting: a significant contribution to participatory democracy". In: *Environment and urbanization* 16.1 (2004), pp. 27–46.

[107] Tree Bressen. "Consensus decision making". In: *The change handbook: The definitive resource on todayÔÇÖs best methods for engaging whole systems* 495 (2007), pp. 212–217.

[108] Guillermo A O'donell. "Delegative democracy". In: *Journal of democracy* 5.1 (1994), pp. 55–69.

[109] Ahmad Mustanir et al. "Democratic Model On Decision-Making At Deliberations Of Development Planning". In: *International Conference on Government Leadership and Social Science (ICOGLASS). Demanding Governance Accountability and Promoting Democratic Leadership for Public Welfare Achievement*. Vol. 110. 2018, p. 115.

[110] Robert Münscher, Max Vetter, and Thomas Scheuerle. "A review and taxonomy of choice architecture techniques". In: *Journal of Behavioral Decision Making* 29.5 (2016), pp. 511–524.

[111] Peter G Neumann. "Security criteria for electronic voting". In: *16th National Computer Security Conference*. Vol. 29. 1993, pp. 478–481.

[112] Steve Kremer, Mark Ryan, and Ben Smyth. "Election Verifiability in Electronic Voting Protocols". In: *Computer Security - ESORICS 2010, 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings*. Ed. by Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou. Vol. 6345. Lecture Notes in Computer Science. Springer, 2010, pp. 389–404. DOI: 10.1007/978-3-642-15497-3_24. URL: https://doi.org/10.1007/978-3-642-15497-3%5C_24.

[113] Yang Yang et al. "PriScore: Blockchain-Based Self-Tallying Election System Supporting Score Voting". In: *IEEE Trans. Inf. Forensics Secur.* 16 (2021), pp. 4705–4720. DOI: 10.1109/TIFS.2021.3108494. URL: https://doi.org/10.1109/TIFS.2021.3108494.

[114] Julio César Pérez Carcía, Abderrahim Benslimane, and Samia Boutalbi. "Blockchain-based system for e-voting using Blind Signature Protocol". In: *IEEE Global Communications Conference, GLOBECOM 2021, Madrid, Spain, December 7-11, 2021*. IEEE, 2021, pp. 1–6. DOI: 10.1109/GLOBECOM46510.2021.9685189. URL: https://doi.org/10.1109/GLOBECOM46510.2021.9685189.

[115] Yannan Li et al. "A Blockchain-Based Self-Tallying Voting Protocol in Decentralized IoT". In: *IEEE Trans. Dependable Secur. Comput.* 19.1 (2022), pp. 119–130. DOI: 10.1109/TDSC.2020.2979856. URL: https://doi.org/10.1109/TDSC.2020.2979856.

[116] Lumin Shi et al. "On capturing DDoS traffic footprints on the Internet". In: *IEEE Transactions on Dependable and Secure Computing* 19.4 (2021), pp. 2755–2770.

[117] Richard H Thaler, Cass R Sunstein, and John P Balz. "Choice architecture". In: *The behavioral foundations of public policy*. Princeton University Press, 2013, pp. 428–439.

[118] Jon Fraenkel and Bernard Grofman. "The Borda Count and its real-world alternatives: Comparing scoring rules in Nauru and Slovenia". In: *Australian Journal of Political Science* 49.2 (2014), pp. 186–205.

[119] Benjamin Reilly. "Social choice in the south seas: Electoral innovation and the borda count in the pacific island countries". In: *International Political Science Review* 23.4 (2002), pp. 355–372.

[120] Arash Mirzaei et al. "Algorand Blockchain". In: *Blockchains: A Handbook on Fundamentals, Platforms and Applications*. Springer, 2023, pp. 173–193.

[121] Fangyu Gai et al. “Cumulus: A Secure BFT-based Sidechain for Off-chain Scaling”. In: *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*. 2021, pp. 1–6. DOI: 10.1109/IWQOS52092.2021.9521363.

[122] Peng Li, Toshiaki Miyazaki, and Wanlei Zhou. “Secure balance planning of off-blockchain payment channel networks”. In: *IEEE INFOCOM 2020-IEEE conference on computer communications*. IEEE. 2020, pp. 1728–1737.

[123] Yaqin Wu, Pengxin Song, Fuxin Wang, et al. “Hybrid consensus algorithm optimization: A mathematical method based on POS and PBFT and its application in blockchain”. In: *Mathematical Problems in Engineering* 2020 (2020).

[124] Junfeng Tian, Jin Tian, and Hongwei Xu. “TSBFT: A scalable and efficient leaderless byzantine consensus for consortium blockchain”. In: *Computer Networks* 222 (2023), p. 109541.

[125] Peiyao Sheng et al. “Player-replaceability and forensic support are two sides of the same (crypto) coin”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2023, pp. 56–74.

[126] Yvo Desmedt and Yair Frankel. “Threshold cryptosystems”. In: *Conference on the Theory and Application of Cryptology*. Springer. 1989, pp. 307–315.

[127] Dan Boneh and Matt Franklin. “Identity-based encryption from the Weil pairing”. In: *Annual international cryptology conference*. Springer. 2001, pp. 213–229.

[128] Aniket Kate and Ian Goldberg. “Distributed private-key generators for identity-based cryptography”. In: *International Conference on Security and Cryptography for Networks*. Springer. 2010, pp. 436–453.

[129] Moni Naor, Benny Pinkas, and Omer Reingold. “Distributed Pseudo-random Functions and KDCs”. In: *Advances in Cryptology - EUROCRYPT ’99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*. Ed. by Jacques Stern. Vol. 1592. Lecture Notes in Computer Science. Springer, 1999, pp. 327–346.

[130] Dan Boneh, Ben Lynn, and Hovav Shacham. “Short signatures from the Weil pairing”. In: *International conference on the theory and application of cryptology and information security*. Springer. 2001, pp. 514–532.

[131] Yvo Desmedt and Yair Frankel. “Shared Generation of Authenticators and Signatures (Extended Abstract)”. In: *Advances in Cryptology - CRYPTO ’91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*. Ed. by Joan Feigenbaum. Vol. 576. Lecture Notes in Computer Science. Springer, 1991, pp. 457–469.

[132] Rosario Gennaro et al. “Robust threshold DSS signatures”. In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 1996, pp. 354–371.

[133] Eleftherios Kokoris Kogias, Dahlia Malkhi, and Alexander Spiegelman. “Asynchronous Distributed Key Generation for Computationally-Secure Randomness, Consensus, and Threshold Signatures.” In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 1751–1767.

[134] Philipp Schindler et al. “Hydrand: Efficient continuous distributed randomness”. In: *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2020, pp. 73–89.

[135] Ittai Abraham, Dahlia Malkhi, and Alexander Spiegelman. “Asymptotically Optimal Validated Asynchronous Byzantine Agreement”. In: *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*. Ed. by Peter Robinson and Faith Ellen. ACM, 2019, pp. 337–346.

[136] Maofan Yin et al. “HotStuff: BFT Consensus with Linearity and Responsiveness”. In: *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*. Ed. by Peter Robinson and Faith Ellen. ACM, 2019, pp. 347–356.

[137] Guy Golan-Gueta et al. “SBFT: A Scalable and Decentralized Trust Infrastructure”. In: *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2019, Portland, OR, USA, June 24-27, 2019*. IEEE, 2019, pp. 568–580.

[138] Daniela Tulone. “A Scalable and Intrusion-tolerant Digital Time-stamping System”. In: *Proceedings of IEEE International Conference on Communications, ICC 2006, Istanbul, Turkey, 11-15 June 2006*. IEEE, 2006, pp. 2357–2363.

[139] Alexis Bonnecaze and Philippe Trebuchet. “Threshold signature for distributed time stamping scheme”. In: *Ann. des Télécommunications* 62.11-12 (2007), pp. 1353–1364.

[140] Theodore M Wong, Chenxi Wang, and Jeannette M Wing. “Verifiable secret redistribution for archive systems”. In: *First International IEEE Security in Storage Workshop, 2002. Proceedings.* IEEE. 2002, pp. 94–105.

[141] Eleftherios Kokoris-Kogias et al. “Verifiable management of private data under byzantine failures”. In: *Cryptol. ePrint Arch., Tech. Rep* 209 (2018), p. 2018.

[142] David Galindo et al. “Fully Distributed Verifiable Random Functions and their Application to Decentralised Random Beacons”. In: *IEEE European Symposium on Security and Privacy, EuroS&P 2021, Vienna, Austria, September 6-10, 2021*. IEEE, 2021, pp. 88–102.

[143] Ewa Syta et al. “Scalable Bias-Resistant Distributed Randomness”. In: *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 2017, pp. 444–460.

[144] Aniket Kate and Ian Goldberg. “Distributed key generation for the internet”. In: *2009 29th IEEE International Conference on Distributed Computing Systems*. IEEE. 2009, pp. 119–128.

[145] Torben P. Pedersen. “Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing”. In: *Advances in Cryptology - CRYPTO ’91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*. Ed. by Joan Feigenbaum. Vol. 576. Lecture Notes in Computer Science. Springer, 1991, pp. 129–140.

[146] Rosario Gennaro et al. “Secure Applications of Pedersen’s Distributed Key Generation Protocol”. In: *Topics in Cryptology - CT-RSA 2003, The Cryptographers’ Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings*. Ed. by Marc Joye. Vol. 2612. Lecture Notes in Computer Science. Springer, 2003, pp. 373–390.

[147] Ran Canetti et al. “Adaptive Security for Threshold Cryptosystems”. In: *Advances in Cryptology - CRYPTO ’99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*. Ed. by Michael J. Wiener. Vol. 1666. Lecture Notes in Computer Science. Springer, 1999, pp. 98–115.

[148] Aniket Kate. “Distributed Key Generation and Its Applications”. PhD thesis. University of Waterloo, Ontario, Canada, 2010.

[149] John F. Canny and Stephen Sorkin. “Practical Large-Scale Distributed Key Generation”. In: *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*. Ed. by Christian Cachin and Jan Camenisch. Vol. 3027. Lecture Notes in Computer Science. Springer, 2004, pp. 138–152.

[150] Wafa Neji, Kaouther Blibech Sinaoui, and Narjes Ben Rajeb. “Distributed key generation protocol with a new complaint management strategy”. In: *Secur. Commun. Networks* 9.17 (2016), pp. 4585–4595.

[151] Pierre-Alain Fouque and Jacques Stern. “One Round Threshold Discrete-Log Key Generation without Private Channels”. In: *Public Key Cryptography, 4th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2001, Cheju Island, Korea, February 13-15, 2001, Proceedings*. Ed. by Kwangjo Kim. Vol. 1992. Lecture Notes in Computer Science. Springer, 2001, pp. 300–316.

[152] Philipp Schindler et al. “ETHDKG: Distributed Key Generation with Ethereum Smart Contracts”. In: *IACR Cryptol. ePrint Arch.* 2019 (2019), p. 985.

[153] Alin Tomescu et al. “Towards Scalable Threshold Cryptosystems”. In: *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*. IEEE, 2020, pp. 877–893.

[154] Jens Groth. “Non-interactive distributed key generation and key resharing”. In: *Cryptology ePrint Archive* (2021).

[155] Liang Zhang et al. “1-Round Distributed Key Generation With Efficient Reconstruction Using Decentralized CP-ABE”. In: *IEEE Transactions on Information Forensics and Security* 17 (2022), pp. 894–907.

[156] Kobi Gurkan et al. “Aggregatable distributed key generation”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2021, pp. 147–176.

[157] Sourav Das et al. “Practical asynchronous distributed key generation”. In: *Cryptology ePrint Archive* (2021).

[158] Abraham Ittai et al. “Reaching Consensus for Asynchronous Distributed Key Generation”. In: *PODC ’21: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, July 26-30, 2021*. ACM, 2021, pp. 363–373.

[159] Sourav Das, Zhuolun Xiang, and Ling Ren. “Asynchronous data dissemination and its applications”. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021, pp. 2705–2721.

[160] Aniket Kate, Yizhou Huang, and Ian Goldberg. “Distributed Key Generation in the Wild”. In: *IACR Cryptol. ePrint Arch.* 2012 (2012), p. 377.

[161] Paul Feldman. “A Practical Scheme for Non-interactive Verifiable Secret Sharing”. In: *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*. IEEE Computer Society, 1987, pp. 427–437.

[162] Rosario Gennaro et al. “Robust Threshold DSS Signatures”. In: *Inf. Comput.* 164.1 (2001), pp. 54–84.

[163] Sisi Duan, Michael K. Reiter, and Haibin Zhang. “BEAT: Asynchronous BFT Made Practical”. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*. Ed. by David Lie et al. ACM, 2018, pp. 2028–2041.

[164] Javier Herranz, Alexandre Ruiz, and Germán Sáez. “Signcryption schemes with threshold unsigncryption, and applications”. In: *Des. Codes Cryptogr.* 70.3 (2014), pp. 323–345.

[165] Feng Wang, Chin-Chen Chang, and Lein Harn. “Simulatable and secure certificate-based threshold signature without pairings”. In: *Security and Communication Networks* 7.11 (2014), pp. 2094–2103.

[166] Javier Herranz, Alexandre Ruiz, and Germán Sáez. "Signcryption schemes with threshold unsigncryption, and applications". In: *Designs, codes and cryptography* 70.3 (2014), pp. 323–345.

[167] Nasrollah Pakniat, Mahnaz Noroozi, and Ziba Eslami. "Distributed key generation protocol with hierarchical threshold access structure". In: *IET Information Security* 9.4 (2015), pp. 248–255.

[168] Benoît Libert and Moti Yung. "Adaptively secure non-interactive threshold cryptosystems". In: *Theor. Comput. Sci.* 478 (2013), pp. 76–100.

[169] Yair Frankel, Philip D. MacKenzie, and Moti Yung. "Adaptively secure distributed public-key systems". In: *Theor. Comput. Sci.* 287.2 (2002), pp. 535–561.

[170] Masayuki Abe and Serge Fehr. "Adaptively Secure Feldman VSS and Applications to Universally-Composable Threshold Cryptography". In: *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*. Ed. by Matthew K. Franklin. Vol. 3152. Lecture Notes in Computer Science. Springer, 2004, pp. 317–334.

[171] Ivan Damgård and Jesper Buus Nielsen. "Adaptive versus Static Security in the UC Model". In: *Provable Security - 8th International Conference, ProvSec 2014, Hong Kong, China, October 9-10, 2014. Proceedings*. Ed. by Sherman S. M. Chow et al. Vol. 8782. Lecture Notes in Computer Science. Springer, 2014, pp. 10–28.

[172] Douglas R. Stinson and Reto Strobl. "Provably Secure Distributed Schnorr Signatures and a (t, n) Threshold Scheme for Implicit Certificates". In: *Information Security and Privacy, 6th Australasian Conference, ACISP 2001, Sydney, Australia, July 11-13, 2001, Proceedings*. Ed. by Vijay Varadharajan and Yi Mu. Vol. 2119. Lecture Notes in Computer Science. Springer, 2001, pp. 417–434.

[173] Dan Boneh, Manu Drijvers, and Gregory Neven. "Compact multi-signatures for smaller blockchains". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2018, pp. 435–464.

[174] Philipp Schindler et al. "Ethdkg: Distributed key generation with ethereum smart contracts". In: *Cryptology ePrint Archive* (2019).

[175] Eleftherios Kokoris-Kogias, Dahlia Malkhi, and Alexander Spiegelman. "Asynchronous Distributed Key Generation for Computationally-Secure Randomness, Consensus, and Threshold Signatures". In: *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*. Ed. by Jay Ligatti et al. ACM, 2020, pp. 1751–1767.

[176] Andrew Miller et al. "The Honey Badger of BFT Protocols". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. Ed. by Edgar R. Weippl et al. ACM, 2016, pp. 31–42.

[177] Michael O. Rabin. “Transaction Protection by Beacons”. In: *J. Comput. Syst. Sci.* 27.2 (1983), pp. 256–267.

[178] Christian Cachin, Klaus Kursawe, and Victor Shoup. “Random Oracles in Constantinople: Practical Asynchronous Byzantine Agreement Using Cryptography”. In: *J. Cryptol.* 18.3 (2005), pp. 219–246.

[179] Chun-Chi Liu et al. “NormaChain: A Blockchain-Based Normalized Autonomous Transaction Settlement System for IoT-Based E-Commerce”. In: *IEEE Internet Things J.* 6.3 (2019), pp. 4680–4693.

[180] Anna Lisa Ferrara et al. “Practical Short Signature Batch Verification”. In: *Topics in Cryptology - CT-RSA 2009, The Cryptographers’ Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009. Proceedings*. Ed. by Marc Fischlin. Vol. 5473. Lecture Notes in Computer Science. Springer, 2009, pp. 309–324.

[181] Claus-Peter Schnorr. “Efficient Identification and Signatures for Smart Cards”. In: *Advances in Cryptology - CRYPTO ’89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*. Ed. by Gilles Brassard. Vol. 435. Lecture Notes in Computer Science. Springer, 1989, pp. 239–252.

[182] L Richard Turner. “Inverse of the Vandermonde matrix with applications”. In: (1966).

[183] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. “A secure and optimally efficient multi-authority election scheme”. In: *European transactions on Telecommunications* 8.5 (1997), pp. 481–490.

[184] Antonio M Larriba, José M Sempere, and Damián López. “A two authorities electronic vote scheme”. In: *Computers & Security* 97 (2020), p. 101940.

[185] Ehab Zaghloul, Tongtong Li, and Jian Ren. “d-BAME: Distributed Blockchain-Based Anonymous Mobile Electronic Voting”. In: *IEEE Internet of Things Journal* 8.22 (2021), pp. 16585–16597.

[186] David Chaum. “SureVote”. In: *International patent WO* 1.55940 (2001), A1.

[187] Ben Adida. “Helios: Web-based Open-Audit Voting.” In: *USENIX security symposium*. Vol. 17. 2008, pp. 335–348.

[188] David Chaum et al. “Scantegrity: End-to-end voter-verifiable optical-scan voting”. In: *IEEE Security & Privacy* 6.3 (2008), pp. 40–46.

[189] Michael R Clarkson, Stephen Chong, and Andrew C Myers. “Civitas: Toward a secure voting system”. In: *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE. 2008, pp. 354–368.

[190] Rui Joaquim, Paulo Ferreira, and Carlos Ribeiro. “EVIV: An end-to-end verifiable Internet voting system”. In: *Computers & Security* 32 (2013), pp. 170–191.

[191] Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. "End-to-end verifiable elections in the standard model". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2015, pp. 468–498.

[192] Filip Zagórski et al. "Remotegrity: Design and use of an end-to-end verifiable remote voting system". In: *International Conference on Applied Cryptography and Network Security*. Springer. 2013, pp. 441–457.

[193] Nikos Chondros et al. "Distributed, end-to-end verifiable, and privacy-preserving internet voting systems". In: *computers & security* 83 (2019), pp. 268–299.

[194] Kristjan Vassil et al. "The diffusion of internet voting. Usage patterns of internet voting in Estonia between 2005 and 2015". In: *Government Information Quarterly* 33.3 (2016), pp. 453–459.

[195] Dimitris A Gritzalis. "Principles and requirements for a secure e-voting system". In: *Computers & Security* 21.6 (2002), pp. 539–556.

[196] Steve Kremer, Mark Ryan, and Ben Smyth. "Election verifiability in electronic voting protocols". In: *European Symposium on Research in Computer Security*. Springer. 2010, pp. 389–404.

[197] David L Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms". In: *Communications of the ACM* 24.2 (1981), pp. 84–90.

[198] Choonsik Park, Kazutomo Itoh, and Kaoru Kurosawa. "Efficient Anonymous Channel and All/Nothing Election Scheme". In: *Advances in Cryptology — EUROCRYPT '93*. Ed. by Tor Helleseth. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 248–259. ISBN: 978-3-540-48285-7.

[199] David Chaum, Peter YA Ryan, and Steve Schneider. "A practical voter-verifiable election scheme". In: *European Symposium on Research in Computer Security*. Springer. 2005, pp. 118–139.

[200] Kazi Md Rokibul Alam et al. "An electronic voting scheme based on revised-SVRM and confirmation numbers". In: *IEEE Transactions on Dependable and Secure Computing* 18.1 (2019), pp. 400–410.

[201] Donghoon Chang, Amit Kumar Chauhan, Jinkeon Kang, et al. "Apollo: End-to-end verifiable voting protocol using mixnet and hidden tweaks". In: *ICISC 2015*. Springer. 2015, pp. 194–209.

[202] Thomas Haines, Rajeev Goré, and Bhavesh Sharma. "Did you mix me? formally verifying verifiable mix nets in electronic voting". In: *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2021, pp. 1748–1765.

[203] David Chaum. "Blind signatures for untraceable payments". In: *Advances in cryptology*. Springer. 1983, pp. 199–203.

[204] Jan L Camenisch, Jean-Marc Piveteau, and Markus A Stadler. "Blind signatures based on the discrete logarithm problem". In: *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer. 1994, pp. 428–432.

[205] Mahender Kumar, Satish Chand, and Chittaranjan Padmanabha Katti. "A secure end-to-end verifiable internet-voting system using identity-based blind signature". In: *IEEE Systems Journal* 14.2 (2020), pp. 2032–2041.

[206] Hongyu Zhang, Qianzi You, and Junxing Zhang. "A lightweight electronic voting scheme based on blind signature and Kerberos mechanism". In: *2015 IEEE 5th International Conference on Electronics Information and Emergency Communication*. IEEE. 2015, pp. 210–214.

[207] Ahsan Aziz. "Coercion-resistant e-voting scheme with blind signatures". In: *2019 Cybersecurity and Cyberforensics Conference (CCC)*. IEEE. 2019, pp. 143–151.

[208] Pascal Paillier. "Public-key cryptosystems based on composite degree residuosity classes". In: *International conference on the theory and applications of cryptographic techniques*. Springer. 1999, pp. 223–238.

[209] Ronald L. Rivest, Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 26.1 (1983), pp. 96–99.

[210] Ronald Cramer et al. "Multi-authority secret-ballot elections with linear work". In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 1996, pp. 72–83.

[211] Martin Hirt and Kazue Sako. "Efficient receipt-free voting based on homomorphic encryption". In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2000, pp. 539–556.

[212] Xun Yi and Eiji Okamoto. "Practical remote end-to-end voting scheme". In: *International Conference on Electronic Government and the Information Systems Perspective*. Springer. 2011, pp. 386–400.

[213] Josh D Cohen and Michael J Fischer. *A robust and verifiable cryptographically secure election scheme*. Yale University. Department of Computer Science, 1985.

[214] Aggelos Kiayias, Thomas Zacharias, and Bingsheng Zhang. "An efficient E2E verifiable e-voting system without setup assumptions". In: *IEEE Security & Privacy* 15.3 (2017), pp. 14–23.

[215] Xingyue Fan et al. "HSE-Voting: A secure high-efficiency electronic voting scheme based on homomorphic signcryption". In: *Future Generation Computer Systems* 111 (2020), pp. 754–762.

[216] Susan Bell et al. "{STAR-Vote}: A Secure, Transparent, Auditable, and Reliable Voting System". In: *2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 13)*. 2013.

[217] Oksana Kulyk et al. "Coercion-resistant proxy voting". In: *computers & Security* 71 (2017), pp. 88–99.

[218] Patrick McCorry, Siamak F Shahandashti, and Feng Hao. "A smart contract for boardroom voting with maximum voter privacy". In: *International conference on financial cryptography and data security*. Springer. 2017, pp. 357–375.

[219] Mohamed Seifelnasr, Hisham S Galal, and Amr M Youssef. "Scalable open-vote network on ethereum". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2020, pp. 436–450.

[220] Christian Killer et al. "Provotum: a blockchain-based and end-to-end verifiable remote electronic voting system". In: *2020 IEEE 45th Conference on Local Computer Networks (LCN)*. IEEE. 2020, pp. 172–183.

[221] Somnath Panja and Bimal Roy. "A Secure End-to-End Verifiable E-Voting System Using Zero-Knowledge Proof and Blockchain". In: *A Tribute to the Legend of Professor CR Rao*. Springer, 2021, pp. 45–48.

[222] Yannan Li et al. "A blockchain-based self-tallying voting protocol in decentralized IoT". In: *IEEE Transactions on Dependable and Secure Computing* (2020).

[223] Gaby G Dagher et al. "Broncovote: Secure voting system using ethereumÔÇÖs blockchain". In: (2018).

[224] Sanjai Bhagat and James A Brickley. "Cumulative voting: The value of minority shareholder voting rights". In: *The Journal of Law and Economics* 27.2 (1984), pp. 339–365.

[225] Jack Santucci. "Variants of ranked-choice voting from a strategic perspective". In: *Politics and Governance* 9.2 (2021), pp. 344–353.

[226] Ralf Küsters et al. "Ordinos: A verifiable tally-hiding e-voting system". In: *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2020, pp. 216–235.

[227] Olivier de Marneffe Ben Adida and Olivier Pereira. *Helios voting system*. Online; date last accessed: 2022-8-21. URL: `https://vote.heliosvoting.org`.

[228] Ben Adida et al. "Electing a university president using open-audit voting: Analysis of real-world use of Helios". In: *EVT/WOTE* 9.10 (2009).

[229] Véronique Cortier et al. "A simple alternative to Benaloh challenge for the cast-as-intended property in Helios/Belenios". In: (2019).

[230] Aggelos Kiayias and Moti Yung. "Self-tallying elections and perfect ballot secrecy". In: *International Workshop on Public Key Cryptography*. Springer. 2002, pp. 141–158.

[231] Feng Hao, Peter YA Ryan, and Piotr Zieliński. "Anonymous voting by two-round public discussion". In: *IET Information Security* 4.2 (2010), pp. 62–67.

[232] Stefano Bistarelli et al. "An end-to-end voting-system based on bitcoin". In: *Proceedings of the Symposium on Applied Computing*. 2017, pp. 1836–1841.

[233] Wei-Jr Lai et al. "Date: A decentralized, anonymous, and transparent e-voting system". In: *2018 1st IEEE international conference on hot information-centric networking (HotICN)*. IEEE. 2018, pp. 24–29.

[234] R Krishnamurthy, Geetanjali Rathee, and Naveen Jaglan. "An enhanced security mechanism through blockchain for E-polling/counting process using IoT devices". In: *Wireless Networks* 26.4 (2020), pp. 2391–2402.

[235] Huilin Li et al. "A blockchain-based traceable self-tallying E-voting protocol in AI era". In: *IEEE Transactions on Network Science and Engineering* 8.2 (2020), pp. 1019–1032.

[236] Chinnapong Angsuchotmetee, Pisal Setthawong, and Sapjarern Udomviriyalanon. "Blockvote: An architecture of a blockchain-based electronic voting system". In: *2019 23rd International Computer Science and Engineering Conference (ICSEC)*. IEEE. 2019, pp. 110–116.

[237] Syada Tasmia Alvi, Mohammed Nasir Uddin, and Linta Islam. "Digital voting: A blockchain-based e-voting system using biohash and smart contract". In: *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE. 2020, pp. 228–233.

[238] Xinyu Zhang et al. "An Efficient E2E Crowd Verifiable E-voting System". In: *IEEE Transactions on Dependable and Secure Computing* (2021).

[239] Julio César Perez Carcia, Abderrahim Benslimane, and Samia Boutalbi. "Blockchain-based system for e-voting using Blind Signature Protocol". In: *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2021, pp. 01–06.

[240] Jens Groth and Markulf Kohlweiss. "One-Out-of-Many Proofs: Or How to Leak a Secret and Spend a Coin". In: *EUROCRYPT 2015*. Ed. by Elisabeth Oswald and Marc Fischlin. Springer, 2015, pp. 253–280.

[241] Dan Boneh, Antoine Joux, and Phong Q Nguyen. "Why textbook ElGamal and RSA encryption are insecure". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2000, pp. 30–43.

[242] Ilan Komargodski, Moni Naor, and Eylon Yogev. "How to share a secret, infinitely". In: *Theory of Cryptography Conference*. Springer. 2016, pp. 485–514.

[243] Ilan Komargodski and Anat Paskin-Cherniavsky. "Evolving secret sharing: dynamic thresholds and robustness". In: *Theory of Cryptography Conference*. Springer. 2017, pp. 379–393.

[244] Amos Beimel and Hussien Othman. "Evolving ramp secret sharing with a small gap". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2020, pp. 529–555.

[245] Sai Krishna Deepak Maram et al. "CHURP: dynamic-committee proactive secret sharing". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2019, pp. 2369–2386.

[246] Fan Zhang et al. "Paralysis proofs: Secure dynamic access structures for cryptocurrency custody and more". In: *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*. 2019, pp. 1–15.

[247] Craig Gentry et al. "YOSO: you only speak once". In: *Annual International Cryptology Conference*. Springer. 2021, pp. 64–93.

[248] Craig Gentry et al. "Random-index PIR and applications". In: *Theory of Cryptography Conference*. Springer. 2021, pp. 32–61.

[249] Jing Chen et al. "Algorand agreement: Super fast and partition resilient byzantine agreement". In: *Cryptology ePrint Archive* (2018).

[250] Peiyao Sheng et al. "Player-Replaceability and Forensic Support are Two Sides of the Same (Crypto) Coin". In: *IACR Cryptol. ePrint Arch.* (2022), p. 1513. URL: https://eprint.iacr.org/2022/1513.

[251] Jon Truby. "Decarbonizing Bitcoin: Law and policy choices for reducing the energy consumption of Blockchain technologies and digital currencies". In: *Energy research & social science* 44 (2018), pp. 399–410.

[252] Sheng Zhang, Jie Wu, and Sanglu Lu. "Collaborative Mobile Charging". In: *IEEE Trans. Computers* 64.3 (2015), pp. 654–667.

[253] Jing Chen and Silvio Micali. "Algorand: A secure and efficient distributed ledger". In: *Theor. Comput. Sci.* 777 (2019), pp. 155–183.

[254] Markus Stadler. "Publicly Verifiable Secret Sharing". In: *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*. Ed. by Ueli M. Maurer. Vol. 1070. Lecture Notes in Computer Science. Springer, 1996, pp. 190–199.

[255] Volkan Dedeoglu et al. "A trust architecture for blockchain in IoT". In: *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 2019, pp. 190–199.

[256] Ruijuan Zheng et al. “A collaborative analysis method of user abnormal behavior based on reputation voting in cloud environment”. In: *Future Generation Computer Systems* 83 (2018), pp. 60–74.

[257] Muhammad Ajmal Azad and Samiran Bag. “Decentralized privacy-aware collaborative filtering of smart spammers in a telecommunication network”. In: *Proceedings of the Symposium on Applied Computing*. 2017, pp. 1711–1717.

[258] Félix Gómez Mármol and Gregorio Martínez Pérez. “TRIP, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks”. In: *Journal of network and computer applications* 35.3 (2012), pp. 934–941.

[259] Hao Hu et al. “REPLACE: A reliable trust-based platoon service recommendation scheme in VANET”. In: *IEEE Transactions on Vehicular Technology* 66.2 (2016), pp. 1786–1797.

[260] Ferheen Ayaz et al. “A voting blockchain based message dissemination in vehicular ad-hoc networks (VANETs)”. In: *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE. 2020, pp. 1–6.

[261] Tonghe Wang et al. “RBT: A distributed reputation system for blockchain-based peer-to-peer energy trading with fairness consideration”. In: *Applied Energy* 295 (2021), p. 117056.

[262] Junqin Huang et al. “Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism”. In: *IEEE Transactions on Industrial Informatics* 15.6 (2019), pp. 3680–3689.

[263] Alex Biryukov and Daniel Feher. “ReCon: Sybil-resistant consensus from reputation”. In: *Pervasive and Mobile Computing* 61 (2020), p. 101109.

[264] Eric Ke Wang et al. “PoRX: A reputation incentive scheme for blockchain consensus of IIoT”. In: *Future Generation Computer Systems* 102 (2020), pp. 140–151.

[265] Audun Jøsang, Shane Hird, and Eric Faccer. “Simulating the Effect of Reputation Systems on E-markets”. In: *Trust Management, First International Conference, iTrust 2003, Heraklion, Crete, Greece, May 28-30, 2002, Proceedings*. Ed. by Paddy Nixon and Sotirios Terzis. Vol. 2692. Lecture Notes in Computer Science. Springer, 2003, pp. 179–194.

[266] Marcela T de Oliveira et al. “Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications”. In: *Computer Networks* 179 (2020), p. 107367.

[267] Chenyu Huang et al. “Repchain: A reputation-based secure, fast, and high incentive blockchain system via sharding”. In: *IEEE Internet of Things Journal* 8.6 (2020), pp. 4291–4304.

[268] Chenyu Huang et al. “ZkRep: A privacy-preserving scheme for reputation-based blockchain system”. In: *IEEE Internet of Things Journal* 9.6 (2021), pp. 4330–4342.

[269] IOHK. *IOHK*. `https://iohk.io`. Accessed: 2021-12-27.