University of Technology Sydney
Faculty of Engineering and Information Technology
School of Electrical and Data Engineering

Proof-of-Stake-based Blockchain Frameworks for Smart Data Management

Cong Thanh Nguyen

A Thesis Submitted
in Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

under the supervision of

Dr. Hoang Dinh
A/Prof. Diep N. Nguyen
Prof. Eryk Dutkiewicz

Sydney, Australia

March 2023

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, CONG THANH NGUYEN, declare that this thesis is submitted in fulfilment of the requirements for the award of DOCTOR OF PHILOSOPHY in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used in the thesis are indicated.

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with the Ho Chi Minh City University of Technology, Vietnam.

This research is supported by the Australian Government Research Training Program.

Student Name: Cong Thanh Nguyen

Student Signature: Production Note:
Signature removed prior to publication.

Date: June 23, 2023

ABSTRACT

Proof-of-Stake-based Blockchain Frameworks for Smart Data Management

by

Cong Thanh Nguyen

Over the last few years, the Proof-of-Stake (PoS) consensus mechanism has emerged as an effective solution to data management in various areas. However, developing the PoS mechanism requires thoughtful design and consideration to overcome significant challenges for different blockchain network architectures. In single-blockchain systems, forming stake pools might significantly increase the blockchain's centralization level, posing serious threats to the network's security. Moreover, in federated-blockchain systems, the transfers of assets among the blockchains within might centralize users to a single blockchain, leaving the other blockchains vulnerable to attacks. Similarly, in sharding-based blockchain networks, the division of the blockchain into shards might weaken the blockchain's security.

This thesis develops PoS-based frameworks and proposes solutions for various data management applications to address the abovementioned issues. In the first study, we develop BlockRoam for roaming management systems, using the PoS consensus mechanism and smart contracts to address roaming fraud. Moreover, an economic model based on the Stackelberg game is proposed to address the stake pool formation's risk. Performance evaluations show that BlockRoam can achieve up to 1100x faster transaction confirmation than Bitcoin and reduce the probability of being attacked by up to 35%.

In the second study, we propose FedChain, a framework for federated-blockchain systems with a cross-chain transfer protocol for secure token transfers. In Fedchain, a novel PoS-based consensus mechanism is developed to satisfy strict security re-

quirements with improved performance. Moreover, a Stackelberg game model is developed to address the centralization risk. Simulation results show that Fedchain can reduce the transaction confirmation time by up to 23% and improve throughput by 27% compared to static approaches.

In the third study, we propose MetaShard, a sharding-based blockchain framework for Metaverse applications. Particularly, we develop a Proof-of-Engagement consensus mechanism to incentivize resource contribution for Metaverse applications. Moreover, to improve the scalability of MetaShard, we propose an innovative sharding management scheme to maximize the network's throughput while protecting the shards from 51% attacks. Numerical experiments show that our approach achieves up to 66.6% higher throughput in less than 1/30 running time and achieves globally optimal solutions in most experiments.

These results demonstrate the applicability of the PoS mechanism in data management and the ability of our solutions to address security issues in various types of PoS-based blockchain networks. Potential research directions include Metaverse, Web 3.0, and alternative consensus design to improve the PoS mechanism.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Dr. Hoang Dinh, A/Prof. Diep N. Nguyen, Prof. Eryk Dutkiewicz, and Dr. Hoang Anh Pham for all the support, guidance, and encouragement. Without them, this dissertation would have been impossible. During my study, they not only guided me to pursue great and impactful research but also gave valuable advice for my career. I am truly privileged and lucky to be supervised by them.

I would like to thank Asst. Prof. Quang-Vinh Dang who supervised my Master's thesis and introduced me to research. Without him, I would not have embarked on this wonderful journey.

I would like to thank all my colleagues and friends at the University of Technology Sydney for their support, discussion, and friendship. Thanks should also go to the SEDE admin team for handling all the paperwork and forms during my PhD study. I would like to thank the university, the UTS-HCMUT Joint Technology and Innovation Research Centre, and the Faculty of Engineering and Information Technology (FEIT) for giving me the opportunities and financial support during the past years.

I would especially like to thank my family and friends for their endless love and support that give me strength to overcome difficulties in my life.

Contents

List of Publications

Journal Papers (Accepted/Published)

J-1. **C. T. Nguyen**, D. T. Hoang, D. N. Nguyen, Y. Xiao, D. Niyato and E. Dutkiewicz, "MetaShard: A novel sharding blockchain platform for Metaverse applications," *IEEE Transactions on Mobile Computing* (accepted). (*Corresponding to Chapter 5)*

J-2. **C. T. Nguyen**, D. T. Hoang, D. N. Nguyen, Y. Xiao, H. A. Pham, E. Dutkiewicz and H. T. Nguyen, "FedChain: Secure Proof-of-Stake-based framework for federated-blockchain systems," *IEEE Transactions on Services Computing*, Early Access, doi: 10.1109/TSC.2023.3240235. (*Corresponding to Chapter 4)*

J-3. **C. T. Nguyen**, D. N. Nguyen, D. T. Hoang, T. K. Phan, D. Niyato, H. A. Pham and E. Dutkiewicz, "Elastic Resource Allocation for Coded Distributed Computing over Heterogeneous Wireless Edge Networks," *IEEE Transactions on Wireless Communications*, Early Access, , doi: 10.1109/TWC.2022.3213256.

J-4. **C. T. Nguyen**, D. N. Nguyen, D. T. Hoang, H. A. Pham, H. T. Nguyen, Y. Xiao and E. Dutkiewicz, "BlockRoam: Blockchain-Based Roaming Management System for Future Mobile Networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 11, pp. 3880-3894, 1 Nov. 2022. (*Corresponding to Chapter 3)*

J-5. **C. T. Nguyen**, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities", *IEEE Access*, vol.7, pp. 85727-45, Jun. 2019. (*Corresponding to Chapter 2)*

J-6. **C. T. Nguyen**, N. V. Huynh, N. H. Chu, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, Q. V. Pham, D. Niyato, E. Dutkiewicz and W. J. Hwang, "Transfer Learning for Wireless Networks: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 110, no. 8, pp. 1073-1115, Aug. 2022.

J-7. **C. T. Nguyen**, Y. M. Saputra, N. V. Huynh, N. T. Nguyen, T. V. Khoa, B. M. Tuan, D. N. Nguyen, D. T. Hoang, T. X. Vu, E. Dutkiewicz, S. Chatzinotas and B. Ottersten, "A Comprehensive Survey of Enabling and Emerging Technologies for Social Distancing—Part I: Fundamentals and Enabling Technologies," *IEEE Access*, vol. 8, pp. 153479-153507, Aug. 2020

J-8. **C. T. Nguyen**, Y. M. Saputra, N. V. Huynh, N. T. Nguyen, T. V. Khoa, B. M. Tuan, D. N. Nguyen, D. T. Hoang, T. X. Vu, E. Dutkiewicz, S. Chatzinotas and B. Ottersten, "A Comprehensive Survey of Enabling and Emerging Technologies for Social Distancing—Part II: Emerging Technologies and Open Issues," *IEEE Access*, vol. 8, pp. 154209-154236, Aug. 2020.

J-9. T. V. Khoa, D. T. Hoang, N. L. Trung, **C. T. Nguyen**, T. T. Quynh, D. N. Nguyen, N. V. Ha and E. Dutkiewicz, "Deep Transfer Learning: A Novel Collaborative Learning Model for Cyberattack Detection Systems in IoT Networks," *IEEE Internet of Things journal*, Early Access, doi: 10.1109/JIOT.2022.3202029.

Conference Papers (Accepted/Published)

C-1. **C. T. Nguyen**, D. T. Hoang, D. N. Nguyen and E. Dutkiewicz, "MetaChain: A Novel Blockchain-based Framework for Metaverse Applications," *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022, pp. 1-5. (*Partly corresponding to Chapter 5)*)

C-2. **C. T. Nguyen**, D. T. Hoang, D. N. Nguyen, H. A. Pham, N. H. Tuong and E. Dutkiewicz, "Blockchain-based Secure Platform for Coalition Loyalty Program Management," *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1-6. (*Partly corresponding to Chapter 4)*)

C-3. **C. T. Nguyen**, D. N. Nguyen, D. T. Hoang, H. -A. Pham, N. H. Tuong and E. Dutkiewicz, "Blockchain and Stackelberg Game Model for Roaming Fraud Prevention and Profit Maximization," *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1-6. (*Partly corresponding to Chapter 3)*)

C-4. **C. T. Nguyen**, D. N. Nguyen, D. T. Hoang, H. A. Pham and E. Dutkiewicz, "Optimize Coding and Node Selection for Coded Distributed Computing over Wireless Edge Networks," *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 1248-1253.

Book Chapter (Accepted/Published)

O-1. **C. T. Nguyen**, D. T. Hoang, D. N. Nguyen and E. Dutkiewicz, "Social distancing and related technologies: Fundamental background", in *Enabling Technologies for Social Distancing*, Institution of Engineering and Technology (IET), July 2022.

Edited Books (Under Review)

B-1. **C. T. Nguyen**, D. T. Hoang, D. N. Nguyen, L. Luu and R. Joyce R, *Proof-of-Stake for Blockchain Networks: Fundamentals, Challenges and Approaches*. London, UK: Institution of Engineering and Technology (IET) publisher, to be published in 2023.

B-2. D. T. Hoang, D. N. Nguyen, **C. T. Nguyen**, E. Hossain E and D. Niyato, *Metaverse Communication and Computing Networks: Applications, Technologies, and Approaches*. Hoboken, NJ: Wiley, to be published in 2023.

List of Figures

Abbreviation

BFT	Byzantine-Fault-Tolerance
CAPTCHA	Completely Automated Public Turing-Test to tell Computers and Humans Apart
CDRs	Call Detail Records
CG	Chain growth
CoA	Chains-of-Activity
CP	Common prefix
CQ	Chain quality
DCH	Data Clearing Houses
DPoS	Delegated Proof-of-Stake
FTS	Follow-the-Satoshi
HLR	Home Location Register
IoT	Internet-of-Things
MILP	Mixed-Integer Linear Programming
MSC/VLR	Mobile Switching Center/Visited Location Register
MSPs	Metaverse service providers
MUs	Metaverse Users
PoA	Proof-of-Activity
PoS	Proof-of-Stake
PVSS	Publicly Verifiable Secret Sharing
PoW	Proof-of-Work
PoX	Proof-of-Concepts
SPV	Simplified Payment Verification
TAP	Transfer Account Procedure
VRF	Verifiable Random Function

Chapter 1

Introduction and Literature Review

This chapter first provides the background on the development of blockchain technology as well as the challenges it is facing. Then, state-of-the-art solutions in addressing these issues are comprehensively discussed. Finally, the main contributions as well as the structure of this thesis are highlighted at the end of this chapter.

1.1 Motivations

Over the last few years, blockchain technology has been considered as one of the most significant technological breakthrough since the invention of the Internet. A blockchain is a distributed database of records shared among network participants. With the help of cryptographic hash functions, digital signatures, and distributed consensus mechanisms, once a record enters the database, it cannot be altered without the consensus of the other network participants [1]. As a result, data stored in a blockchain can be conventionally verified even in a decentralized environment, leading to to numerous blockchain applications. Cryptocurrencies, the most famous blockchain applications, have the total market capitalization of more than $200 billion by the time this article is written, with more than 2000 cryptocurrencies networks [2]. Beyond cryptocurrencies, blockchain applications have also been emerging in various areas, such as finance, healthcare, military, and Internet-of-Things (IoT) networks [3].

The consensus mechanism is the core component of a blockchain network, which ensures that every participant agrees on the state of the network in such trustless en-

vironments. Early blockchain networks were developed based on the Proof-of-Work (PoW) consensus mechanism which relies on intensive computing processes to reach consensus. As a result, PoW has several disadvantages including huge energy consumption and low transaction processing capabilities. To overcome these problems, a new consensus mechanism has been developed recently, namely Proof-of-Stake, which replace the computing process of PoW with stake ownership proofs. As a result, PoS has negligible energy consumption and low transaction delay. Therefore, this mechanism is expected to become a cutting-edge technology for future blockchain networks and has the potential to be the backbone of blockchain-based frameworks for many applications, including mobile roaming data management, federated blockchain systems, and blockchain-based Metaverse applications.

Particularly, PoS consensus mechanism can be utilized for mobile roaming data management. With the popularity of IT technologies and smart devices, over 5 billion people have been subscribed to mobile services, generating a $1.03 trillion revenue globally in 2018 [4]. Although the number of subscribers and the revenues will continue to grow, mobile service providers have been facing several obstacles, especially for roaming services. Among them, fraud management is one of the biggest challenges for mobile service providers with over $32.7 billion annual loss throughout the world [5]. Roaming fraud exploits the inefficiency in managing data exchanges between two mobile service providers in order to use illegal free-riding services. To address this issue, the PoS consensus mechanism and blockchain technology can be leveraged to significantly reduce the data exchange delay in traditional roaming system, enhance the privacy of the mobile roamers, and reduce the dependency on middleman services.

In addition, the PoS consensus mechanism can also be leveraged to manage federated blockchain systems. Particularly, in a federated-blockchain system, there are multiple blockchains, and users in the system can transfer their assets to any

blockchain within. However, the widely used PoW consensus mechanism is not suitable for federated-blockchain systems, due to its huge energy consumption and very low transaction processing capabilities [3, 6, 7]. Moreover, the ability to transfer assets between multiple chains may lead to centralization to a single chain which poses a security threat to the other chains in the same federation [8]. Therefore, a secure and effective framework, which can address both security and performance issues for cross-chain transfers, is of urgent need for the future development of blockchain networks.

Furthermore, Metaverse has recently attracted paramount attention due to its potential for future Internet. However, to fully realize such potential, Metaverse applications have to overcome various challenges such as massive resource demands, interoperability among applications, and security and privacy concerns. To address the aforementioned challenges, blockchain has been considered to be a promising solution [9]. In particular, thanks to the smart contract mechanisms [10], blockchain can manage and automate complex interactions among various entities in Metaverse, such as Metaverse Service Providers (MSPs), users, and digital content creators. Moreover, with outstanding benefits of immutability and transparency [6], blockchain can play a key role in ensuring data integrity and protecting digital assets in Metaverse applications. Furthermore, with the asymmetric key and digital signature mechanisms [6], blockchain can enhance the users' privacy and anonymity [9]. However, blockchain, especially when applied in Metaverse, also faces various challenges [9]. Particularly, the PoW mechanism is not suitable for Metaverse applications due to the high delay and huge energy consumption. Moreover, scalability is another challenge of the PoW mechanism. Particularly, due to the requirement of the PoW mechanism, it is difficult to expand the blockchain network to improve its transaction processing capability. Consequently, these limitations are the major obstacles to the future implementation of blockchain-based Metaverse applications

that are expected to provide time-sensitive services to millions of users.

1.2 Literature Review and Contributions

This section first reviews the advantages and limitations of existing works in addressing the aforementioned issues. Then, the main contributions of this thesis are highlighted.

1.2.1 Literature Review

1.2.1.1 Mobile Roaming Data Management

Typically, a roaming fraud protection system consists of preventive and reactive layers [11]. The preventive layer prevents fraud perpetration by validating subscribers' authentication, auditing subscribers' credit, limiting services duration, and so on. Although these measures can help to mitigate roaming frauds, they have a negative impact on the Quality-of-Service provided to the subscribers, e.g., frequent validation and service limitation will lower customer satisfaction. The reactive layer typically consists of four main stages to detect and react to roaming fraud attacks. The roaming data, e.g., service records, exchanged between mobile service providers is first collected at the data collection stage and processed at the fraud detection stage to detect potential fraud cases [11]. Each case is then supervised manually in the supervision stage. The service usage is terminated if a fraud attack is confirmed at the response stage. Among these stages, data collection is often the bottleneck in the roaming fraud protection system. Techniques employed at this stage can only support data collection in near real-time with a limited number of subscribers, e.g., Fraud Information Gathering System [12], or shorten the data exchanging delay to 4 hours, e.g., Near Real Time Roaming Data Exchange [13]. Due to the sequential nature of the system, other stages cannot be activated if the data has not been collected. Consequently, although fraud attacks such as SIM cloning can also perpe-

trate locally in the HPMN, their consequences are much more severe in the roaming scenario due to the delay in data exchange, e.g., it takes up to 18 hours on average before an international roaming fraud attack can be stopped with the current system [14].

With outstanding performance in data integrity, decentralization, and privacy-preserving, blockchain has been emerging to be a secure and effective solution for data management in many decentralized networks. As a result, blockchain-based solutions for mobile roaming have been introduced recently by some organizations, e.g., IBM [15], Deutsche Telekom and SK Telecom [16], and Enterprise Ethereum Alliance [17], focusing on identity management, automating billing processes, and fraud prevention. In particular, these solutions focus on developing blockchain's asymmetric keys and digital signatures to manage subscriber identities and propose smart contracts to set up roaming pacts and automate billing processes. With enhanced identity management and automatic billing, fraud attacks can be significantly reduced. However, most of these solutions are still at the early stage of development and are facing several technical challenges.

Specifically, most of current blockchain-based data management systems often employ the PoW consensus mechanism, e.g., Bitcoin [18]. However, the PoW mechanism consumes massive amounts of energy, e.g., the Bitcoin network's energy consumption is higher than that of many countries [19]. Moreover, PoW-based networks often take a long time to reach consensus, e.g. one hour on average [6]. Thus, a new consensus mechanism, namely Proof-of-Stake (PoS), has been developed with significant advantages over the PoW mechanism, including reduced energy consumption and delay [6]. Recently, a PoS-based blockchain network, namely Bubbletone [20], has been introduced for mobile service providers to address roaming fraud problems. Using the PoS-based consensus mechanism and smart contracts, the blockchain-based Bubbletone system provides a general platform for various mobile service

provider-to-mobile service provider and mobile service provider-to-subscriber interactions in the roaming environment. Nevertheless, the consensus mechanism design is not thoroughly discussed in [20].

In addition, more users (e.g., mobile subscribers) participate in a PoS-based blockchain network means better the performance and security of the network are. Thus, it is important to incentivize more users to participate in the network. In current PoS-based blockchain systems, some stakes, e.g., network tokens, are paid to the users as a reward for consensus participation. However, a user with a few stakes is less likely to receive the reward. Moreover, some blockchain networks such as [20] impose a high stake requirement for consensus participation. Consequently, the stakeholders, i.e., subscribers, are inclined to join a stake pool (formed by mobile service providers) to earn more rewards. Furthermore, a stake pool can earn profits from the investments of the stakeholders by charging a portion of each stakeholder's reward [6]. As a result, the formation of a stake pool can be beneficial if it can incentivize more subscribers and mobile service providers to join the network. Therefore, the design of stake pool and network parameters has a significant impact on the performance of a blockchain network, yet studies on this topic are still limited. The stake pool formation in PoS-based blockchain networks was analyzed in our previous work in [6]. However, [6] only considers the investment strategies of the users while the stake pool's pricing policy is assumed to be static. In practice, however, the pool has to design its pricing policy to maximize the profits while attracting more investments from the stakeholders.

1.2.1.2 Ferderated-blockchain System

Sidechain technology was first introduced in [8] as a novel method to facilitate cross-chain transfers. Particularly, sidechain technology's mechanisms, such as two-way peg and Simplified Payment Verification (SPV) proof [8], enable a set of valida-

tors to verify and confirm transactions between different blockchains. Although this work paves the way for many research works and applications, the security and performance issues of sidechain are only briefly mentioned and not well investigated [8]. After the introduction of the sidechain technology, there have been several notable real-world applications such as PoA [21], Liquid [28], and RSK [29]. However, these applications are facing several challenges. In particular, the PoA approach relies on a fixed federation of 23 validators to validate the cross-chain transactions between the Ethereum [30] and several sidechains. This results in a low decentralization level for the consensus process. Moreover, these validators' identities are publicly known, making them easier to be targeted by attackers. Similarly, the Liquid approach [28] also relies on a federation to validate cross-chain transactions. Although these validators are not publicly known, they are chosen only by the network operators, and thus Liquid is not a public blockchain network. Moreover, Liquid is using a version of the PoW consensus mechanism which requires even more computational resources than Bitcoin (Liquid requires the validators to run a Bitcoin node in parallel with a Liquid node). Similar to Liquid, RSK employs a federation to validate transactions via a PoW-based mechanism. Although RSK is more decentralized, i.e., the federation in RSK is determined by public voting, RSK is still limited by the huge energy consumption of the PoW mechanism.

Different from the PoW mechanism, the PoS mechanism enables the blockchain participants to reach the consensus by proving tokens ownership. As a result, the PoS mechanism is much more energy-efficient and can achieve higher transaction processing speed compared to those of the PoW mechanism [3, 6, 7]. Due to those advantages, recent research works in the area of the sidechain technology have shifted towards the PoS mechanism. In [31], a protocol is developed for cross-chain transfers between a primary blockchain (main chain) and a secondary chain (sidechain). To validate the cross-chain transactions, the protocol relies on a set of certifiers

who are chosen by the main chain. A major advantage of the proposed protocol is the independence between the side chain and main chain in terms of security and operations. However, the security of this protocol is not guaranteed. In [32], the authors propose a sidechain system, in which both the sidechain and the main chain employ a PoS mechanism, i.e., Ouroboros. Unlike the previous works, this work focuses more on the security aspects of the sidechain technology, providing formal definitions and robust security analyses. However, the risk of centralization is not addressed. Similar to [31], the authors in [33] also introduce a cross-chain transfer protocol to allow interoperability between a main chain and a side chain. The cross-chain transfer protocol in [33] is proposed with formal definitions, and a consensus mechanism is also presented in a similar way as in [32]. However, the security of the protocol is not guaranteed, and the risk of centralization is also unaddressed. In [34], a PoS-based framework is proposed for a federated-blockchain system. Cross-chain transactions in this framework are processed by a group of validators. These validators are chosen based on their stakes once per day, and they are rewarded for their validation. However, this framework lacks formal security analysis, and it requires more than 66% of the network stakes to be controlled by honest users (Fedchain only requires 51%). In [35], a framework for federated-blockchain is developed based on the Tendermint consensus mechanism [26]. In this framework, cross-chain transactions are processed by a group of fixed validators. Such setting may result in a higher risk of centralization as these validators are predetermined and known by the whole network. In [36], a novel cross-chain transfer method is proposed. By requiring the transaction's sender and receiver to vote on a transaction, this method allows the transfers of assets among different parties on different blockchain network. In [37], a cross-chain commitment protocol is developed to enable asset transfers among different blockchains. Different from previous work, this protocol consider the cases where users need to send their transactions on time to a specific smart contract to

transfer their assets. A common limitation of both [36] and [37] is that the risk of centralization is not considered.

To the best of our knowledge, the risk of centralization in federated-blockchain systems has not been addressed in any previous work. Specifically, the ability to transfer tokens between chains may lead to situations where the users centralize to a single chain in the system, e.g., the chain which gives the highest rewards for consensus participation. Such centralization of tokens and users may have negative impacts on the security and performance of the other blockchains in the same system. The reason is that the state of each PoS blockchain is determined by the majority of stakes (tokens), i.e., users who have more stakes will be very likely to be selected to add new blocks. Consequently, it is easier for attackers to target the blockchains that have fewer tokens. This can significantly impact these blockchains' security. Furthermore, since the cross-chain transfer requires the confirmation of transactions in both the originating and destination chains, the centralization of stakes also reduces the overall system performance.

1.2.1.3 Blockchain-based Frameworks for Metaverse Applications

As Metaverse is an emerging topic, applications of blockchain in Metaverse are still very limited. There are just a few recent works [45–47] focusing on this topic. Specifically, in [45], the authors propose a blockchain-based secure mutual authentication scheme for Metaverse environments. In this approach, the MUs need to send their pseudo-identity, personal information, and public key to a central authority to verify. If the verification is successful, the central authority stores the MUs' identities and public keys in a public blockchain for Metaverse applications to query. Similarly, the authors in [46] develop a blockchain-based framework for Metaverse to manage MUs' identities and transactions. Particularly, the proposed framework is composed of four parts, namely New User Engine, Transaction Centre, Authenti-

cator Engine, and Repo. In this framework, the New User Engine is responsible to provide new MUs with blockchain addresses. MUs can then send their transactions to the Transaction Centre to process, and the Authenticator Engine's responsibility is to validate the MUs' identities and transactions. If the transaction is successfully validated, it will be recorded in the Repo (which is a distributed ledger) along with the resulting change in MUs' accounts. In [47], the authors propose a blockchain-enabled framework for Metaverse service management. Particularly, in the proposed framework, the mobile network operators can offer their services to MUs with different service level agreements and prices. The MUs can then choose one of the options based on a proposed utility function with a trade-off between service quality and prices. In this framework, the blockchain serves as a platform to verify MUs' identities, and the blockchain tokens are used as the currency for payment.

From the above, we can observe that [45–47] only utilize conventional blockchain technology for managing MUs identities and transactions without taking into account specific challenges of Metaverse, such as the huge resource demand or the associated scalability issues. To the best of our knowledge, our proposed MetaShard framework is the first in the literature that can encourage MUs to contribute resources to the Metaverse and blockchain network as well as address the scalability issue of blockchain.

In [49], the authors propose a sharding protocol for public blockchain networks. Although the protocol is proven to be secure, it utilizes PoW to authenticate the consensus participants' identities. Another PoW-based sharding scheme is proposed in [120], where nodes with high computing power in the system can participate in several shards simultaneously. Similar to [49], this scheme requires consensus participants to solve a PoW puzzle to become validators, and shards' security is proven based solely on the number of consensus participants. However, since the consensus participants are required to solve a PoW puzzle, the adversary can split

their computing power to simultaneously solve different puzzles and thus able to gain more slots. As a result, the computing power distribution needs to be taken into account, but it is not discussed in [49] and [120]. Another PoW-based sharding scheme is proposed in [50]. Although the scheme's security is proven, it relies on PoW, which is unsuitable for Metaverse due to the high delay and huge energy consumption.

To address the limitations of PoW, other sharding protocols were developed with energy-saving alternative ways to select consensus participants. For example, in [51], a sharding protocol is developed based on Byzantine Fault Tolerance (BFT) and Trusted Execution Environment (TEE). Particularly, the consensus participants need a special type of hardware to ensure the TEE. A similar approach that relies on TEE is proposed in [121]. Particularly, a sharding scheme is developed that utilizes two separate blockchains to decouple the transaction recording and consensus processes. Similar to [51], the proposed scheme relies on TEE, and thus it also requires special hardware to participate in the consensus process. Although the schemes in [51] and [121] can enhance the security of the network, the hardware requirement makes them much less attractive to public users, especially MUs who already need a lot of computing power for AR/VR rendering. In [52], the authors develop a sharding scheme based on Practical Byzantine Fault Tolerance (PBFT). Although the security of the protocol is proven, how to select the consensus participants is not discussed. Moreover, similar to [49], this protocol relies on the number of consensus participants, without taking into account the ability of the adversary to create many identities to gain more consensus participants slots. In [53], a reputation-based sharding scheme is developed. Particularly, the consensus participants are selected based on their reputation scores stored in a separate blockchain. Then, the selected consensus participants execute a BFT-based protocol for each shard's consensus process. However, the adversary in this case can also create many identities to

increase the number of consensus participants it controls, as the reputation is based solely on previous behaviors. In [54], a BFT-based sharding protocol is developed. However, similar to [49] and [53], the protocol relies on the number of consensus participants, which can be adversely impacted by the adversary. In [55], a dynamic sharding protocol is proposed in which the consensus participants are selected via smart contracts. Moreover, to mitigate Sybil attacks, the proposed protocol requires that each consensus participant must come from a different IP address. Nevertheless, this still cannot prevent the adversary from influencing the selection process, as IP addresses can be fake.

From the abovementioned approaches, we can observe that they rely on the PoW consensus mechanism which is inappropriate for Metaverse due to the huge energy consumption and large delay. In contrast, our proposed PoE consensus mechanism is much more energy-efficient. Moreover, the security of these approaches relies on the number of consensus participants without considering that this number can be unfairly affected by the adversary. On the contrary, our proposed approach considers the MUs' engagement instead of the number of participants, thereby enhancing the security and robustness of the system against Sybil attacks.

1.2.2 Contributions

From the above literature review, we can observe that developing the PoS mechanism requires thoughtful design and consideration to overcome significant challenges for different blockchain network architectures. In single-blockchain systems, forming stake pools might significantly increase the blockchain's centralization level, posing serious threats to the network's security. Moreover, in federated-blockchain systems, the transfers of assets among the blockchains within might centralize users to a single blockchain, leaving the other blockchains vulnerable to attacks. Similarly, in sharding-based blockchain networks, the division of the blockchain into shards

might weaken the blockchain's security. This thesis develops PoS-based frameworks and proposes solutions for various data management applications to address the abovementioned issues. The detailed contributions of this thesis are presented in the following.

1.2.2.1 Mobile Roaming Data Management

The main contributions of this thesis regarding the application of the PoS consensus mechanism for mobile roaming data management are briefly summarized as follows:

- We propose BlockRoam, an effective blockchain-based roaming service management system to provide a transparent, secure, and automatic platform for data exchanging between the mobile service providers. In particular, by employing the PoS consensus mechanism, BlockRoam can achieve a delay of less than 3 minutes, which is much lower than the 4-hour delay of traditional roaming management systems. In addition to the reduced latency, BlockRoam can automate various roaming processes thanks to smart contracts [10], and thus roaming frauds can be significantly reduced. Moreover, the mobile service providers often rely on Data Clearing Houses (DCHs) to process and exchange data, which incurs additional costs [11]. In our proposed system, the transactions are stored in the blockchain and processed by smart contracts, and thus the service fees for DCHs can be eliminated. Furthermore, the privacy and security of the subscribers in BlockRoam are significantly enhanced thanks to the blockchain's advanced cryptography techniques [3].

- We analyze existing PoS-based consensus mechanisms [21–27] to show that they are not suitable for roaming management due to their limitations in terms of security and performance. Therefore, we develop a consensus mechanism for BlockRoam, which can meet strict security requirements, mitigate a wide

variety of blockchain attacks, and achieve a much better performance in terms of transaction confirmation time compared to those of existing mechanisms.

- We introduce an economic model based on the Stackelberg game theory in order to jointly maximize the profits of the stake pool and the stakeholders. By analyzing utility functions of the stake pool and stakeholders, we develop a Mixed Integer Linear Programming model to find the Stackelberg equilibrium of our proposed game. We also propose an effective method that can guarantee to achieve the unique equilibrium for this game. The proposed economic approach can help to maximize the profits of the stake pool and the stakeholders, as well as attracting more investment and improving BlockRoam's security and performance.

- Extensive simulations have been performed to evaluate the performance of our game theoretic model. Particularly, we simulate the game to show that the model can brings additional benefits for the stake pool and the stakeholders. Moreover, we also examine the influence of important parameters on the outcome of the game. Furthermore, adversarial attacks scenarios are also simulated to show that the proposed economic model can help to improve the network's security and performance by attracting more investments to the network. These results are especially crucial in designing appropriate parameters (e.g., total network stakes, pool fees, and rewards) to improve BlockRoam's security and performance.

1.2.2.2 Ferderated-blockchain System

The major contributions of this thesis regarding the application of the PoS mechanism for federated-blockchain systems can be summarized as follows:

- Propose FedChain, an effective and secure framework for cross-chain transfers

in federated-blockchain systems. Particularly, Fedchain facilitates two-way transfers of assets between any two different chains in the system by utilizing the sidechain technology.

- Develop a novel PoS-based consensus mechanism for the individual blockchains in FedChain that can satisfy the persistence and liveness properties [38], prevent many blockchain-specific attacks, and achieve a more desirable transaction confirmation time compared to other mechanisms such as [21–26, 39].

- Develop an incentive mechanism using a Stackelberg game model [40] for FedChain in order to address the problem of centralization in the sidechain technology, provide additional benefits for the users, and enhance FedChain's security and performance. To the best of our knowledge, this is the first framework addressing the risk of centralization in federated-blockchain systems. Furthermore, we can prove the uniqueness of the Stackelberg equilibrium and find the exact formula for this equilibrium. These results help the stakeholders to determine their best investment strategies and the chain operators to design the optimal incentive policy.

- Perform extensive simulations to evaluate the system performance of FedChain. The simulation results then confirm the analytical results and show that FedChain can help the users to maximize their profit and the blockchain operators to determine their optimal blockchain parameters to improve the system's security and performance.

1.2.2.3 Blockchain-based Frameworks for Metaverse Applications

The major contributions of this thesis regarding the application of the PoS mechanism for Metaverse applications can be summarized as follows:

- We propose MetaShard, a novel sharding blockchain framework for Metaverse

applications that can manage MUs identities, digital assets, and transactions, facilitate various interactions between the MSP and the MUs, leverage MUs resources contributions, encourage more MUs to the Metaverse, and improve scalability while ensuring the network security.

- We develop PoC, a new consensus mechanism that can encourage and reward MUs' data and computing resources contribution, thereby alleviating the massive resource demands and incentivizing MUs to be more engaged in the Metaverse.

- We propose a sharding management scheme to improve the scalability of MetaShard. We then formulate a score allocation optimization problem to ensure and improve the security of the shards. To further enhance the efficiency of MetaShard, we develop a lightweight hybrid approach to efficiently solve the problem, thereby allowing frequent shards reconfiguration and improving the network security.

- We conduct extensive simulations to evaluate the performance of our proposed approach. The results show that, compared to existing approaches, our proposed lightweight approach can obtain solutions that are more secure and with higher throughput (up to 66.6%). Moreover, the running time of the proposed approach is much shorter (up to 30 times faster). Furthermore, we study the impacts of important parameters on the system and show that the proposed approach is more robust to stronger adversaries.

1.3 Thesis Organization

The rest of this thesis is organized as follows.

- Chapter 2: This chapter provides the fundamental background of blockchain technology. In particular, Section 2.1 introduces the fundamental concepts of

blockchain technology. Section 2.2 provides a comprehensive views of blockchain consensus mechanisms, including PoW, Proof-of-Concept, and PoS. Finally, Section 2.3 provides detailed information about the PoS consensus mechanism.

- Chapter 3: This chapter presents our proposed PoS-based framework for mobile roaming data management. Specifically, Section 3.1 describes the system model. Section 3.2 discusses BlockRoam's consensus mechanism in details. The economic model is proposed in Section 3.3. Evaluation results are then discussed in Section 3.4. Finally, conclusions and future works are given in Section 3.5.

- Chapter 4: This chapter introduces our proposed PoS-based frameworks for federated-blockchain systems. In particular, Section 4.1 describes the proposed FedChain framework, and Section 4.2 proposes FedChain's consensus mechanism. Then, Section 4.3 introduces the problem formulation. After that, the evaluation results are discussed in Section 4.4. Finally, conclusions are drawn in Section 4.5.

- Chapter 5: This chapter present our proposed PoS-based framework for Metaverse applications. Particularly, Section 5.1 presents MetaShard's system overview. The proposed PoC consensus mechanism and sharding management scheme are presented in detail in Section 5.2. Section 5.3 presents the sharding management problem and our proposed lightweight approach, and its performance is evaluated in Section 5.4. Finally, conclusions are drawn in Section 5.5.

- Chapter 6: This chapter outlines the conclusion and future research directions of this thesis.

Chapter 2

Background

This thesis aims to develop PoS-based solutions for smart data management in mobile networks. In the following, the fundamentals of blockchain technology and consensus mechanisms are first provided. Then, the PoS consensus mechanisms are discussed in details.

2.1 Fundamental Background and Applications of Blockchain Networks

2.1.1 Blockchain Networks

As illustrated in Fig. 2.1, in the blockchain, transactions (data) are stored in blocks which form an ever-growing sequence (chain) shared among participants in the network. Transactions are the fundamental units of a blockchain. For example, when Alice wants to send money to Bob, she creates a transaction (Tx1 in Fig. 2.1) which consists of her address as the input, her digital signature to verify that this transaction is made by her, the amount of money to be sent, and Bob's address as the output. Alice then broadcasts this transaction to the network. A miner, i.e., a consensus participant, after receiving the transaction will validate and include Alice's transaction, along with other transactions received from other users, into a block. If the block is mined successfully, the miner (e.g., Miner 3 in Fig. 2.1) will broadcast the block to the network for other nodes to verify the mined block. If this block (e.g., Candidate Block 3 in Fig. 2.1) is verified successfully and identified to be the first block mined after the last block in the chain, it will be integrated into the

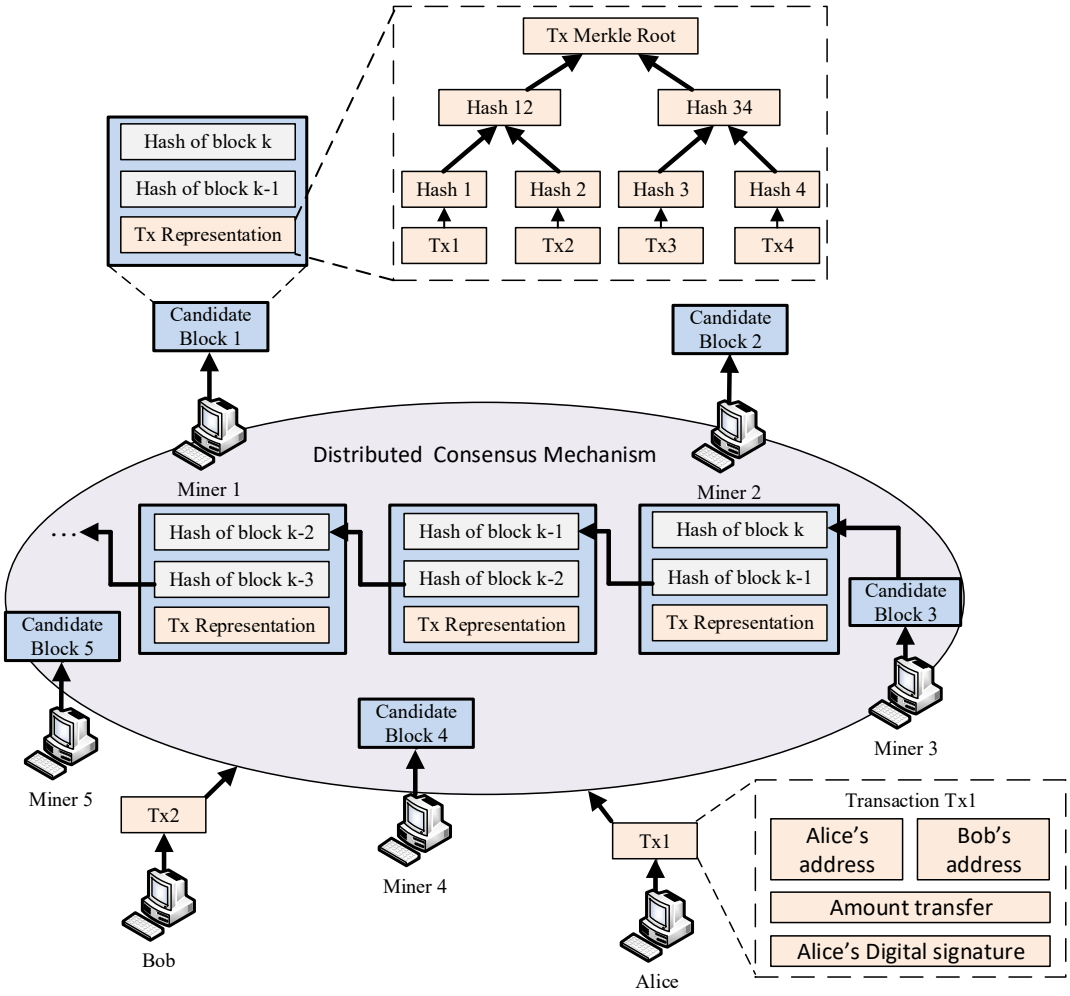


Figure 2.1 : An illustration of a blockchain network.

chain and marked as the latest block in the chain. Besides the transactions, a block also contains a hash pointer created by hash functions to map all the block contents to the hash pointer. The main feature of the hash functions is to ensure that the chain is tamper-evident. It means that any change in the previous data will result in a different hash value in the next block, and it can be traced back to the genesis block, i.e., the first block of the chain. A block can also contain additional data depending on requirements of different consensus mechanisms. To reduce storage space, the transactions in a block can be stored in the form of a Merkle tree [1].

2.1.2 Benefits and Applications

Although blockchain technology attracts a lot of attention due to the successful implementation of cryptocurrencies, its benefits extend far beyond. The key benefits of blockchain technology are as follow:

- *Decentralization:* Blockchain networks are not controlled by a central controller. Thus, they do not have any single point of failure. Instead, all the nodes reach the agreement on the state of the network by participating in the distributed consensus mechanisms.

- *Transparency:* Data stored in a blockchain is visible to all network participants.

- *Immutability:* Once the data are stored in the blockchain, it is extremely difficult to be altered. Moreover, thanks to the distributed consensus mechanisms, the network can achieve consensus on the data even in a trustless environment.

- *Security and Privacy:* Using cryptographically secure mechanisms, the privacy and security of the network participants can be significantly enhanced. Users in the network use a pair of public and private keys for identification and verification. When a user makes a transaction, a digital signature is used, which can be easily verified but impossible to forge.

Given the aforementioned outstanding benefits, blockchain technology has many applications in a number of areas. Some major applications of blockchain technology are as follow:

- *Cryptocurrencies:* Cryptocurrencies, e.g., Bitcoin [18], Ethereum [30], Cardano [56], are the most famous applications of blockchain technologies. With

high value and daily trade volume, cryptocurrencies can be utilized for various financial applications, such as digital assets and online retail.

- *IoT network:* Its anonymity and security make blockchain applicable to many IoT networks, e.g., Internet-of-Vehicles [57–60], energy trading [61,62], electric vehicle charging [63], and smart home [64], for operations management, trading automation, and security enhancements.

- *Healthcare:* Blockchain technologies have been adopted by many healthcare systems to enhance the privacy of patient data [65], improve interoperability across devices [66], and maintain an immutable decentralized database of medical records [67].

- *Military:* Blockchains have the potential to be applied in various military operations, such as enhancing data integrity in supply chain management, ensuring transparency in equipment management [68], and providing a distributed and decentralized database for military intelligence [69].

- *Service providers:* Blockchain networks have also been employed by many service providers. Blockchain technology can support automatic payments, contents distribution, and services delivery [70, 71].

2.2 Consensus Mechanism

Nodes in a blockchain network can be faulty, performing arbitrary or malicious behaviors, or possessing misinformation due to connection latency, i.e., Byzantine failures. The consensus mechanism is thus the core component of a blockchain network, which ensures that every participant agrees on the state of the network in such trustless environments. The consensus mechanism also governs other operations of the network, such as transaction adding and incentivizing the participants to behave properly.

2.2.1 Proof-of-Work

Early blockchain networks were developed based on PoW mechanism. Generally, the nodes in a PoW-based blockchain network reach consensus by participating in a solution searching process, where each node must find a nonce for its proposed new block. When the nonce, the previous block's hash, and the transactions in the new block are used as the input of the hash function, e.g., SHA-256, the hash function output must be in a target range so that the block can be accepted. Due to the property of the hash function, the nonce can only be found by repeatedly trying different nonce values until the output is within the target range. When a participant finds the nonce, it will broadcast the block along with the transactions to other nodes. Then, if the new block is verified and determined to be the first block mined after the last block in the chain, it will be integrated into the current chain and become the latest block in the chain.

In PoW, the participants compete with each other to be the first to find the correct nonce. This solution searching procedure can be considered to be a weighted random coin-tossing process where a participant with a higher hash rate (computational power) might have higher chances to be the block winner (leader) who can receive the reward. The probability p_i that participant i is selected to be the leader in a network of N participants is

$$p_i = \frac{c_i}{\sum_{j=1}^{N} c_j}, \tag{2.1}$$

where c_i is the hash rate of participant i. This computation leads to the large amount of energy consumption for blockchains using PoW consensus mechanisms, as the participants try to increase their hash rates to have a higher chance to be the leader and receive rewards. Moreover, since participants with low hash rates have very low chances to win a block and receive rewards, they often join mining pools to have more opportunities to get revenues. A mining pool consists of participants

who want to collaborate by contributing their computing resources to the pool. In this way, mining tasks will be distributed to the miners, and due to huge computing resources, mining pools often get much higher opportunities to win a new block than individuals. While joining a mining pool provides more stable incomes, the nodes in the pool often do not contribute to the transaction validation and propagation since they only perform the nonce search process in a specific range. Thus, mining pools have been dominating processes making new blocks in most of current blockchain networks. For example, the top five mining pools control up to 62.7% total hash rate of the Bitcoin network [72]. This is the most serious issue of PoW-based blockchain networks because it is against the decentralized spirit of blockchain technology. Another issue of PoW protocols is delay. In a PoW-based blockchain network, when a block is added to the chain, there is still a possibility that this block will not be included in the main chain for several reasons, e.g., network delay causing several versions of the chain or two participants finding two blocks simultaneously. This possibility decreases exponentially as the block is deeper in the chain. Therefore, a block is considered to be finalized only when it is a certain k, usually six blocks deep in the chain. This delays the transaction confirmation significantly. Moreover, PoW mechanism is also vulnerable to 51% attack. In particular, if a single party controls more than 51% of the network's total computational power, they can spend their coins multiple times (in cryptocurrency networks) or prevent other transactions by adding conflicting blocks to the chain. While 51% attacks might not be a serious problem for large blockchain networks, the newly established networks with small and limited total computational power are especially vulnerable [3].

2.2.2 Proof-of-Concepts

Based on the PoW framework, the Proof-of-Concepts (PoX) consensus mechanisms have been developed with two major aims: to replace the PoW solution

searching with useful calculations and to improve the performance of PoW in terms of security, incentives, and resource usage. To make better use of the computational resource, several consensus mechanisms require the participants to solve practical mathematical problems such as searching for three types of prime number chains in Primecoin [73], solving matrix product problems in Proof-of -Exercise [74], and calculating useful functions in Proof-of-Useful-Work [75]. Other PoX consensus mechanisms are designed for distributed data storage service such as Permacoin [76], KopperCoin [77], and Filecoin [78]. Generally, these consensus mechanisms divide the data files into segments and distribute them to multiple participants in the network. To participate in the mining process, the nodes have to provide proofs of storage, and the more storage volume a node offers, the better chances it is selected to be a leader.

Other PoX consensus mechanisms have been developed with the aim to improve the performance of PoW. The problem of mining pool formation is addressed by designing nonoutsourceable puzzles to replace the PoW solution searching process, such as in [79] and [80]. In these networks, the solution searching processes financially disincentivize mining pools formation because the node who found the solution can steal the reward. Other consensus mechanisms have been developed to reduce the computational requirement of PoW. The Spacemint [81] network employs a Proof-of-Space protocol, in which the consensus nodes must provide proof of storage when participating in the solution searching process. Different from [76–78], the stored files are not useful and only serve as proofs. Nevertheless, this is still beneficial as storing a large file consumes negligible energy compared to nonce searching. In Proof-of-Human-Work protocol [82], the Completely Automated Public Turing-Test to tell Computers and Humans Apart (CAPTCHA) is employed to involve human activities and reduce computational requirements in the solution searching process.

2.2.3 Proof-of-Stakes

The first PoS network, Peercoin [83], was developed as a PoX consensus mechanism with the aim to reduce the computational requirements of PoW. Participants with higher coin age, i.e., product of network tokens and their holding time, have higher chances to be selected. Specifically, each node in Peercoin solves a PoW puzzle with its own difficulty, which can be reduced by consuming coin age. In the more recent PoS networks, the solution searching is completely removed, and the block leaders are no longer selected by computational power. Instead, they are selected based on the stakes that they are holding.

With the stake-based leader selection process, a node's chance to be selected to be a leader no longer depends on its computational power, and thus energy consumption of PoS mechanisms is significantly reduced compared with that of PoW. Moreover, the block generation and transaction confirmation speeds are kept at relatively low constant rates by the PoW networks to ensure security because there are many different blocks proposed by the miners. In contrast, since only one block is made in each round of PoS mechanisms, the block generation and transaction confirmation speeds are usually much faster, and thus PoS mechanism starts to become popular recently.

2.2.4 Hybrid consensus mechanisms

Aiming to reduce the high resources consumption of PoW, early PoS-based protocols are developed from standard PoW consensus mechanisms, and thus still incorporate some PoW elements, which makes hybrid PoW-PoS protocols. The Peercoin protocol discussed above can be considered to be a hybrid consensus mechanism, which utilizes PoS to reduce the high computational requirement of PoW. Another typical example is the Proof-of-Activity (PoA) protocol [21], which employs the PoW to create empty blocks and the PoS to verify blocks and add transactions. Based on

Table 2.1 : Consensus Mechanisms Comparisons

	PoW	PoS	Hybrid
Leader selection	Based on hash rate	Based on stake	Depends on variant
Energy consumption	Significant	Negligible	Medium to negligible
Hardware requirement	High	None	Medium to none
Block generation speed	Slow	Fast	Medium to high
Transaction confirmation speed	Slow	Fast	Medium to high
Applications	Bitcoin, Ethereum, etc.	Cardano, Algorand, etc.	Casper, Peercoin, etc.

the PoA, the Snow White protocol [84] was developed in which the main difference is that PoS is employed first to choose a number of candidates. These candidates then compete with each other via the PoW to create blocks.

Other hybrid consensus mechanisms often elect a committee to verify blocks and confirm transactions. The Hybrid Consensus protocol periodically elects a committee based on the hashes of previous blocks to add and confirm transactions. The Peercensus protocol [85] selects committee members from the previous block creators. Different from the Hybrid Consensus protocol [86], the committee is responsible for both transaction adding and block confirmation in the Peercensus protocol.

The hybrid protocols inevitably inherit the strength and weakness of the consensus mechanisms that they are created from to some extent. Typically, the energy consumption of these consensus mechanisms is lower than that of the PoW, but it is still higher than that of pure PoS protocols. In addition, the block generation and transaction confirmation speeds are also higher than those of PoW due to their usage of PoS and voting committee. The major differences between the protocols can be found in Table 2.1.

2.3 Proof-of-Stake-based Mechanisms

2.3.1 Proof-of-Stake: Fundamental Background

PoS protocols were developed as energy-saving alternatives to PoW. Instead of computational power resources, leaders are selected based on their stakes, i.e., contributions to the blockchain network. Particularly in the PoS consensus mechanism, the stake of a node is the number of digital tokens, e.g., coins in cryptocurrencies, that it holds or deposits. Instead of consuming a lot of energy for the searching process as in the PoW, a leader will be selected based on its stakes to perform mining process and add a new block to the chain as illustrated in Fig. 2.2. To simulate the stake-based leader selection process, the Follow-the-Satoshi (FTS) algorithm has been adopted in many PoS-based blockchain networks such as Cardano, Sp8de, and Tezos. In these networks, all the tokens are indexed. The FTS algorithm is a hash function that takes a seed (i.e., a string of arbitrary length such as the previous block's header or a random string created by some other selected nodes) as the input. The FTS algorithm then outputs a token index. Using the index, the algorithm searches the transaction history to find and select the current owner of that token to be the leader. Therefore, the probability p_i that node i is selected to be the leader in a network of N participants is

$$p_i = \frac{s_i}{\sum_{j=1}^{N} s_j}, \tag{2.2}$$

where s_i is the stake of participant i. This means that the more stake a node holds, the higher chance it is selected to be the leader.

Besides the advantage of low energy consumption, the PoS mechanisms have faster transaction confirmation speed than that of the PoW mechanisms. In a blockchain network, the confirmation of a transaction depends on two main factors, namely transaction throughput and block confirmation time. The transaction

throughput is the number of transactions per second Tx/s a network can process, which is vital to the performance of the network especially when there are many pending transactions. Tx/s can be calculated by

$$Tx/s = \frac{Block_{size}}{Tx_{size} \times Block_{time}}. \tag{2.3}$$

For example, the Bitcoin network has $Block_{size} = 1MB$, $Tx_{size} = 250bytes$, and $Block_{time} = 600s$, so it can process around 7 transactions per second. The Tx/s determines how quickly a transaction is added to the chain, whereas the block confirmation time dictates how fast the transaction is confirmed after it is added. The block confirmation time depends on $Block_{time}$, i.e., the average time it takes for a new block to be added to the chain, and the finality of the consensus mechanisms. In the Bitcoin network, a transaction usually has to wait for $k = 6$ blocks before it can be confirmed, so the average confirmation time is $k \times Block_{time} = 3600s = 1hr$. Typically in PoS networks, the block size is larger, and the block time is much shorter, thus the transaction throughput is much higher, e.g., up to $875Tx/s$ in [24]. Moreover, some PoS networks can achieve immediate finality, i.e., $k = 1$, so their transaction confirmation time is significantly shorter, e.g., down to 1 second in [26]. Similar to PoW, some PoS protocols such as [21–23, 25, 83, 84] adopt the longest chain rule which ensures that when there are multiple versions of the chain (forks), the honest participants will only adopt the longest fork. As a result, the finality in these protocols is delayed. In contrast, protocols such as [24, 26] can achieve immediate finality by voting to confirm block after each round.

The security of PoS protocols depends on various factors. Among them, network synchrony is crucial to the security of many PoS protocols because the leader selection processes are simulated by voting rounds, where the voters send their votes to other participants. Since the network cannot guarantee that all the messages are properly sent in practice due to network delay and connection complexity, network synchrony has to be taken into account when considering the protocol's security.

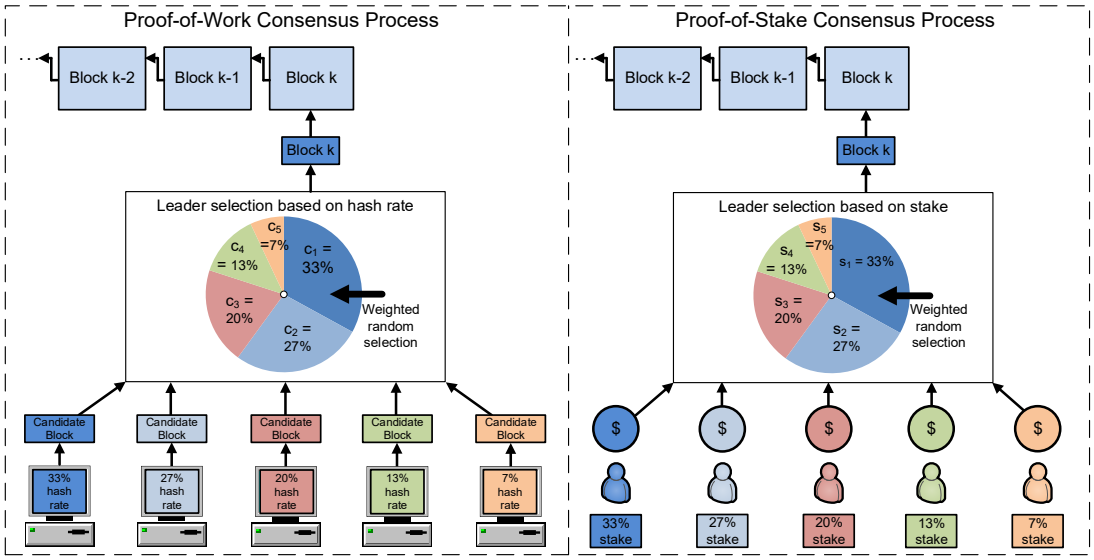


Figure 2.2 : PoW and PoS consensus mechanisms comparison.

Some PoS protocols are proven to be secure as long as the network is partially synchronous, where messages sent will reach their destinations within a certain time limit, or asynchronous, i.e., messages may not reach their destinations.

Apart from the network synchrony, the incentive mechanism is also vital to the security of a PoS consensus mechanism. On the one hand, the reward scheme has to incentivize consensus participation by rewarding block creators and validators. On the other hand, it also has to penalize malicious behaviors and prevent various attacks that specifically target PoS, such as the attacks that involve creating a large number of blocks because it is much easier to create blocks in PoS. The PoS protocols often have both reward and penalty mechanisms, such as [22, 25, 26].

Below, we discuss in more details some emerging PoS-based protocols which have been widely implemented in practice, namely Ouroboros, Chains-of-Activity, Casper, Algorand, and Tendermint. Their core components, namely the consensus processes, are illustrated in Fig. 2.3, and the protocols are then compared in Table 2.2.

2.3.2 Ouroboros

Ouroboros [23] is a pure stake-based protocol, which employs a dynamic committee selected based on the stake distribution. The protocol divides time into epochs. In each epoch, the committee members participate in a 3-phased coin-tossing protocol to create the seeds for the FTS algorithm. The FTS algorithm then outputs some coin indices, and the current owners of the chosen coins are selected to be the leaders and become the committee members in the next epoch. Different from PoW protocols, in Ouroboros the leaders only create empty blocks. The input endorsers are responsible for confirming and adding the transactions to the blocks. The block rewards are shared between the committee members, the leaders, and the input endorsers to encourage participation in the consensus process. A stake delegation mechanism, i.e., stakeholders can delegate their right to participate in the committee, is also incorporated to incentivize small stakeholders to contribute to the consensus processes.

Under a partial synchrony network assumption, Ouroboros is proven to be safe when the adversary controls strictly less than 51% of the total stake. Since partial synchrony cannot be guaranteed in practice, Ouroboros considers the asynchronous nodes to be a part of the adversary nodes. The dynamic stake distribution is also taken into account and incorporated into the adversary's stake. It was also shown in [23] that the seed creation process cannot be biased by the adversary, and thus grinding attack, i.e., the block proposers may try different block's hash in the attempt to influence the next leader selection round, is mitigated. The attacks where the adversary secretly builds alternative forks to later overtake the main chain, e.g., nothing-at-stake attack and long-range attack, are mitigated by having only one designated leader in each round. The incentive mechanism is also analyzed in the paper, and being honest is proven to be a δ-equilibrium strategy for the participants. However, the protocol still cannot withstand 51% attacks, and bribe attacks are not

formally discussed.

Ouroboros has the advantages of low transaction confirmation time, e.g., 2 minutes [56], and high transaction throughput, e.g., around 257 Tx/s [87]. Moreover, because only the chosen leaders can create blocks in Ouroboros, energy consumption is negligible compared with those of PoW-based networks. Another advantage of Ouroboros over many protocols, including some PoS protocols, is that it has formal definitions and strong theoretical background to support its security and incentive compatibility. As a result, Ouroboros has been adopted by several cryptocurrencies, such as Cardano* and Sp8de†.

2.3.3 Chains-of-Activity

Similar to Ouroboros, in the Chains-of-Activity (CoA) protocol [22], the leader is selected by the FTS algorithm. However, the seed for the FTS algorithm is different from Ouroboros. In CoA, the chain is divided into groups of blocks of length l, and time is divided into epochs such that in each epoch, exactly l blocks are added to the chain. The hash of each block is used to determine a seed of that block. The seeds of all the blocks created in an epoch are combined to seed the FTS algorithm for determining the next epoch's leaders. At each round in an epoch, a leader is selected by the FTS algorithm to collect transactions and create a new block. The selected leader has to make a deposit before creating a block. The block reward can be claimed by the leader if the block is created properly, and the deposit will be confiscated in cases of malicious behavior. The CoA protocol also introduces the checkpoint blocks, i.e., the blocks that extend the chain by exactly T blocks, to solidify the chain and prevent long adversarial forks from taking over.

The CoA protocol is proven to be secured against a number of attacks. By seed-

*https://www.cardano.org

†https://sp8de.com

ing the FTS algorithm with hashes from the previous group of blocks, the protocol can effectively mitigate grinding attacks. Similar to the Ouroboros protocol, there is only one designated leader to create a block in each round. Thus, nothing-at-stake and long-range attacks are mitigated. Long-range attack is an attack that specifically targets the protocols where the leaders are determined before their designated epoch. In these protocols, after realizing that they are going to be leaders in the next epoch, the stakeholders might sell their stakes, so that they can behave maliciously without consequences. With the checkpoint blocks mechanism, every block from the first block to the second most recent checkpoint block can never change, and thus long-range attack is mitigated by the CoA protocol. The deposit scheme helps to prevent double-spending attacks, where the attackers create conflicting blocks to revert confirmed transactions, and bribe attacks, where the attackers bribe the leaders to conduct double-spending attacks.

In the CoA protocol, there is only one block created at each round, and thus energy consumption is small compared with that of the PoW mechanisms. CoA also has low transaction confirmation time, around 6 minutes [88], and high transaction throughput, $40Tx/s$ [89]. However, the incentive compatibility is not formally analyzed, and the network synchrony and adversary toleration threshold, which is crucial to the network security, are completely ignored in the paper. The cryptocurrency Tezos[‡] is designed partially based on the CoA protocol.

2.3.4 Casper

The Casper protocol [25] was developed by the Ethereum network in an attempt to ease the transition from the current PoW protocol to a pure PoS protocol, i.e., it can work on top of existing PoW protocols. In this context, Casper does not interfere with the leader selection process. Instead, it employs a dynamic committee,

[‡]https://tezos.com

which votes via a Byzantine-Fault-Tolerance (BFT) protocol to justify the checkpoint blocks at every fixed interval, e.g., every 100 blocks. Every block up to the second latest justified checkpoint is considered to be finalized. To join the committee, a validator has to make a deposit to gain voting right proportional to that deposit, which will be slashed for malicious behaviors.

Casper is proven to be secure as long as 2/3 of the voting power is controlled by honest validators in a partially synchronous network. By incorporating a withdrawal delay, i.e., the validator has to wait for a long period of time before the deposit can be withdrawn, the protocol can handle dynamic stake distribution and long-range attack. The other security issues are implied to be handled by the underlying chain.

Another advantage of Casper is that it can work on top of other PoW protocols, thereby providing additional security to the underlying chain. However, Casper's performance relies on the underlying PoW mechanism. In addition, another issue is that the incentive mechanism is undefined in the paper, despite its key roles in ensuring the participants follow the protocol properly. Ethereum[§] has been developing Casper, and it is expected to be implemented for future PoW-based blockchain protocols.

2.3.5 Algorand

Similar to Ouroboros, the Algorand [24] protocol also operates under a committee. However, the protocol uses a cryptographic sortition mechanism instead of the FTS algorithm to select the leaders and committee members based on the stake distribution. The cryptographic sortition [24] is a Verifiable Random Function (VRF) that takes a private key of a consensus node and a seed as inputs and outputs a hash and a proof for public verification. Each consensus node is assigned a range of hash values proportional to its stake amount. If the hash is within a node's assigned

[§]https://www.ethereum.org

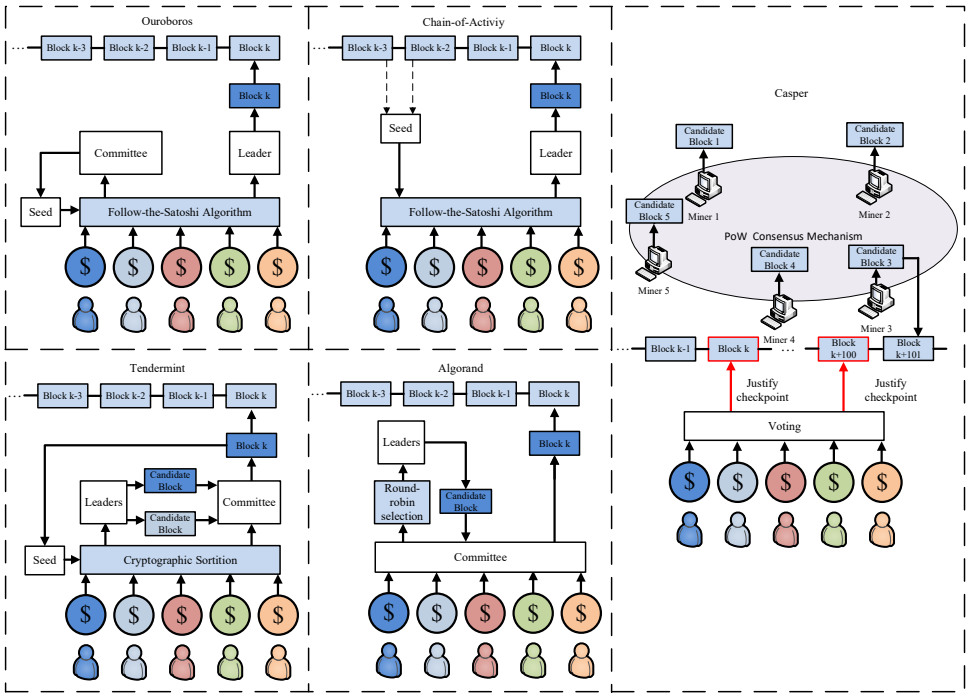


Figure 2.3 : Illustrations of several PoS consensus processes.

range, the node is selected, and thus the node's chance to be selected is directly proportional to its stake amount. The main difference between the cryptographic sortition mechanism and the FTS algorithm is that with cryptographic sortition, the selected node is not revealed until it submits the proof, and thus the node will not be targeted in advance by the adversaries. The initial seed for the VRF is generated at the beginning using distributed random number generator and subsequently used to create a new seed via VRF for the next round. The protocol also does not rely solely on the leader selection process for security. The committee is responsible for voting blocks which will be added to the chain in each round, meaning that the block is immediately finalized.

Algorand can operate for an asynchronous period, as long as they are followed by a synchronous period. Under this assumption, Algorand is proven to be safe as long as 51% of the total stake is controlled by honest participants. Because the

committee votes to finalize every block, i.e., there is no fork, many attacks associated with forks, e.g., double-spending, long-range, nothing-at-stakes, and bribe attacks, are mitigated. By using a node's private key and the seed as inputs, and distributing the private key in advance of the seed, grinding attack is mitigated as the adversary needs to influence the leader selection process at the same time.

Although there is more than one block created at each round in Algorand, the number of blocks created is still small, and the participants do not compete in hash rate to create blocks. Thus, the energy consumption of the Algorand protocol is low compared to that of the PoW mechanisms. Moreover, Algorand has a high transaction throughput, up to 875 Tx/s [24]. The protocol also has a significant advantage over many other PoS and PoW protocols since it provides immediate finality, i.e., the blocks and transactions are immediately finalized, and thus the transaction confirmation time is much faster, e.g., around 20 seconds [24], than those of the protocols adopting the longest chain rule such as Ouroboros and PoW protocols. However, similar to Casper, a significant issue is that the incentive mechanism is undefined in the paper. Algorand is currently adopted by several cryptocurrencies, including Algorand[¶] and Arcblock[‖].

2.3.6 Tendermint

The Tendermint protocol [26] employs the BFT voting protocol for block confirming. In Tendermint, the validators gain the right to vote by making a deposit. A proposer is selected from the validators based on their voting right to propose a block and include transactions in each round via a deterministic round-robin selection scheme. Similar to Algorand, the validators vote to confirm the proposed blocks in Tendermint, and thus blocks and transactions are immediately finalized.

[¶]https://www.algorand.com

[‖]https://www.arcblock.io

The block rewards are distributed among validators to incentivize consensus participations, and the deposits are confiscated for malicious behaviors.

Under the assumption of partial synchrony network, Tendermint is proven to be secure as long as 2/3 of the voting power is controlled by honest participants. Similar to Algorand, there is no fork in Tendermint, and thus fork related attacks are mitigated. However, the round-robin leader selection scheme is not clearly defined. The dynamic stake distribution is also ignored in the paper.

The energy consumption of the Tendermint protocol is low compared to PoW mechanisms because there is only one block created in each round. Similar to Algorand, Tendermint has high transaction throughputs, e.g., up to 800 Tx/s, and low transaction confirmation time, e.g., 1 second on average [90], due to the blocks being immediately finalized. Although proven to be secure against several types of attacks, the protocol generally lacks formal definitions and theoretical background, and the incentive mechanism is not analyzed. Currently, Tendermint has several applications in practice, such as BigchainDB[**], a blockchain database, and Ethermint[††], a cryptocurrency network.

[**]https://www.bigchaindb.com

[††]https://ethermint.zone

Table 2.2 : Summary of PoS-based Protocols

Protocol	Ouroboros [23]	Chains-of-Activity [22]	Casper [25]	Algorand [24]	Tendermint [26]
Type	PoS	PoS	PoS-PoW hybrid	PoS	PoS
Consensus Process	-Dynamic committee -Leader selection by 3-phased coin-tossing protocol -Utilize FTS algorithm	-Leader selection by stake and previous blocks -Utilize FTS algorithm	-Leader selection by PoW -Validators vote via BFT protocol to justify the checkpoint blocks.	-Dynamic committee -Leader selection based on stake -Utilize VRF	-Leader selection by round-robin selection -Validators vote to confirm blocks.
Transaction Adding	Input endorsers	Block creator	Block creator	Block creator	Block creator
Incentive Mechanism	Rewards are divided between the slot leaders and the input endorsers	-Leader collect reward -Leader's deposit will be confiscated for malicious behaviors.	Deposit is confiscated for malicious behaviors.	Undefined	-Rewards divided between validators. -Deposit is confiscated for malicious behaviors
Network Synchrony	Partial Synchrony	Undefined	Partial Synchrony	Asynchronous period in between synchronous periods	Partial Synchrony
Toleration	1/2	Undefined	1/3	1/2	1/3
Security issues	51% attack, bribe attack	Ignore adversary toleration, network synchrony, and dynamic stake distribution	Depends on underlying chain	Ignore incentive compatibility	-Ignore dynamic stake distribution -Leader selection is not clearly defined
Finality	Delayed	Delayed	Delayed	Immediate	Immediate
Transaction confirm	2 minutes	6 minutes	Depends on underlying chain	20 seconds	1 second
Transaction throughput	257 Tx/s	40 Tx/s	Depends on underlying chain	875 Tx/s	800 Tx/s
Applications	Cardano, Sp8de	Tezos	Ethereum (planned)	Algorand, Arcblock	BigchainDB, Ethermint

Chapter 3

BlockRoam: Blockchain-based Roaming Management System for Future Mobile Networks

In this chapter, we introduce BlockRoam, a novel blockchain-based roaming management system that provides an efficient data exchange platform among mobile service providers and mobile subscribers. Utilizing the PoS consensus mechanism and smart contracts, BlockRoam can significantly shorten the information exchanging delay, thereby addressing the roaming fraud problems. Through intensive analysis, we show that the security and performance of such PoS-based blockchain network can be further enhanced by incentivizing more users (e.g., subscribers) to participate in the network. Moreover, users in such networks often join stake pools (e.g., formed by mobile service providers) to increase their profits. Therefore, we develop an economic model based on Stackelberg game to jointly maximize the profits of the network users and the stake pool, thereby encouraging user participation. We also propose an effective method to guarantee the uniqueness of this game's equilibrium. The performance evaluations show that the proposed economic model helps the mobile service providers to earn additional profits, attracts more investment to the blockchain network, and enhances the network's security and performance.

The rest of this chapter is organized as follows. Section 3.1 describes the system model. Section 3.2 discusses BlockRoam's consensus mechanism in details. The economic model is proposed in Section 3.3. Evaluation results are then discussed in Section 3.4. Finally, conclusions and future works are given in Section 3.5.

3.1 Background and System Model

3.1.1 Current Roaming Systems

The current roaming system is illustrated in Fig. 3.1 [11]. In the current system, firstly, a roaming pact is established between two mobile service providers. Then, when a subscriber wants to use services from its HPMN while being in the service area of the VPMN, the subscriber sends a request to the VPMN. Then, the VPMN queries the HPMN about the services that the subscriber has subscribed to. This information is stored in the Home Location Register (HLR) database of the HPMN. If the subscription information is correct, the VPMN will provide the subscriber access to the corresponding services (e.g., voice or data service) through the Mobile Switching Center/Visited Location Register (MSC/VLR). The Call Detail Records (CDRs) are then sent to both networks where the CDRs are processed for subscription billings and invoices generation. Afterward, the VPMN sends a Transfer Account Procedure (TAP) file which contains the CDR information to the HPMN. Usually, there is a DCH company acting as a middleman, which validates and transmits the TAP files for the VPMN. Once the HPMN receives the TAP files, it will pay the VPMN in accordance with the roaming pact [11].

Fraud attacks in roaming occur when a subscriber gains access to the roaming services, but the HPMN is unable to charge the subscriber for the services provided. In this case, the HPMN still has to pay the VPNM for the facilities provided during the roaming process, which may result in significant financial loss. For example, a fraudulent SIM can use up to 18 hours of service on average, and in some incidents, the loss rate is up to €40,000 per hour [14]. The current roaming system is vulnerable to roaming fraud attacks mainly because of the delay in data exchanging between the HPMN and the VPMN. Even with the Near Real Time Roaming Data Exchange scheme [13], the data exchange can be delayed up to 4 hours, and thus it may take

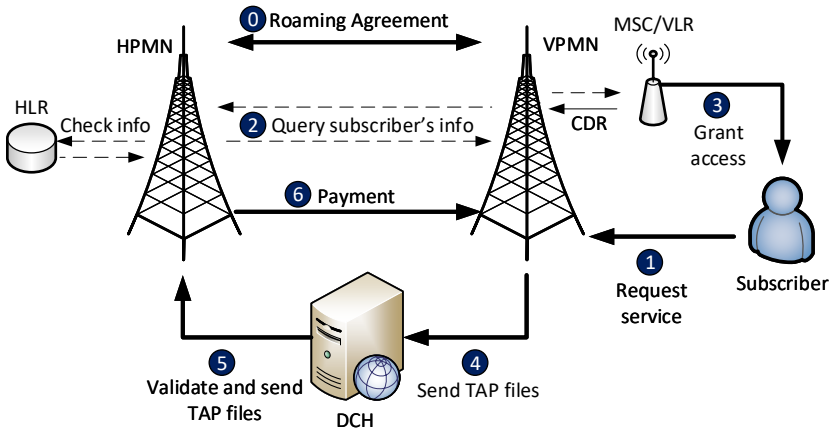


Figure 3.1 : Illustration of a typical roaming system [11].

a long time to detect and determine the fraud. Even if the fraud is found, it is still difficult for the HPMN to response as it does not have direct control over the VPMN's facilities [11].

3.1.2 Smart contracts and Consensus Mechanisms

A smart contract is a program stored in the blockchain network consisting of a set of rules created by users. If the rules are satisfied, the contract will automatically be enforced by the consensus mechanism. The content of a smart contract is visible to all network users, thus transparency is ensured [10]. For example, an HPMN and a VPMN can negotiate with each other and make a smart contract on the blockchain, which is triggered when a transaction with CDR data is sent to the smart contract address. Then, when the transaction is verified and added into the blockchain, all consensus participants execute the contract code and trigger the events according to the terms of agreement written in the contract, e.g., the HPMN automatically pays the VPMN as per their agreement.

The distributed consensus mechanism is the backbone of a blockchain network, which governs most of the blockchain's operations and ensures that once the data is

stored in a block, it is extremely difficult to be altered without the consensus of most of the nodes in the network. Currently, most of the blockchain networks have been employing the PoW consensus mechanisms. In the PoW, the users compete with each other in a solution searching procedure where a user with higher computational power may have higher opportunities to be the block winner who will add a new block to the chain and receive the reward. This competition leads to the waste of energy in PoW-based blockchain networks. Moreover, PoW-based blockchain networks often experience high delays in reaching consensus due to security reasons. This makes PoW consensus mechanisms inappropriate to implement in mobile roaming systems requiring low delay for fraud prevention.

Unlike the PoW, each block in PoS-based blockchain networks is dedicated to an authorized participant (leader) for mining in advance based on stakes of stakeholders in the network. This mechanism has many advantages over the PoW, including lower energy consumption and delay, and thus PoS-based blockchain applications can be employed effectively in networks with thousands of users [6]. Currently, there are several variations of the PoS mechanism, each has some desirable characteristics that are suitable for roaming management as well as some limitations that hinder their applicability in this specific context. In the following, we discuss advantages as well as disadvantages of each mechanism in details.

- *Proof-of-Activity (PoA)* [21] is one of the first PoS mechanisms proposed. This mechanism uses the block header of previous blocks to determine the leader for the current block, which helps to ensure unbiased randomness and prevent grinding attacks as proven in [21]. However, this mechanism is a hybrid PoW-PoS mechanism, and thus it has inherent limitations of PoW mechanism such as high energy consumption and long delay.

- *Casper* [25] is another PoW-PoS hybrid mechanism. Although this mecha-

nism is proven to be secure and able to mitigate many attacks, it still has performance limitations because of the PoW mechanism.

- *Chain-of-Activity (CoA)* [22] is a pure PoS mechanism, and thus it can achieve a relatively low delay (transaction confirmation time) (6 minutes) and requires negligible energy consumption. Nevertheless, the security of this mechanism is not proven rigorously in the paper, and its real-world application network has a relatively low transaction throughput (60 transactions per second).

- *Tendermint* [26], developed based on a BFT protocol, can achieve very low delay and high throughput. However, Tendermint relies on a set of validators to vote for the consensus, but how these validators are chosen is not discussed in the paper. Moreover, this mechanism requires high a communication complexity, i.e., $O(n^2)$, and the security analysis in the paper is not extensive (does not consider several attacks).

- *Ouroboros* [23] is a PoS mechanism with strong theoretical background and rigorous security analysis. The mechanism is proven to be secure, satisfying the persistence and liveness properties [38] with overwhelming probability, and able to mitigate many attacks. However, in case of a strong adversary, the delay is significantly increased.

- *Algorand* [24] is proven to be secure and can achieve high performance. However, the mechanism can tolerate only an adversarial ratio of 1/3, and there is no incentive mechanism and attack analysis in the paper

- *Delegated Proof-of-Stake (DPoS)* [27] is a variation of PoS that employs a committee to create blocks. However, this mechanism requires a lot more communications, is more prone to centralization, and can tolerate a smaller adversarial ratio compared to those of the other PoS mechanisms

The main advantages and limitations of the considered consensus mechanisms are summarized in Table 3.1. As observed in the table, all the consensus mechanisms have security flaws or performance limitations that make these mechanism unsuitable for the roaming management application. Thus, in the next Section, we will propose a consensus mechanism for BlockRoam to address these issues.

3.1.3 BlockRoam

3.1.3.1 Network Model

Our proposed blockchain-based system consists of two main components, namely the roaming management platform and the consensus mechanism as illustrated in Fig. 3.2. The roaming management platform supports complex interactions between the users, automates various roaming processes, and provides a universal currency, i.e., blockchain network tokens, for payments. In addition to the roaming processes, the network can also take part in the consensus mechanism to maintain the network's operations and security, store data (e.g., roaming pacts, subscriber information, and transaction history), and execute roaming processes such as payments and processing CDRs.

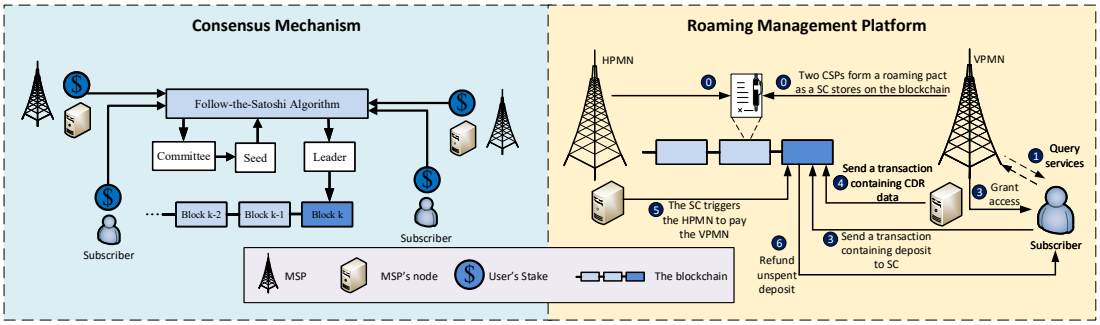


Figure 3.2 : Illustration of the proposed BlockRoam system.

Table 3.1 : Advantages and limitations of several PoS consensus mechanisms

Consensus Mechanism	**Advantages**	**Limitations**
Proof-of-Activity [21]	Low communication complexity, can mitigate several attacks	Need PoW, high energy consumption, long delay, security analysis is not extensive
Chain-of-Activity [22]	Low delay, low communication complexity, can mitigate several attacks	Low transaction throughput, security analysis is not extensive
Casper [25]	Secure, can mitigate several attacks	Need PoW, high energy consumption, long delay
Tendermint [26]	Low delay	Security analysis is not extensive, high communication complexity
Ouroboros [23]	Secure, can mitigate several attacks, defined incentive mechanism, low communication complexity	Long delay in case of adversarial attacks
Algorand [24]	Secure, low delay, high transaction throughput, low communication complexity	Can tolerate low adversarial ratio, no incentive mechanism, does not analyze attacks
DPoS [27]	Secure, low delay	High communication complexity, more centralized, can tolerate low adversarial ratio.

3.1.3.2 Roaming Management Procedure

The roaming process, the main procedure of the roaming management platform, consists of seven main steps as follows:

- *Step 0:* Two mobile service providers form a roaming pact consisting of tariff plans for services offered to the subscribers and the payment agreement between two mobile service providers. This roaming pact is made in the form of a smart contract and stored in the blockchain.
- *Step 1:* When a subscriber (roamer) wants to use services from its HPMN, the subscriber queries the VPMN and receives available tariff plans as per the roaming agreement between the VPMN and the HPMN.
- *Step 2:* If the subscriber agrees to use the service, the subscriber sends a transaction containing a sufficient amount of money (in form of digital tokens) to the smart contract's address.
- *Step 3:* When the transaction is verified and sent successfully, the VPMN will grant the subscriber access to roaming facilities.
- *Step 4:* When the subscriber finishes its roaming service, the VPMN sends a transaction to the smart contract's address, which consists of the CDR data of the provided service.
- *Step 5:* The smart contract then automatically calculates the subscriber's service fee and sends it to the HPMN. The smart contract also triggers a transaction from the HPMN to the VPMN for payment of the service.
- *Step 6:* Finally, the smart contract sends the unused tokens to the subscriber.

3.1.3.3 Benefits

BlockRoam has the following advantages over the traditional roaming system:

- *Roaming fraud prevention:* The main obstacle to prevent and react to fraud attacks is the significant delay in data exchange, i.e., up to 4 hours. Our proposed system employs the PoS mechanism to speed up the data exchanging process, e.g., approximately 3 minutes on average as later shown in Section 3.2, and thus fraud attacks can be detected much earlier. Moreover, by using smart contracts, the billing process is executed right after the service usage finished. As a result, roaming fraud can be significantly mitigated.

- *Cost saving:* In our proposed system, the CDRs are stored in the blockchain and processed by smart contracts. Therefore, the DCHs are no longer needed, and thus the middleman fees are eliminated. Moreover, our system automates various processes, such as subscribers billing and HPMN payments, which can further reduce operational costs. Furthermore, our system's energy consumption is negligible compared to that of PoW-based systems, and thus our energy cost is much lower.

- *Security and privacy:* Using cryptographically secure mechanisms, the privacy and security of the subscribers can be significantly improved. Each subscriber in the network uses a pair of public and private keys for identification and verification. The network only needs the subscriber's digital signature which can be easily verified and almost impossible to forge. This also protects the anonymity of the subscribers, as the subscriber's real-life identity is completely unrelated to the network identity.

3.2 BlockRoam's Consensus Mechanism

We have shown that the existing PoS mechanisms are not suitable for the roaming management application due to either security flaws, or insufficient performance ability. Therefore, in this section, we propose a novel consensus mechanism for BlockRoam. We also conduct analyses to show that the proposed consensus mechanism can satisfy the strict security requirements and achieve more desirable performance compared to existing mechanisms.

3.2.1 Proposed Consensus Mechanism

3.2.1.1 Epochs and Time Slots

In our proposed consensus mechanism, time is divided into epochs, each of which is composed of N_e time slots. At the first time slot of epoch e_k, a committee, consists of some users (stakeholders), executes an election protocol to elect one leader for each time slot in epoch e_k. The election protocol also selects the committee members for the epoch e_{k+1}. If a leader fails to broadcast its block during its designated time slot (being offline during its time slot), an empty block will be added to the chain. The leader is also instructed to not change its broadcast blocks at any later time.

3.2.1.2 Leader and Committee Election Protocol

To elect the leaders and committee members, the committee members of epoch e_k execute the Publicly Verifiable Secret Sharing (PVSS) protocol [91] to create seeds for the FTS algorithm [6]. The PVSS protocol allows the protocol participants to produce unbiased randomness in the form of strings and any network user to verify these strings. Moreover, the PVSS protocol can tolerate an adversarial ratio of up to 1/2, and this protocol is very efficient in terms of communication complexity, i.e., $O(m)$ where m is the number of committee members [91]. Once the random strings are created, they are used as the seeds for the FTS algorithm (a hash function that

takes any string as input and outputs some token indices [6]). The current owners of these tokens are then chosen as leaders of epoch e_k and committee members of epoch e_{k+1}.

3.2.1.3 Incentive Mechanism

The incentive mechanism plays a crucial role in ensuring that the stakeholders follow the consensus mechanism properly. To this end, the incentive mechanism needs to incentivize participation in the consensus mechanism via a reward scheme and penalize malicious behavior via a penalty scheme. In the reward scheme, a leader will receive a fixed number of tokens when the leader adds a new block to the chain. The probability P_n that user n is selected by the FTS algorithm in a network of N stakeholders is

$$P_i = \frac{s_n}{\sum_{n=1}^{N} s_n}, \tag{3.1}$$

where s_n is the number of stakes (tokens) of stakeholder n. As observed from 3.1, the more stakes a stakeholder has, the higher chance it can be selected to be the leader and able to obtain the reward. For the penalty scheme, the leader is required to make a deposit that will be locked during its designated epoch to prevent nothing-at-stake, bribe [6], and transaction denial attacks [23]. The stakes of committee members are also locked during the epoch that they are serving in the committee to prevent long-range attacks [6].

3.2.2 Security Analysis

3.2.2.1 Blockchain Properties

To maintain the blockchain's operations and security, a consensus mechanism must satisfy the following properties [38]:

- **Persistence:** Once a transaction is confirmed by an honest user, all other honest users will also confirm that transaction, and its position is the same for

all honest users.

- **Liveness:** After a sufficient period of time, a valid transaction will be confirmed by all honest users.

In our proposed system, persistence ensures that once a transaction is confirmed, it cannot be reverted. Without persistence, a fraudster can use the roaming services for free. For example, a fraudster can perform a double-spending attack by firstly sending a transaction Tx_1 to the smart contract. Then, after the VPMN has granted the fraudster access to the roaming service, the fraudster broadcasts a transaction Tx_2 which sends the tokens of Tx_1 to another address (e.g., the fraudster's second account). If Tx_1 has not been confirmed, Tx_2 is still valid and may be confirmed by honest users.

While the persistence property ensures data immutability, the liveness property ensures that every valid transaction will eventually be included in the chain. Without liveness, an attacker might successfully block every transaction coming from the Mobile Service Provider, and consequently, the roaming process cannot commence. It has been proven in [38] that the persistence and liveness properties are ensured if the consensus mechanism satisfies the following properties:

- **Common prefix (CP) with parameter $\kappa \in \mathbb{N}$:** For any pair of honest users, their versions of the chain $\mathcal{C}_1, \mathcal{C}_2$ must share a common prefix. Specifically, assuming that $\mathcal{C}_2$ is longer than $\mathcal{C}_1$, removing κ last blocks of $\mathcal{C}_1$ results in the prefix of $\mathcal{C}_2$.

- **Chain growth (CG) with parameter $\varsigma \in \mathbb{N}$ and $\tau \in (0, 1]$:** A chain possessed by an honest user at time $t + \varsigma$ will be at least $\varsigma\tau$ blocks longer than the chain it possesses at time t.

- **Chain quality (CQ) with parameter $l \in \mathbb{N}$ and $\mu \in (0, 1]$:** Consider any

part of the chain that has at least l blocks, the ratio of blocks created by the adversary is at most $1 - \mu$.

We prove that our proposed consensus mechanism can satisfy the common prefix, chain growth, and chain quality properties with overwhelming probabilities in the following Theorem.

Theorem 3.1. *BlockRoam's consensus mechanism satisfies the common prefix, chain growth, and chain quality properties with overwhelming probabilities.*

Proof. See Appendix A.1. □

3.2.2.2 Roaming Fraud Protection Ability

To evaluate the roaming fraud protection ability of our system, we focus on the average resolution time t_{total}, i.e., the average time between the occurrence of a roaming fraud attack and the execution of the responses to the attack. t_{total} is the sum of every stage's duration at the reactive layer, i.e., $t_{total} = t_C + t_D + t_S + t_R$. Since our proposed system can achieve a much lower t_C compared to the traditional roaming system, i.e., approximately 3 minutes (as later shown in Section 4.2.3) compared to 4 hours, the t_{total} of our system is nearly 4 hours shorter than that of the traditional roaming system.

3.2.2.3 Blockchain Attacks Mitigation

In the following Theorem, we prove that our proposed BlockRoam can also be able to mitigate and prevent a variety of emerging blockchain attacks such as double spending, grinding, bribe, nothing-at-stakes, and long-range attacks.

Theorem 3.2. *BlockRoam can mitigate double-spending, grinding, nothing-at-stakes, bribe, transaction denial, and long-range attacks as long as the adversary does not control more than 50% total network stakes.*

Proof. See Appendix A.2. □

When the adversary controls more than 50% of the total network stakes, both the persistence and liveness properties are no longer guaranteed [23]. Consequently, attacks such as double-spending, nothing-at-stakes, and transaction denial attacks can no longer be mitigated.

3.2.3 Performance Analysis

In Table 3.2, we examine and compare the transaction confirmation times under different adversarial ratio (percentage of stakes in PoS or computational power in PoW that the adversary controls) of a PoW blockchain network (Bitcoin), a PoS network with delayed finality (Cardano), and BlockRoam. The transaction confirmation time is the time it takes to reach a common prefix violation probability $\Pr_{\text{CP}} \leq 0.1\%$. Based on (14), κ can be determined, and then κ is multiplied with the slot time to calculate the transaction confirmation time. Our slot time is set to be 20 seconds (the same as that of Cardano [56]). The transaction confirmation times of Bitcoin and Cardano are presented in [23].

As observed in Table 3.2, the more stakes the adversary controls, the longer the transaction confirmation time is. Moreover, 51% attack [23] can break most of the PoW-based and PoS-based blockchain networks. Specifically, an adversary controlling more than 51% of total computational power in a PoW-based network or 51% of total stakes in a PoS-based network can successfully perform many attacks, including double-spending, nothing-at-stakes, and transaction denial attacks. Therefore, it is critical to attract more participants to our PoS-based blockchain system in order to increase the network's total stakes, thereby improving the common prefix violation probability and transaction confirmation time. In the next section, we will introduce an effective economic model that can jointly maximize profits for the participants, encouraging them to participate in the network and thus improving

Table 3.2 : Transaction confirmation times in minutes

Adversarial ratio	Bitcoin	Cardano	BlockRoam
0.10	50	5	1
0.15	80	8	1.3
0.20	110	12	1.6
0.25	150	18	1.6
0.30	240	31	2
0.35	410	60	2.3
0.40	890	148	2.6
0.45	3400	663	3

the network's performance and security.

3.3 Economic Model

3.3.1 Stake Pools and Stakeholders

In a PoS-based blockchain network, the probability that an individual user (stakeholder) with a small number of stakes is selected to be the leader is low as shown in (3.1). Moreover, when a stakeholder is selected to be the leader, it needs to be online during its designated time slot to (1) collect transactions from other users, (2) validate these transactions, (3) create a block containing valid transaction, (4) broadcast the block to the network. Therefore, if the stakeholder's connection is poor, it fails to create a valid block, and consequently it cannot obtain the block reward. Thus, stakeholders who participate in the consensus process need to maintain a strong connection to the network, which incurs an operational cost, e.g., \$40 to \$300 per month [92]. Therefore, small stakeholders often pool their stakes together to increase their opportunities to be the leaders and share operational costs,

which results in the formation of stake pools, e.g., [93–95]. Such formation of a stake pool is also beneficial for the blockchain because no transaction is processed when the leader fails to create a valid block (which reduces transaction throughput). In BlockRoam, the stakeholders, e.g., the subscribers, might be more inclined to join the stake pool (e.g., formed by mobile service providers) to reduce their operational costs and have more stable incomes. A stake pool often charges a part of the stakeholder's profits for joining the pool, e.g., the Stakecube pool charges 3% of each reward a stakeholder receives [94]. In this section, we introduce an economic model using Stackelberg game in order to jointly maximize the profits of the stake pool and stakeholders, which is beneficial for mobile service providers and BlockRoam's operation and security.

We consider a PoS-based blockchain network with one stake pool and N stakeholders. The stakeholders have stake budgets $\mathbf{B} = (B_1, \ldots, B_N)$ and individual operational costs $\mathbf{C} = (C_1, \ldots, C_N)$. The stake pool has its own stake σ, and the pool defines a cost c and a fee α in advance for users who are interested in participating in the pool. The pool's cost is charged for joining the pool and maintaining its operations. The pool's fee is the profit margin of the pool's owner, which usually ranges from 1% to 9% in real-world stake pools, e.g., [93–95]. The stakeholders can use their budgets to invest p_i stakes to the pool and m_i stakes for self-mining (individually participate in the consensus process), such that $p_i + m_i \leq B_i$. Let denote $\mathcal{N}_p$ to be the set of stakeholders who invest in the pool, the probability P^w that the pool is selected to be the leader and obtains a block reward R is proportional to the pool's stakes in the total network stakes, i.e.,

$$P^w = \frac{\sigma + \sum_{n \in \mathcal{N}_p} p_n}{\sigma + \sum_{n \in \mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j}. \tag{3.2}$$

After receiving the reward R, the pool calculates each stakeholder's reward r_i^p based on the proportion P_i^p of stakeholder i's stakes in the total stakes of the pool, which

is

$$P_i^p = \frac{p_i}{\sigma + \sum_{n \in \mathcal{N}_p} p_n}. \tag{3.3}$$

The pool then charges a fee for α percentage from each stakeholder's reward and a cost of $c\mathrm{e}^{-p_i}$ before the reward is finally sent to each stakeholder. Since the cost decreases exponentially as the stakes increase, it encourages the stakeholders to invest more stakes to the pool. Thus, when a stakeholder i invests p_i stakes to the pool, the stakeholder's expected reward r_i^p is given by

$$\begin{aligned} r_i^p &= P^w P_i^p (1-\alpha) R - c\mathrm{e}^{-p_i}, \\ &= \frac{p_i}{\sigma + \sum_{n \in \mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j} (1-\alpha) R - c\mathrm{e}^{-p_i}. \end{aligned} \tag{3.4}$$

In the case if the stakeholder i uses m_i stakes to self-mine, its expected reward is

$$r_i^m = \left(\frac{m_i}{\sigma + \sum_{n \in \mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j} \right) R - C_i, \tag{3.5}$$

where $\dfrac{m_i}{\sigma + \sum_{n \in \mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j}$ represents the proportion of stakeholder i's stakes in the total network stakes. Then, the profit of the pool can be calculated as follows:

$$\begin{aligned} U_p = &\frac{\sigma}{\sigma + \sum_{n \in \mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j} R \\ &+ \sum_{i \in \mathcal{N}_p} \left(\frac{p_i \alpha}{\sigma + \sum_{n \in \mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j} R + c\mathrm{e}^{-p_i} \right). \end{aligned} \tag{3.6}$$

The total profit of the pool consists of the profits from its own stakes, i.e., the first term in (3.6), and the costs and fees it charges the stakeholders, i.e., the second term in (3.6).

3.3.2 Stackelberg Game Formulation

In practice, a pool usually announces its cost and fee first, e.g., the fee to join the Stakecube pool can be found on its website [94]. Based on that information, the stakeholders will decide how much to invest. As a result, the interaction between

the stake pool and stakeholders can be formulated to be a single-leader-multiple-followers Stackelberg game [40]. In this game, the leader is the stake pool who first announces its strategy, i.e., costs and fees to join the pool, and then the stakeholders, i.e., followers, will make their decisions, e.g., to invest to the pool or not.

We denote s_p and s_i to be the strategies of the leader and follower i, respectively. Furthermore, we denote $\mathcal{S}_i$ to be the set of all possible strategies of follower i. Then, the best response s_i^* of a follower i can be defined to be the strategy set which gives the follower the best payoff given a fixed strategy $s_p = (\alpha, c)$ of the leader, i.e.,

$$U_i(s_i^*, s_p) \geq U_i(s_i', s_p), \forall s_i' \in \mathcal{S}_i. \tag{3.7}$$

Based on the follower's best response, the Stackelberg strategy for the leader is a strategy s_p^* such that

$$s_p^* = \underset{s_p}{\operatorname{argmax}} U_p(s_p, s_i^*). \tag{3.8}$$

Then, the Stackelberg solution can be defined as the tuple (s_p^*, s_i^*), and its corresponding utility tuple (U_p^*, U_i^*) is the Stackelberg equilibrium of the game. To find the Stackelberg equilibrium, the game can be divided into two stages. At the first stage, the leader announces its strategy. Then, at the second stage, the followers determine their strategies based on the leader's strategy. In the following, the backward-induction-based analysis is carried out to examine the Stackelberg equilibrium of this game.

3.3.2.1 Follower strategy

In this game, a follower's possible strategies can be divided into four cases:

- *Case 1:* Only invest stakes to the pool.
- *Case 2:* Only invest stakes for self-mining.
- *Case 3:* Simultaneously invest stakes to the pool and for self-mining.

- *Case 4:* Do not invest stakes to the PoS-based blockchain network.

We prove in the following Theorem that a follower's best response is use all its stakes either to invest to the pool or for self-mining.

Theorem 3.3. *A stakeholder's best response is to invest all stakes either to invest to the pool or for self-mining.*

Proof. See Appendix A.3. □

Since the a stakeholder's best response is to invest all its stakes, the best response can be deduced from either p_i^* or m_i^*. Therefore, from now on, we can denote the best response of follower i by the number of stakes it invest to the pool p_i^*. Then, the best response p_i^* of follower i can be expressed as a function of the pool's cost and fee as follows

$$p_i^*(\alpha, c) = \begin{cases} 0 & \text{if } C_i < \dfrac{B_i \alpha R}{\sigma + \sum_{j=1}^{N} B_j} + c\mathrm{e}^{-B_i}, \\ B_i & \text{if } C_i \geq \dfrac{B_i \alpha R}{\sigma + \sum_{j=1}^{N} B_j} + c\mathrm{e}^{-B_i}. \end{cases} \tag{3.9}$$

Theorem 3.4. *Given a strategy of the leader, there exists an optimal strategy for every follower and this strategy is unique.*

Proof. From (3.9), it can be seen that for every fixed strategy of the leader, a unique best response of every follower can be straightforwardly determined. □

3.3.2.2 Leader strategy

The backward induction mechanism [40] can be used to find the best strategy of the leader, which is the strategy that yields the highest payoff given the best responses of all followers, i.e., we have

$$\begin{aligned} s_p^* = \underset{s_p=(c,\alpha)}{\operatorname{argmax}} U_p(s_p, p_i^*) = \frac{\sigma}{\sigma + \sum_{j=1}^{N} B_j} R + \\ \sum_{i \in \mathcal{N}_p} \left(\frac{p_i^* \alpha}{\sigma + \sum_{j=1}^{N} B_j} R + c\mathrm{e}^{-B_i} \right). \end{aligned} \tag{3.10}$$

Since the total network stakes can be considered a constant, the profit from the pool owner's stake is also a constant (the first term in (A.10)) and does not need to be optimized. Moreover, since $p_i^*(\alpha, c)$ can only take two values, i.e., 0 or B_i, it can be represented by a binary decision variable $x_i \in \mathbf{x} = \{x_1, \ldots, x_N\}$, such that when $x_i = 1$, $p_i^* = B_i$ and when $x_i = 0$, $p_i^* = 0$. This helps to transform the optimization problem (A.10) into a Mixed-Integer Programming (MIP) optimization as follows:

$$\begin{aligned} \max_{\alpha,c,\mathbf{x}} \quad & \sum_{i=1}^{N} x_i \left(\frac{B_i R \alpha}{\sigma + \sum_{j=1}^{N} B_j} + c\mathrm{e}^{-B_i} \right), \\ \text{s.t.} \quad & \frac{B_i R \alpha}{\sigma + \sum_{j=1}^{N} B_j} + c\mathrm{e}^{-B_i} \leq L(1 - x_i) + C_i \quad \forall i \in \mathcal{N}, \\ & x_i \in \{0, 1\} \qquad\qquad\qquad\qquad\qquad\quad \forall i \in \mathcal{N}, \end{aligned} \tag{3.11}$$

where L is a sufficiently large number. The goal of (3.11) is to find the optimal values of $(\alpha, c, \mathbf{x})$ to maximize the pool's profit. The objective function represents the profit of the pool, where the stake pool can only charge the stakeholders who have invested in the pool. The first set of constraints ensures that only when the pool charges follower i less than C_i, x_i can take the value of 1, and thus the profit can be added to the total profit of the pool. The second set of constraints ensures that every x_i is a binary number. However, the objective function is nonlinear, i.e., it contains a multiplication of two decision variables x_i and α, which makes it much more complex to solve [96]. Thus, we transform (3.11) into an equivalent

Mixed-Integer Linear Programming (MILP) model as follows:

$$
\begin{aligned}
\max_{\alpha,c,\mathbf{x},\mathbf{y}} \quad & \sum_{i=1}^{N} y_i, \\
\text{s.t.} \quad & \frac{B_i R\alpha}{\sum_{j=1}^{N} B_j} + c\mathrm{e}^{-B_i} \leq L(1-x_i) + C_i && \forall i \in \mathcal{N}, \\
& y_i - Lx_i \leq 0 && \forall i \in \mathcal{N}, \\
& y_i - L(1-x_i) \leq \frac{B_i R\alpha}{\sum_{j=1}^{N} B_j} + c\mathrm{e}^{-B_i} && \forall i \in \mathcal{N}, \\
& x_i \in \{0,1\} && \forall i \in \mathcal{N}, \\
& y_i \in \mathbb{R}^+ && \forall i \in \mathcal{N}.
\end{aligned}
\tag{3.12}
$$

The transformation from (3.11) to (3.12) is done by a standard transformation technique which ensures the equivalence of the two models [97]. In particular, we introduce a new set of continuous variables $\mathbf{y} = \{y_1, \ldots, y_N\}$ which represents the profit which the pool can yield from follower i. Two new sets of auxiliary constraints, i.e., the second and third sets of constraints, are added to set the upper bound for y_i. If $x_i = 0$, i.e., follower i does not invest stakes to the pool, y_i will be upper-bounded by 0. If $x_i = 1$, y_i will be upper-bounded by $\frac{B_i R\alpha}{\sum_{j=1}^{N} B_j} + c\mathrm{e}^{-B_i}$. Thus, the optimal solution of (3.12) consists of two optimal values of α and c as shown in (A.10). Since the objective function is now linear, it can be solved efficiently by commercial solvers such as CPLEX [127].

3.3.2.3 Existence of the Stackelberg equilibrium

The existence of the Stackelberg equilibrium is proven via the existence of the optimal solutions of (3.12) in the following Theorem.

Theorem 3.5. *There exists at least one Stackelberg equilibrium in the considered stake pool game.*

Proof. See Appendix A.4. □

3.3.2.4 Uniqueness of the Stackelberg equilibrium

Although there always exists at least one Stackelberg equilibrium in this game, the uniqueness of the equilibrium cannot be guaranteed because both α and c are continuous variables. Consequently, there may be multiple pairs of α and c to achieve the same optimal utility as will be shown later in Section 3.4. In the conventional Stackelberg game model, the leader has only one primary priority, that is, to maximize the profit. Therefore, we propose a secondary priority for the leader, which is to minimize α. This serves two purposes, i.e., to attract followers with high stakes (as the amount the pool charges via the fee is proportional to the stakes) and to determine the unique optimal strategy for the game (i.e., the unique optimal strategy for both the leader and followers). Under the proposed approach, we can always obtain the unique Stackelberg equilibrium as proven in Theorem 3.6.

Theorem 3.6. *The considered stake pool game admits a unique Stackelberg equilibrium.*

Proof. See Appendix A.5. □

Based on this unique Stackelberg equilibrium, the stake pool can design appropriate parameters, i.e., cost and fee, to maximize its profits and attract more stakeholders to invest in the pool, and at the same time, the stakeholders can determine their best investment strategies to maximize their profits.

3.4 Performance Evaluation

3.4.1 Parameter Settings

We first study three small game instances, i.e., $\mathcal{G}_1$ to $\mathcal{G}_3$, to clearly show the relation between the leader and the followers in different situations. In these instances, we examine the utility functions of the stake pool and stakeholders. Particularly, we

present their corresponding utilities over a range of fees and costs, thereby demonstrating the effects of the stake pool strategy on the profit of the stakeholders and the stake pool. In $\mathcal{G}_1$, we consider a small game consisting one stakeholder and one stake pool with $C_1 = 0.1$, $b_1 = 5$, $R = 10$, and $\sigma = 10$. Then, we extend this game to $\mathcal{G}_2$ by considering five followers with the same configurations as that of the follower in $\mathcal{G}_1$, while other parameters are unchanged. After that, we consider game $\mathcal{G}_3$. Parameters are similar as those of $\mathcal{G}_2$ except that the followers have different budgets $\mathbf{B} = (5, 10, 13, 6, 8)$, operational costs $\mathbf{C} = (0.1, 0.3, 0.2, 0.6, 0.5)$, and $R = 50$.

To evaluate more general cases, we simulate 13 instances $\mathcal{G}_4$ to $\mathcal{G}_{16}$, each with 1,000 followers and different parameters as shown in Table 3.3. Among them, the first five games $\mathcal{G}_4$ to $\mathcal{G}_8$ are simulated with network parameters, such as R, $\mathbf{C}$, and $\mathbf{B}$, generated based on several real-world PoS-based blockchain networks [98–102]. Particularly, the values of R is determined using the number of coins these networks pay out (as block reward) per one block. For the values of $\mathbf{C}$, we first calculate the reference value C_r as follow:

$$C_r = \frac{100}{V_R N_b}, \tag{3.13}$$

where 100 is the average cost per month (in \$) to participate in the consensus process, V_R is the monetary value of each coin, and N_b is the number of blocks produced per month. As a result, C_r represents on average how many coins it costs to participate in the consensus process for one block. Then, the ranges of $\mathbf{C}$ can be determined based on C_r. For $\mathbf{B}$, we estimate the ranges by dividing the total number of coins in circulation and the total number of stakeholders in the network. Then, B_n and C_n of each stakeholder are generated randomly with normal distribution in the ranges listed in Table 3.3. The eight instances $\mathcal{G}_9$ to $\mathcal{G}_{16}$ are simulated to study the impacts of important parameters, i.e., R, $\mathbf{B}$, $\mathbf{C}$, and σ, on the game outcome. Taking $\mathcal{G}_4$ as a reference, we vary a single parameter at a time to evaluate the impacts of each parameter. For example, to study the impacts of R,

we decrease R ten times (compared to $\mathcal{G}_4$) in $\mathcal{G}_9$ and increase R ten times in $\mathcal{G}_{10}$, while all the other parameters are kept the same. The results, including the optimal leader strategy, optimal profit, and percentage of the network stakes invested in the pool, are obtained by solving the MILP optimization (3.12).

To evaluate the effects of the economic model on the network's security and performance, we simulate six game instances, $\mathcal{G}_{17}$ to $\mathcal{G}_{22}$. In instance $\mathcal{G}_{17}$, we simulate the network with a stake pool, similar to the previous game instances. In contrast, we simulate the network without a stake pool in instance $\mathcal{G}_{18}$. Since there is no stake pool, each stakeholder in this instance only has two choices, i.e., to participate in the consensus process if its operational cost is less than its profit ($C_i < \frac{B_i R}{\sum_{n=1}^{N} B_i}$), or does not participate in the consensus process if its operational cost is higher than its profits. Then, we examine the cases where there is an adversary who tries to attack the network with the same adversarial budget $B_{\mathcal{A}}$ in both instances. Under such adversarial attacks, we compare the security and performance of the network (with and without the stake pool) in terms of common prefix violation probability and transaction confirmation time. The common prefix violation probability is calculated using (14). Based on this, we find the minimum value of κ such that $\mathrm{Pr}_{\mathrm{CP}} < 0.1\%$ and multiply it with a block time of 20 seconds to determine the transaction confirmation time. For $\mathcal{G}_{17}$ and $\mathcal{G}_{18}$, we simulate a weak adversary with $B_{\mathcal{A}} = 20,000$ tokens. Similarly, we simulate a medium adversary with $B_{\mathcal{A}} = 40,000$ tokens for $\mathcal{G}_{20}$ and $\mathcal{G}_{21}$ and a strong adversary with $B_{\mathcal{A}} = 60,000$ tokens for $\mathcal{G}_{21}$ and $\mathcal{G}_{22}$. The other parameters of $\mathcal{G}_{17}$ to $\mathcal{G}_{22}$ are the same as those of $\mathcal{G}_4$.

3.4.2 Numerical Results

3.4.2.1 Leader and Follower's Utilities

The best response function of follower 1 in $\mathcal{G}_1$ is illustrated in Fig. 3.3a. Based on its best response, the profit of follower 1 can be determined. In this game, the profit

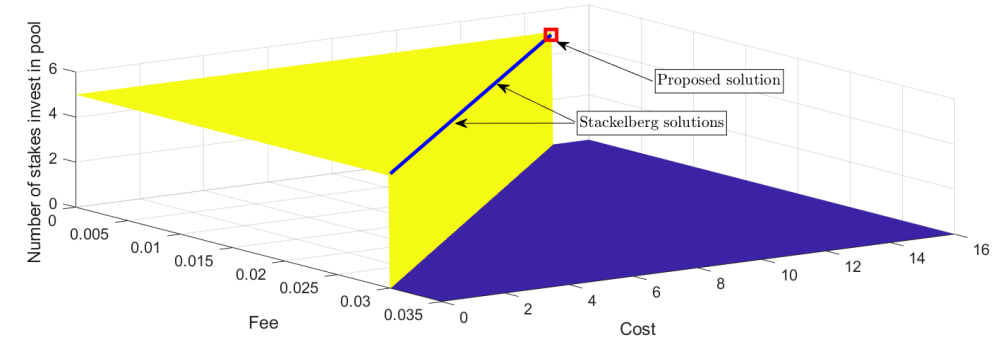
Table 3.3 : Parameters and results of 13 simulation instances.

$\mathcal{G}$	**Parameters**					**Stackelberg equilibrium**			
	R	**B** range	**C** range	σ	Based on	c^*	$\alpha^*(\%)$	U_p^*	% stake of the pool
4	1000	[1,250]	[0.05,0.1]	1000	Cardano [98]	3.2	4.0	28.95	69.5
5	200	[1,1000]	[0.0001,0.15]	1000	Algorand [99]	0.06	1.6	1.81	56.6
6	3.81	[1,400]	[0.0001,0.002]	1000	Cosmos [100]	0.1	14.4	0.35	61.2
7	78	[80,160]	[0.0001,0.02]	1000	Tezos [101]	40.1	6.1	2.29	48.9
8	500	[1,5000]	[0.001,0.3]	1000	NEM [102]	0.003	13.01	40.92	62.9
9	**100**	[1,250]	[0.05,0.1]	1000	Cardano	0.003	40.4	28.08	69.5
10	**10000**	[1,250]	[0.05,0.1]	1000	Cardano	0.207	0.4	29.13	69.5
11	1000	[1,250]	**[0.01,0.02]**	1000	Cardano	0.04	0.8	5.82	69.5
12	1000	[1,250]	**[0.25,0.5]**	1000	Cardano	0.04	20.5	140.54	69.5
13	1000	**[1,25]**	[0.05,0.1]	1000	Cardano	0.2	4.7	36.51	72.1
14	1000	**[1,2500]**	[0.05,0.1]	1000	Cardano	356.1	4.0	28.21	70.1
15	1000	[1,250]	[0.05,0.1]	**1**	Cardano	0.04	4.0	28.31	69.5
16	1000	[1,250]	[0.05,0.1]	**100000**	Cardano	0.02	10.9	28.15	69.5

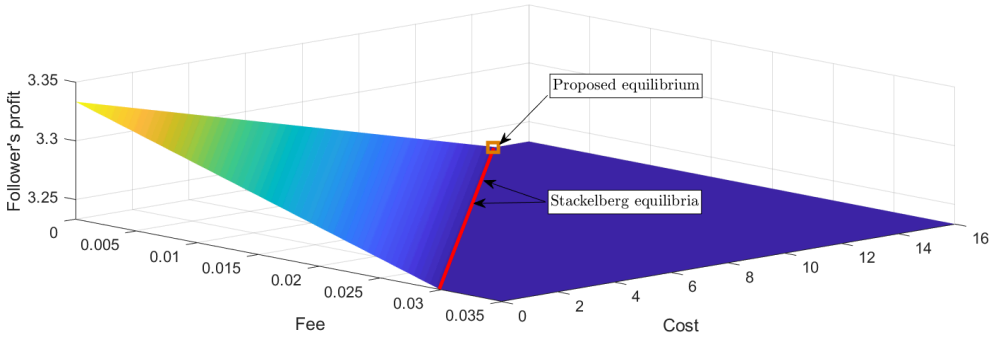
of the follower decreases as the pool's fee and cost increase as shown in Fig. 3.3b, but it is still higher than self-mining. The profit of the pool is illustrated in Fig. 3.3c. Since there is only one follower in $\mathcal{G}_1$, the profit of the pool only comes from follower 1, and thus it is upper-bounded by C_1. In this game, any pair of (c, α) that satisfies $\frac{\alpha R B_i}{\sigma + C_i} + c\mathrm{e}^{-B_i} = C_i = \frac{50}{15}\alpha + 0.007c = 0.1$ is a Stackelberg solution, which leads to multiple Stackelberg equilibria. Nevertheless, under our proposed approach, we can find the unique Stackelberg equilibrium for this game at $(c^*, \alpha^*) = (14.8, 0)$.

In $\mathcal{G}_2$, since the followers have the same budgets and operational costs, their best response and profit functions are the same, which are illustrated in Fig. 3.4a and Fig. 3.4b, respectively. These functions are similar to that of $\mathcal{G}_1$, except that the fee threshold is higher (7%). This is because there are more followers in $\mathcal{G}_2$, and thus (c, α) must satisfy $\frac{\alpha R B_i}{\sigma + C_i} + c\mathrm{e}^{-B_i} = C_i = \frac{50}{35}\alpha + 0.007c = 0.1$. The pool's profit in $\mathcal{G}_2$ is illustrated in Fig. 3.4c, which is upper-bounded by $5C_i$ in this game. The unique proposed equilibrium of this game has a corresponding solution $(c^*, \alpha^*) = (14.8, 0)$ as shown in Fig. 3.4c.

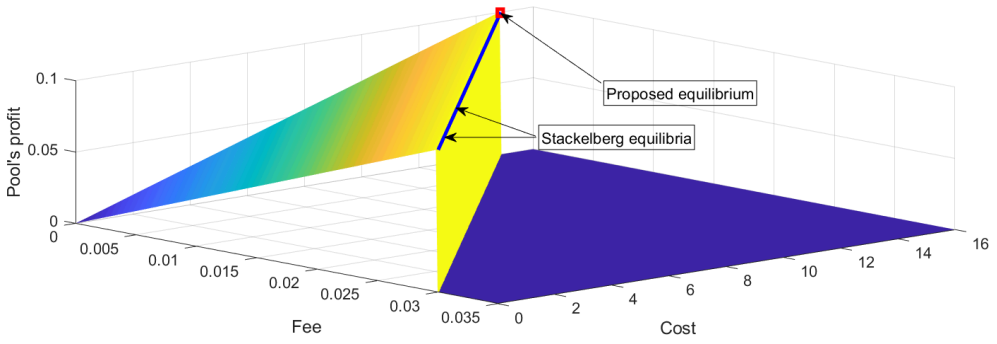
In $\mathcal{G}_3$, each follower's best response is illustrated in 3.5a. Typically, the higher



(a) Best response function of follower 1

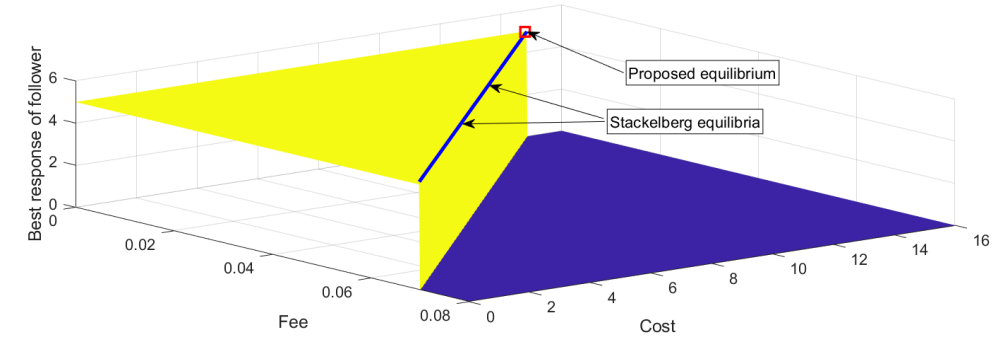


(b) Profit of follower 1

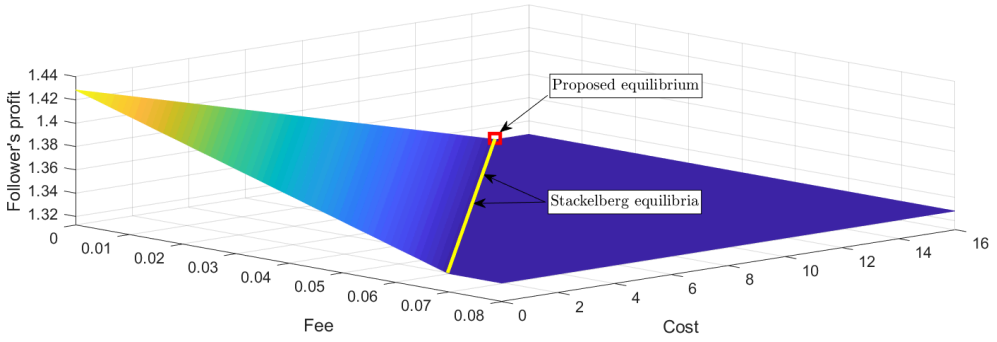


(c) Pool's profit

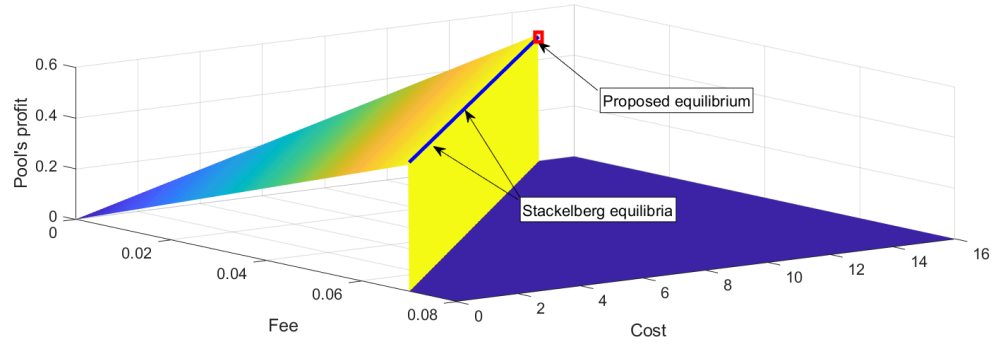
Figure 3.3 : Profit and best response of the leader and follower in $\mathcal{G}_1$.



(a) Best response function of follower 1



(b) Profit of follower 1

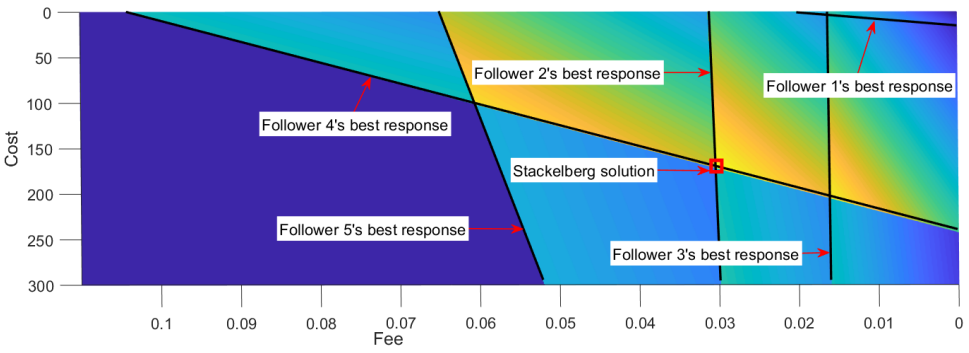


(c) Pool's profit

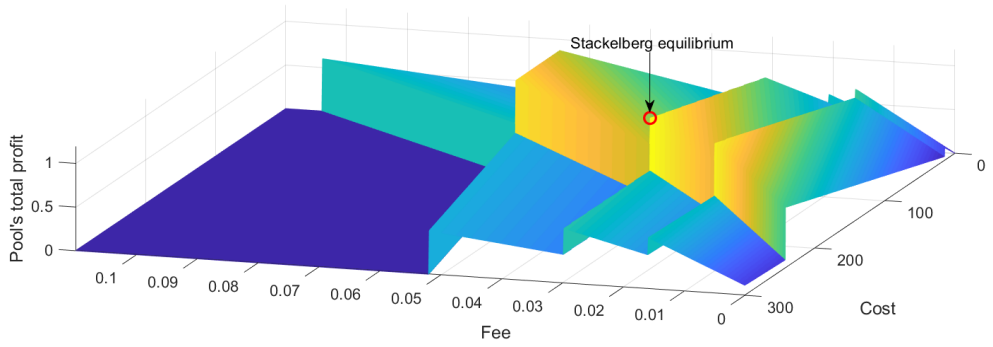
Figure 3.4 : Profit and best response of the leader and follower in $\mathcal{G}_2$.

a follower's budget is, the higher cost and the lower fee that follower is willing to accept, and vice versa. For example, follower 3 with the highest budget only accepts a fee of no more than 1.6%, and follower 1 with the lowest budget only accepts a cost lower than 15. This is because the budget is proportional to the fee the pool charges, while the cost decreases exponentially as the budget increases. The pool's profit in $\mathcal{G}_3$ is illustrated in Fig. 3.5b, with the leader's optimal strategy $(c^*, \alpha^*) = (171.3, 3.0\%)$ and optimal profit $U_p^* = 1.19$. Fig. 3.5c illustrates the profit the pool receives from each follower. Interestingly, at the obtained Stackelberg equilibrium of $\mathcal{G}_3$, the follower with the highest stake, i.e, follower 3, does not invest to the pool. The reason is that follower 3 has a relatively low operational cost, and thus the follower is more inclined to mine if the pool's cost and fee are too high. If the pool tries to incentivize all followers to invest by reducing α and c, its profit is only $U_p = 0.68$.

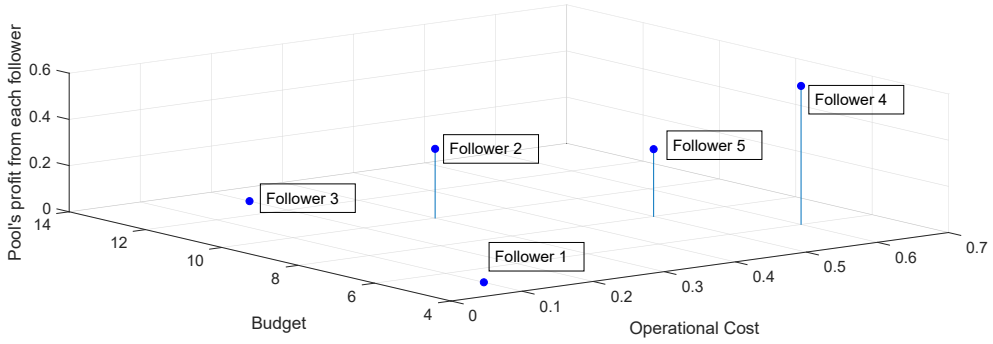
The results of more general cases are shown in Table 3.3. The five instances $\mathcal{G}_4$ to $\mathcal{G}_8$ are simulated with parameters adopted from several real-world blockchain networks [98–102]. The results show that the leader's optimal strategy and profit are significantly influenced by the network's parameters. For example, we obtain the optimal solution of $\mathcal{G}_4$ where $(c^*, \alpha^*) = (3.2, 4.0\%)$, $U_p^* = 28.95$, and approximately 69.5% of the total network's stakes (including σ) are invested to the pool. The profit that the pool earns from each follower depends on each follower's budget and operational cost, as shown in Fig. 3.6. Typically, a follower with higher cost and budget can give the pool more profit. However, similar to $\mathcal{G}_3$, if the budget is too high, the follower might not want to invest stakes to the pool, e.g., the followers with budget B_i greater than 150 do not join the pool in $\mathcal{G}_4$.



(a) Best responses of followers



(b) Pool's total profit



(c) Pool's profit from each follower

Figure 3.5 : Profit and best response of the leader and followers in $\mathcal{G}_3$.

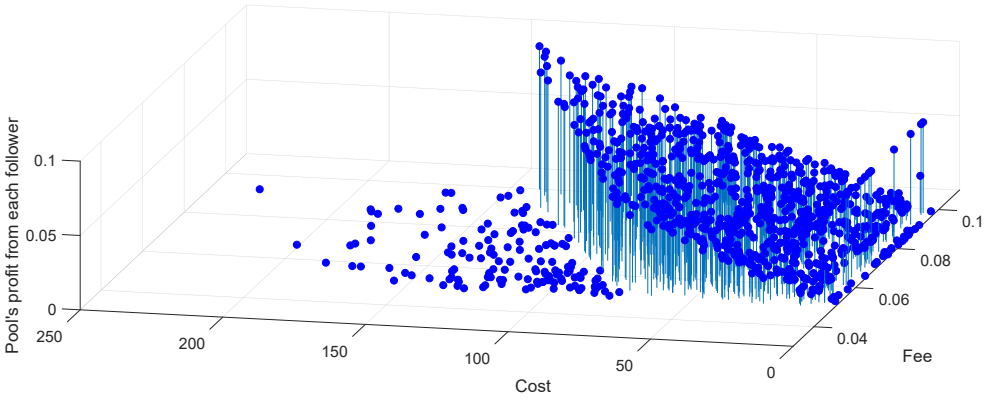


Figure 3.6 : Pool's profit from each follower in $\mathcal{G}_4$.

3.4.2.2 Impacts of Parameters

The eight games $\mathcal{G}_9$ to $\mathcal{G}_{16}$ are simulated to study the impacts of important parameters R, $\mathbf{B}$, $\mathbf{c}$, and σ, on the game's outcome. The impacts of those parameters are briefly described as follows:

- *Block reward R:* $\mathcal{G}_9$ and $\mathcal{G}_{10}$ are simulated to show the impact of R. As R increases, the pool's profit increases. However, the followers' operational costs are constant. Therefore, the pool has to decrease α when R increases, otherwise the followers will self-mine.

- *Operational costs* $\mathbf{C}$*:* $\mathcal{G}_{11}$ and $\mathcal{G}_{12}$ show how the followers' operational cost impacts the game's outcome. As the $\mathbf{C}$ increase, the pool can increase its profit by increasing α. The reason is that the followers' profits from self-mining are inversely proportional to the $\mathbf{C}$, and thus self-mining becomes less profitable if $\mathbf{C}$ are too high.

- *Budgets* $\mathbf{B}$*:* $\mathcal{G}_{13}$ and $\mathcal{G}_{14}$ show that as the budgets of followers increase, the pool can increase c but it has to reduce α. This is because the profit the pool receives via α is proportional to $\mathbf{B}$, while the profit the pool gets from c decreases

exponentially as **B** increase. Moreover, as **B** increase, the stakeholders invest fewer stakes to the pool and consequently the pool's profit decreases. The reason is that when **B** increase, the profit from self-mining also increases, and thus the followers prefer to self-mine.

- *The pool owner's stake σ:* The last two games show that as σ increases, although there are more stakes invested in the pool, its profit slightly decreases. The reason is that σ is inversely proportional to the pool's profit from each follower, and thus increasing σ means that the pool charges less from each follower. Consequently, the pool's profit decreases even though more followers invest to the pool.

3.4.2.3 Network Security and Performance

Fig. 3.7 illustrates the common prefix violation probability in instances $\mathcal{G}_{17}$ to $\mathcal{G}_{22}$. As observed from the figure, the instances with a stake pool achieve a lower common prefix violation probability compared to the instances without a stake pool. For example, for the medium adversary setting, the network achieves a 1.28 % violation probability, whereas the probability is 2.20 % if there is no stake pool. This is because if there is no stake pool, the stakeholders with small budgets may have negative utility if they participate in the consensus process (if their operational costs are higher than the reward they can obtain). Thus, the stakeholders holding few stakes may not participate in the consensus process, resulting in lower total network stakes. Consequently, the adversarial ratio can be increased, and the adversary may have higher chances to successfully attack the network.

Fig. 3.8 illustrates the transaction confirmation time in instances $\mathcal{G}_{17}$ to $\mathcal{G}_{22}$. Similar to the common prefix violation probability, the transaction confirmation time of the instances with a stake pool are lower than those of the instances without a stake pool. The reason is that, when the common prefix violation probability

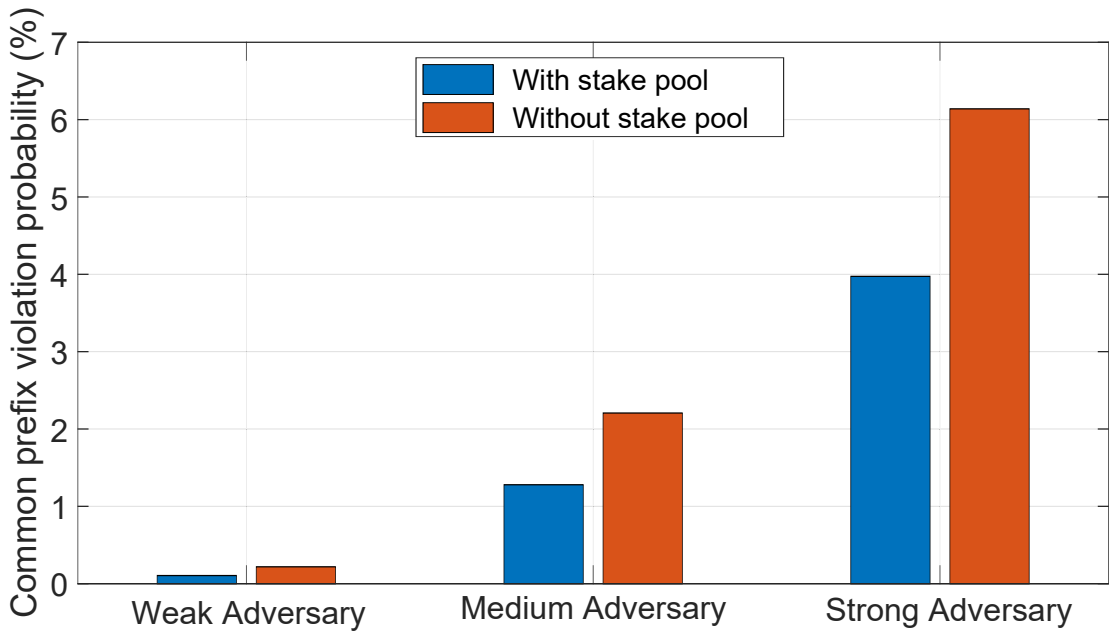


Figure 3.7 : Common prefix violation probability under different adversarial power.

is higher than 0.1%, the stakeholders have to wait for more blocks (higher κ) to confirm a transaction. Since the common prefix violation probabilities are higher in the instances without stake pool as discussed above, the transaction confirmation time is also higher in these cases. For example, in $\mathcal{G}_{22}$, the stakeholders have to wait for 15 blocks to confirm a transaction, whereas they have to wait for 13 blocks in $\mathcal{G}_{21}$, and thus the transaction confirmation time is lower in $\mathcal{G}_{21}$

3.4.3 Key Findings and Lessons

The key findings of the considered stake pool game are summarized as follows:

- We have proved that for a rational stakeholder, its best strategy is to invest all stakes from its budget to the blockchain network.
- We have proved that for each stakeholder, its best strategy is to invest all its stakes either to the pool or for self-mining.
- We have proposed an approach for the leader to decide its optimal strategy.

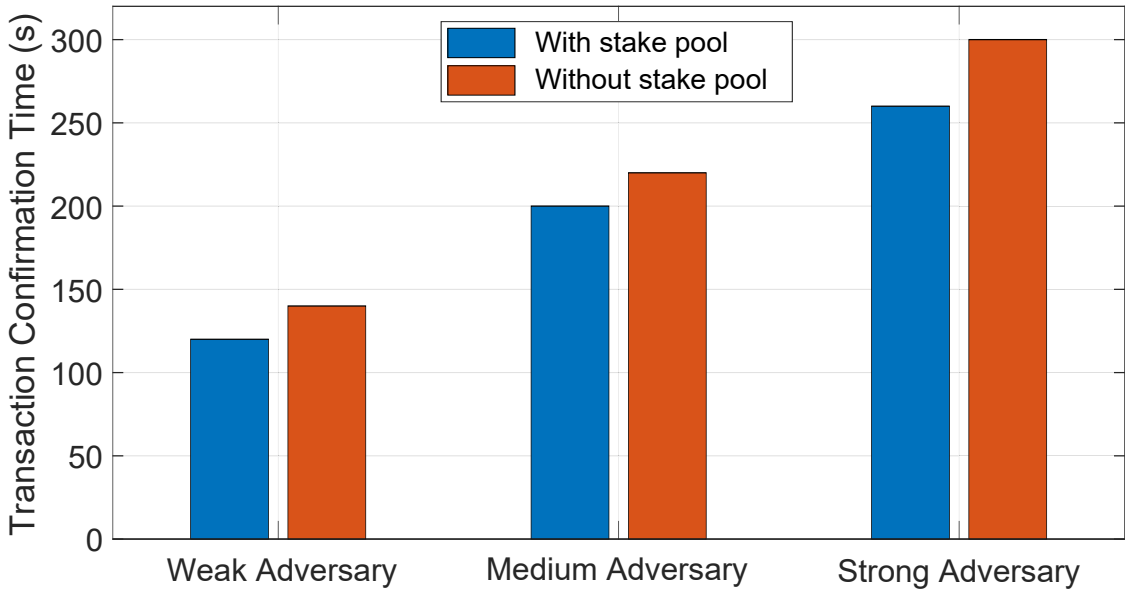


Figure 3.8 : Transaction confirmation time under different adversarial power.

Under this approach, there always exists the optimal and unique best strategies for the stakeholders and the stake pool owner. This approach also helps the stake pool to attract stakeholders with high stakes.

- We have shown that the proposed economic model can enhance the network's security and performance.

3.5 Conclusion

To address the problem of roaming fraud for mobile service providers, we have proposed BlockRoam, a novel blockchain-based roaming management system which consists of our thoroughly analyzed PoS consensus mechanism and a smart-contract-enabled roaming management platform. Moreover, we have analyzed and showed that BlockRoam's security and performance can be enhanced by incentivizing more users to participate in the network. Therefore, we have developed an economic model based on Stackelberg game to jointly maximize the profits of network users, thereby incentivizing their participation. We have analyzed and determined the best

strategies for the stakeholders and the stake pool. We have also proposed an effective solution that results in a unique equilibrium for our economic model. Lastly, we have evaluated the impacts of important parameters on the strategies and the equilibrium of the game. The proposed economic model can help the mobile service providers to earn additional profits, attract more investment to the blockchain network, and enhance the network's security and performance.

Chapter 4

FedChain: Secure Proof-of-Stake-based Framework for Federated-blockchain Systems

In this chapter, we propose FedChain, a novel framework for federated-blockchain systems, to enable effective transferring of tokens between different blockchain networks. Particularly, we first introduce a federated-blockchain system together with a cross-chain transfer protocol to facilitate the secure and decentralized transfer of tokens between chains. We then develop a novel PoS-based consensus mechanism for FedChain, which can satisfy strict security requirements, prevent various blockchain-specific attacks, and achieve a more desirable performance compared to those of other existing consensus mechanisms. Moreover, a Stackelberg game model is developed to examine and address the problem of centralization in the FedChain system. Furthermore, the game model can enhance the security and performance of FedChain. By analyzing interactions between the stakeholders and chain operators, we can prove the uniqueness of the Stackelberg equilibrium and find the exact formula for this equilibrium. These results are especially important for the stakeholders to determine their best investment strategies and for the chain operators to design the optimal policy to maximize their benefits and security protection for FedChain. Simulations results then clearly show that the FedChain framework can help stakeholders to maximize their profits and the chain operators to design appropriate parameters to enhance FedChain's security and performance.

The rest of this chapter is organized as follows. Section 4.1 describes the proposed FedChain framework, and the proposed consensus mechanism is presented in Section 4.2. Then, Section 4.3 introduces the problem formulation. After that, the

evaluation results are discussed in Section 4.4. Finally, conclusions are drawn in Section 4.5.

4.1 Federated-blockchain System

4.1.1 System Overview

Before elaborating on our proposed consensus mechanism and incentive mechanism, we provide a brief overview of the federated-blockchain system and the cross-chain transfer procedure in this section [8,103]. As illustrated in Fig. 4.1, the system is composed of two types of entities as follows:

- **Chains (blockchains):** In FedChain, individual blockchain networks, managed by blockchain operators, can communicate with each other via the cross-chain transfer protocol. Each chain has its own type of token and an individual consensus mechanism. When a new blockchain network wants to join the system, it only needs to negotiate with the existing chains and create smart contracts accordingly.

- **Users:** Users are the participants of the chains in the system. These users can freely exchange different types of tokens by using the smart contracts created by the operators. They can also participate in the consensus mechanism in every chain to earn economic profits through block rewards.

4.1.2 Cross-chain Transfer Procedure

The SPV mechanism [10] allows tokens from one chain to be securely transferred to another at a predetermined rate. When a user wants to prove that a transfer transaction from an originating chain to a destination chain is valid (not conflicting, digital signature matched the account), an SPV proof is submitted. This proof shows that the transfer transaction belongs to a valid block of the originating

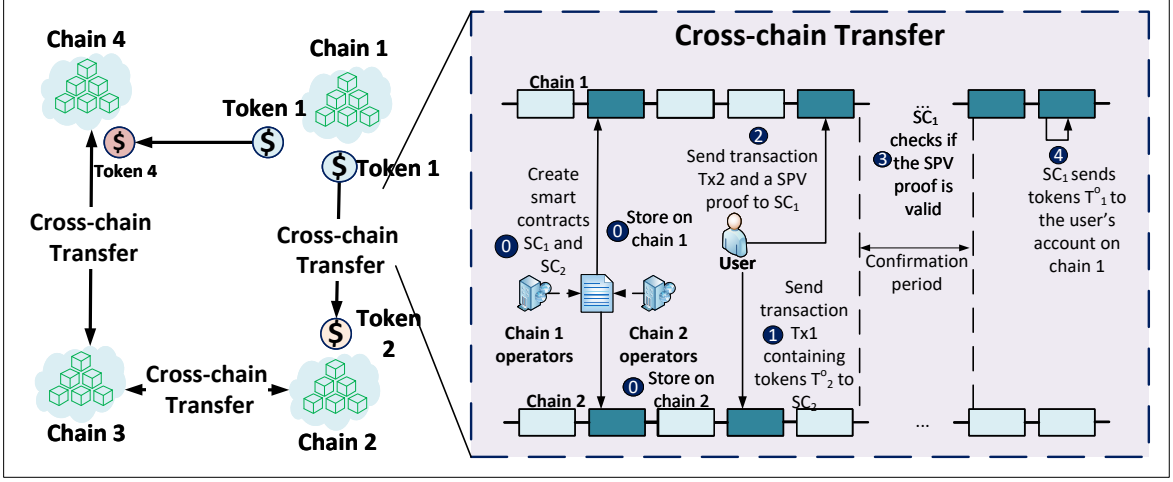


Figure 4.1 : The federated-blockchain system.

chain. Although this process takes a long time for confirmation, it eliminates the risk of centralization and single-point-of-failure compared to those of the centralized and federated scheme [103]. Therefore, the SPV proof is selected as the cross-chain transfer mechanism in our proposed FedChain. As illustrated in Fig. 4.1, the SPV-based token exchange procedure consists of several steps as follows:

- *Step 0:* Two chains negotiate an agreement which specifies the exchange rate between the two tokens. The chain operators then create in each chain a smart contract according to the agreement.
- *Step 1:* When a user wants to exchange T_2^o tokens into T_1^o tokens, the user sends a transaction Tx1, containing T_2^o tokens, from its account on chain 2 to the smart contract SC_2.
- *Step 2:* The user then sends a transaction Tx2 and an SPV proof from its account on chain 1 to SC_1. Tx2 then triggers SC_1 to validate the SPV proof.
- *Step 3:* During the confirmation period, SC_1 checks (1) the validation of the SPV proof and (2) any conflicts of the submitted SPV proof.

- *Step 4:* After the confirmation period, SC_1 sends a number of T_1^o tokens to the customer's address on chain 1 in accordance with the exchange rate.

The security features of the SPV proof mechanism are proven in [8]. The SPV proof points to the block that contains the cross-chain transfer transaction in the originating chain. Therefore, the validators only have to validate the block that contains the transaction. Thus, the security of the SPV proof only relies on the security of the originating chain, i.e., the SPV proof is secure if the originating chain is secure. However, this leads to a drawback of the SPV proof mechanism, which is the low confirmation speed (the validators have to wait until the transaction is confirmed on the originating chain). Moreover, as the stakes can be transferred between chains, if the security of one chain is violated, the whole system will fail. Therefore, in the next section, we will propose an effective consensus mechanism that can achieve lower transaction confirmation time compared to other conventional mechanisms while satisfying the persistence and liveness properties [38] and being able to prevent various blockchain attacks.

4.2 FedChain's Consensus Mechanism

In this section, we develop an effective consensus mechanism for FedChain with four new consensus rules based on the consensus mechanism proposed in [23]. Compared with other conventional consensus mechanisms such as [21–26, 39], our proposed consensus mechanism can satisfy both the liveness and persistence properties, prevent various blockchain attacks, and achieve an especially low transaction confirmation time as discussed in the following.

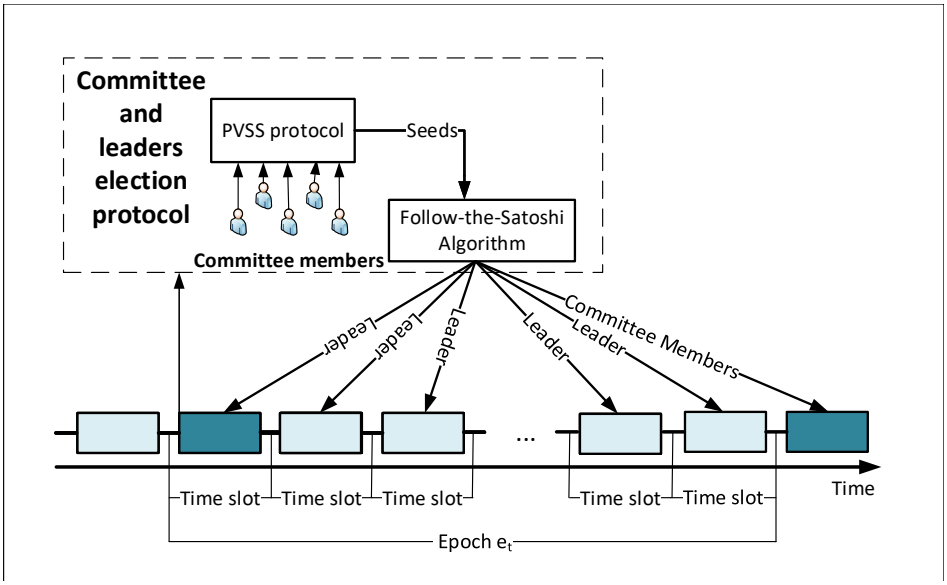


Figure 4.2 : Epoch-based committee and leader election.

4.2.1 Proposed Consensus Mechanism

4.2.1.1 Epochs and time slots

As illustrated in Fig. 4.2, time is divided into epochs, and each epoch is divided into time slots in FedChain's consensus mechanism. At the first time slot of epoch e_k, a committee consisting of some users (stakeholders) executes an election protocol to elect the leaders for the epoch e_k, such that for each time slot there is one designated leader who adds one new block to the chain. Similar to [23], we assume that the network is synchronous [104], and a time slot duration of 20 seconds is sufficient for the leader to broadcast a block to every node in the chain. The committee also select the committee members for the epoch e_{k+1}.

4.2.1.2 Leaders and committee election protocol

To elect the leaders and committee, the current epoch's committee members execute the Publicly Verifiable Secret Sharing (PVSS) protocol [91] to create seeds for the Follow-the-Satoshi (FTS) algorithm [6]. The PVSS protocol allows the participants to produce unbiased randomness in the form of strings and any network

user to verify these strings, as long as the majority (51%) of participants are honest (abiding by the rule of the consensus mechanism), as proven in [91]. Once the random strings are created, they are used as the seeds for the FTS algorithm. The FTS algorithm is a hash function that takes any string as input and outputs token indices [6]. The current owners of these tokens are then chosen as the leaders of this epoch or committee members of the next epoch. The probability P_n that user n is selected to be the leader and committee member by the FTS algorithm in a network of N stakeholders is

$$P_n = \frac{s_n}{\sum_{i=1}^{N} s_i}, \tag{4.1}$$

where s_n is the number of stakes of stakeholder n. As observed in (4.1), the more stakes a stakeholder has, the higher chance it can be selected to be the leader. Compared to [23], we design four new consensus rules as follow:

- I_1: After executing the PVSS protocol, the leader list is broadcast to every node in the chain.
- I_2: If a leader fails to broadcast its block during its designated time slot (e.g., being offline during its time slot), an empty block will be added to the chain
- I_3: Once a block is broadcast, the designated leader will not change the block at any later time.
- I_4: Upon receiving two forks (different versions of the chains), honest users adopt the longest valid fork, i.e., the longest fork that has no conflicting blocks and each block is signed by a designated leader.

Rule I_1 can be implemented by instructing the committee members to publish their votes (secret shares) that they used in the PVSS protocol execution, e.g., in the Data field of the block. As long as the adversary does not control more than 50% of the committee, the PVSS protocol can guarantee the unbiased randomness

of the result and allow everyone to verify [91]. Rules I_2 and I_3 can be implemented by instructing the leaders to not change their blocks, e.g., change the block's header or transactions. These two rules make sure that a leader cannot change its block once it is broadcast. As a result, every block created by an honest leader will become a checkpoint block. This helps to solidify the whole chain from the genesis block up to the latest honest block. Moreover, Rule I_2 also helps to maintain the chain growth even if the leader cannot broadcast the block in time, e.g., under DDoS attacks. Rule I_4 can be trivially implemented by instructing the stakeholder to check the leader list. Existing consensus mechanisms, e.g., [22, 23, 25, 26, 39], often adopt the longest chain rule to guarantee chain growth. Alternatively, in our proposed consensus mechanism, we have Rule I_2 to guarantee the chain growth property, and thus we can adopt a more secure version of the longest chain rule, i.e., I_4. These new consensus rules help to considerably reduce the probability that an adversary can successfully create an alternative version of the chain, thereby significantly improving the chain's security and performance. The detailed analysis will be discussed in Theorem 4.1.

4.2.1.3 Incentive mechanism

The incentive mechanism plays a crucial role in ensuring that the stakeholders follow the consensus mechanism properly. To this end, the incentive mechanism needs to incentivize consensus participants via a reward scheme and penalize malicious behavior via a penalty scheme. Note that, there are several research works on the design of blockchain's incentive mechanism, such as [105–108], but they are only applicable for individual chains with a specific application, e.g., blockchain-based mobile edge computing, consortium blockchains, and vehicular ad-hoc networks. Hence, they cannot be applied for federated-blockchain systems due to strong relations as well as competitions among blockchain service providers and stakeholders.

For the reward scheme, a leader will receive a fixed number of tokens when the leader adds a new block to the chain. This is also to incentivize the leaders to be online during their designated time slots. In single-blockchain settings such as Bitcoin [18] and Cardano [23], the block reward is set at a fixed value for a long period of time, e.g., 4 years in Bitcoin. However, in FedChain, having a fixed block reward scheme may pose security threats. The reason is that the stakes can be transferred between chains in our system, and the total network stakes can also vary in times, e.g., stakes increase from block rewards, and the stakes decrease from cross-chain transfers, etc. Since the probability that a stakeholder is elected to be the leader and able to obtain a block reward depends on the individual chain's stakes, stakeholders may transfer their stakes to a chain with a higher block reward to earn more profits. Consequently, this may attract stakes into a single chain and make it easier for adversaries to control the majority of stakes in the other chains. Therefore, in the following sections, we analyze the stakeholder rational strategy and propose a dynamic reward scheme to protect the decentralization of the whole system. With our proposed dynamic reward scheme, at the end of each epoch, the chains will adjust new block reward values for the next epoch, taking the total network stakes and the final stakes distribution among the chains in the current epoch into account. The dynamic reward scheme will be discussed in more details in Section 4.3.

For the penalty scheme, the leader is required to make a deposit that will be locked during its designated epoch to prevent nothing-at-stake, bribe [6], and transaction denial attacks [23]. The stakes of committee members are also locked during the epoch that they are serving in the committee to prevent long-range attacks [6]. How the proposed penalty scheme can prevent the mentioned attacks will be discussed in the following security analysis.

4.2.2 Security Analysis

4.2.2.1 Adversary and attack models

Since the SPV proof mechanism's security depends on the security of the individual chains, the security of the whole system also relies on the security of each chain. We consider two types of adversaries that target the individual chains, aiming to perform attacks such as double-spending, grinding, nothing-at-stakes, bribe, transaction denial, and long-range attacks [6]. As illustrated in Fig. 4.3, the considered types of adversaries are:

- **Static Adversary:** This type of adversary uses a stake budget B_A to attack a chain. Let B_n and γ denote the stake budgets of stakeholder n and the honest stake ratio, respectively. Then, the adversarial ratio, i.e., the ratio of adversarial stakes to the total network stakes, is $1-\gamma = \dfrac{B_A}{\sum_{n=1}^{N} B_n + B_A}$.
- **Adaptive Adversary:** In contrast to the static adversary setting, the adaptive adversary does not have a fixed number of stakes. However, this type of adversary can choose to corrupt N_A honest stakeholders and use their stakes to attack. Let $\mathcal{N}_A$ denote the set of corrupted stakeholders, the budget of the adaptive adversary can be defined by $B_A = \sum_{i \in \mathcal{N}_A} B_i$.

The models for the blockchain-specific attacks considered in this chapter are as follows:

- **Double-spending attack:** For such kind of attack, the attacker aims to revert a transaction that has been confirmed by the network (to gain back the tokens it has already spent). First, the attacker creates a transaction Tx1 in block $\mathcal{B}_i$ and waits until the block is confirmed. Then, the attacker can either create a conflicting transaction Tx2 or erase the block $\mathcal{B}_i$ from the chain, so that the proof of its spending is gone.

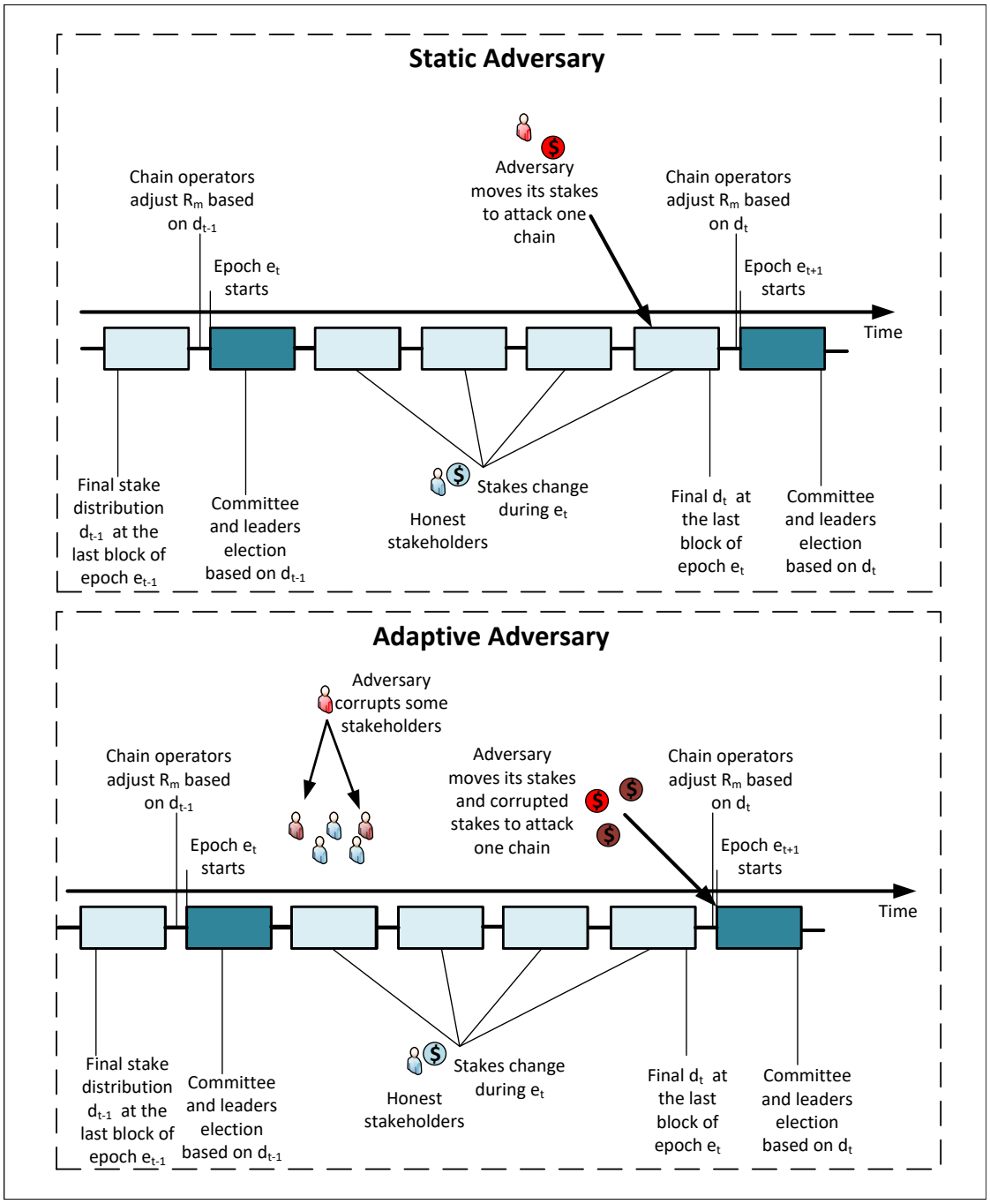


Figure 4.3 : Illustrations of the considered adversaries.

- **Grinding attack:** In grinding attacks, the attacker attempts to influence the leader election protocol to unfairly increase its chance to be selected as a leader. Generally, in protocols where the seeds of the FTS algorithm are derived from the block header, the attacker can check many possible different block contents (because block headers are created by hashing the block contents) to determine which one can give the attacker the best chance to be elected as a leader.

- **Nothing-at-stake attacks:** This type of attack specifically targets the PoS blockchains because, in contrast to PoW, blocks in PoS can be created with very little computation. In this attack, the attacker tries to create many forks or conflicting transactions. For example, the attacker can create two transactions to spend the same tokens at two vendors, i.e., Tx1 in fork $\mathcal{C}_1$ and Tx1 in fork $\mathcal{C}_2$. At this point, although both the transactions are not *confirmed*, they are both *valid* (not conflicted within their own fork).

- **Bribe attacks:** For such attacks, the attacker tries to bribe the leaders to create specific blocks, e.g., to support other types of attacks such as double-spending or transaction denial.

- **Transaction denial attack:** In this attack, the attacker tries to prevent transactions of every or some specific users from being included in the chain. To achieve this objective, the attacker has to either block the users' connection to the blockchain or not include the transactions when the attacker is the leader.

- **Long-range attack:** In a long-range attack, a leader immediately transfers its stakes to another account at the beginning of its designated epoch, and thus it can behave maliciously, e.g., performing attacks, for the rest of the epoch without consequences.

4.2.2.2 Blockchain properties

To maintain the blockchain's security, a consensus mechanism must satisfy the following properties [38]:

- **Persistence:** Once a transaction is more than κ blocks deep in the chain of an honest user, all other honest users will have that transaction in the same position in their chains.
- **Liveness:** After a sufficient period, a valid transaction will be confirmed by all the honest users.

In FedChain, persistence ensures that once a transaction is confirmed, i.e., more than κ blocks deep in the chain, it cannot be reverted. Without the persistence property, the adversary can successfully perform a double-spending attack by firstly sending a transaction to spend some tokens. After that transaction is confirmed, the adversary can create a fork to erase the transaction from the blockchain. If that fork is accepted by the honest users, the adversary can gain back the tokens it already spent. While the persistence property ensures data immutability, the liveness property ensures that every valid transaction will eventually be included in the chain. Without liveness, an attacker can block every transaction in a blockchain. The persistence and liveness properties are ensured if the consensus mechanism satisfies the following properties [38]:

- **Common prefix (CP) with parameter $\kappa \in \mathbb{N}$:** For any pair of honest users, their versions of the chain $\mathcal{C}_1, \mathcal{C}_2$ must share a common prefix. Specifically, assuming that $\mathcal{C}_2$ is longer than $\mathcal{C}_1$, removing κ last blocks of $\mathcal{C}_1$ results in the prefix of $\mathcal{C}_2$.
- **Chain growth (CG) with parameter $\varsigma \in \mathbb{N}$ and $\tau \in (0, 1]$:** A chain

possessed by an honest user at time $t+\varsigma$ will be at least $\varsigma\tau$ blocks longer than the chain it possesses at time t.

- **Chain quality (CQ) with parameter $l \in \mathbb{N}$ and $\mu \in (0,1]$:** Consider any part of the chain that has at least l blocks, the ratio of blocks created by the adversary is at most $1-\mu$. In the ideal case, $1-\mu$ equals the adversarial ratio $1-\gamma$.

Let $\Pr_{\text{CP}}$, $\Pr_{\text{CG}}$, and $\Pr_{\text{CQ}}$ denote the probabilities that the CP, CG, and CQ properties are violated. We prove that FedChain's consensus mechanism can satisfy the CP, CG, and CQ properties with overwhelming probability, i.e., $\Pr_{\text{CP}}$, $\Pr_{\text{CG}}$, and $\Pr_{\text{CQ}}$ are overwhelmingly low ($< 0.1\%$), in Theorem 4.1.

Theorem 4.1. *FedChain's consensus mechanism can satisfy the CP, CG, and CQ properties with* $\Pr_{\text{CP}} = (1-\gamma)^{\kappa}$, $\Pr_{\text{CG}} = 1$, *and* $\Pr_{\text{CQ}} < 1 - \exp\left(\frac{\mathrm{l}(\gamma-1)\delta^2}{2}\right)$.

Proof. We first prove $\Pr_{\text{CP}}$ by showing that the adversary needs to be the leader for κ consecutive blocks to violate CP. We then prove $\Pr_{\text{CG}} = 1$ by using Rule I_2. Finally, we prove $\Pr_{\text{CQ}} < 1 - \exp\left(\frac{\mathrm{l}(\gamma-1)\delta^2}{2}\right)$ by using the random walk and Chernoff bound. The detailed proof is provided in Appendix B.1. □

Fig. 4.4 illustrates the CP and CQ violation probabilities under different parameter values. As the adversarial ratio increases (i.e., the adversary controls more stakes in the chain), the attacker has more chances to successfully attack. However, the higher κ is, the lower the CP violation probability is. This means that the longer since a transaction is added to the chain, the more stable the transaction becomes. For example, if a transaction is at least seven blocks deep in the chain, the adversary has less than 1% chance to revert it, even if the adversary controls nearly 50% of the total network stakes. In contrast, if the transaction is only four blocks deep, the adversary with 49% stakes has more than 5% chance to revert the transaction. This

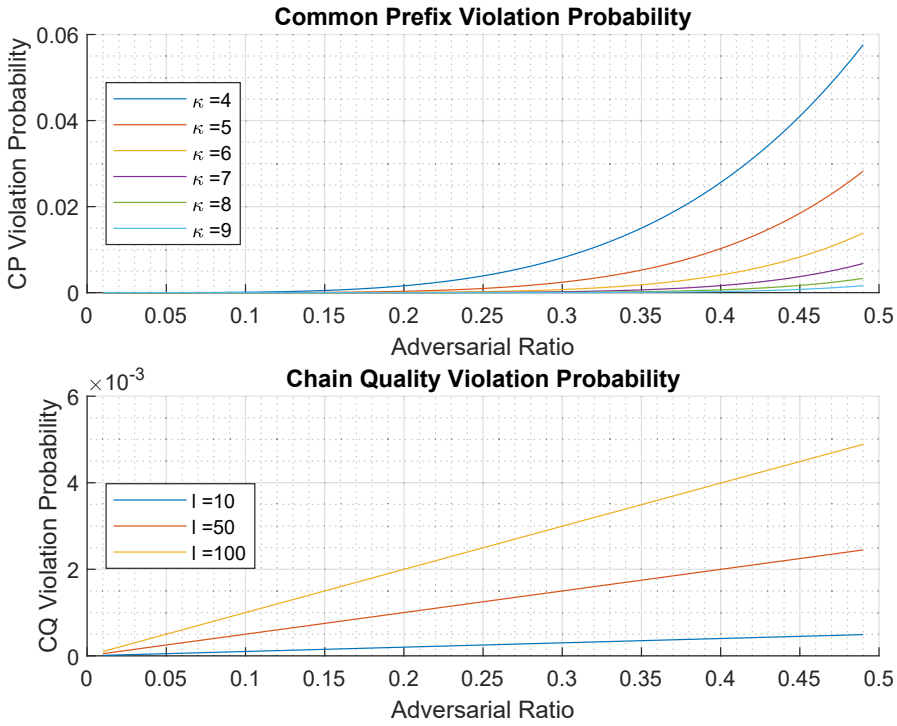


Figure 4.4 : Blockchain properties violation probabilities.

implies that the more stakes the adversary controls, the longer it takes to confirm a transaction, which is directly related to the performance and security of the chain.

For the $\Pr_{\mathrm{CQ}}$, the more blocks we consider, the higher chance the adversary can create more than $(1-\gamma)l$ blocks. For example, an adversary controlling 30% of network stakes has less than 0.1% chance to create more than three in ten blocks, but it has around 0.3% chance to create more than 30 in 100 blocks. This could be harmful to the network if the adversary wants to reduce the network's throughput (i.e., blocks/time slot). For example, an adversary with 30% network stakes has 0.3% chance to reduce the network throughput by 30% during 100 time slots by creating only empty blocks every time it is elected to be the leader.

4.2.2.3 Blockchain attacks prevention

In the following Theorem, we prove that our FedChain's consensus mechanism is able to prevent a variety of emerging blockchain attacks such as double spending, grinding, bribe, nothing-at-stakes, and long-range attacks.

Theorem 4.2. *FedChain's consensus mechanism can prevent double-spending, nothing-at-stakes, bribe, transaction denial attacks, grinding, and long-range attacks according to the considered adversary models.*

Proof. We prove that double-spending and nothing-at-stakes attacks are prevented if CP is not violated. Then, we prove that grinding attacks can be prevented by the PVSS protocol, and bribe attacks are prevented because the adversary does not know the leader in advance. Moreover, transaction denial attacks could be prevented if CG and CQ hold. Furthermore, long-range attacks are prevented because the leader's stakes are locked during the epoch. The detailed proof is provided in Appendix B.2. □

4.2.3 Performance Analysis

From the security perspective, we prove that the higher the adversarial ratio is, the higher the probabilities that the adversary can successfully perform attacks on the chain. Similarly, the adversarial ratio also has a negative impact on the performance of the network. In this performance analysis, we aim to analyze and compare the performance of our proposed consensus mechanism when it is employed by individual blockchains in the federated-blockchain system. As shown in Table 4.1, we examine and compare the transaction confirmation time under different adversarial ratio (percentage of stakes in PoS or computational power in PoW that the adversary controls) of a PoW blockchain network (Bitcoin), a PoS network with delayed

Table 4.1 : Transaction confirmation time in minutes

Adversarial Ratio	**Bitcoin**	**Cardano**	**FedChain's Consensus Mechanism**
0.10	50	5	1
0.15	80	8	1.3
0.20	110	12	1.6
0.25	150	18	1.6
0.30	240	31	2
0.35	410	60	2.3
0.40	890	148	2.6
0.45	3400	663	3

finality (Cardano), and FedChain's consensus mechanism. The transaction confirmation time of Cardano and Bitcoin, obtained from [23], is under optimal network conditions. This means that the time is theoretically calculated, only taking into account the effects of the adversarial ratio [23]. Specifically, the transaction confirmation time is the time it takes to reach a CP violation probability $\Pr_{\text{CP}} \leq 0.1\%$. For Fedchain's consensus mechanism, κ can be determined based on (B.1), and then κ is multiplied with the time slot duration to calculate the transaction confirmation time. Our time slot duration is set to be 20 seconds (the same as that of Cardano [56]).

As observed in Table 4.1, the more stakes the adversary controls, the longer the transaction confirmation time is. Moreover, the PVSS protocol no longer ensures unbiased randomness if the adversary controls more than 50% stakes in a chain. Therefore, it is critical to attract more participants to individual chains in order to increase the network's total stakes and prevent the adversary from controlling

more than 50% of network stakes. In the next section, we will introduce an effective incentive mechanism developed based on a Stackelberg game model that can jointly maximize profits for the participants and significantly enhance the network's performance and security for chain operators.

4.3 Stackelberg Game Formulation

In practice, chains usually announce their block rewards first, and then the stakeholders will decide how much to invest accordingly. Therefore, the interaction between the chains and stakeholders in FedChain can be formulated as a multiple-leaders-multiple-followers Stackelberg game model [40]. In this game, the leaders are the chains (managed by the chain operators) who first announce their block rewards, and then the stakeholders, i.e., followers, will make their decisions, e.g., how much to invest in each chain. It is worth noting that there are some approaches that apply the Stackelberg game models to blockchain systems in the literature, such as [109–111]. Nevertheless, these models can be applied for individual blockchains only, and thus they cannot be directly adopted for federated blockchain systems in which competitions between multiple blockchain service providers and stakeholders are taken into considerations.

4.3.1 Stakeholders and Chain Operators

FedChain consists of a set $\mathcal{M}$ of M chains and a set $\mathcal{N}$ of N followers. The leaders offers block rewards $\mathbf{R} = (R_1, \ldots, R_M)$. Stakeholders possess stakes with budgets, denoted as $\mathbf{B} = (B_1, \ldots, B_N)$. The stakeholders can use their stakes to take part in the consensus process of every chain to earn additional profits. Particularly, when stakeholder n invests s_n^m to chain m, its expected payoff U_n^m is:

$$U_n^m = \frac{s_n^m}{s_n^m + \sum_{i \in \mathcal{N}_{-n}} s_i^m} R_m, \tag{4.2}$$

where $\mathcal{N}_{-n}$ is the set of all stakeholders except stakeholder n. In the considered system, the stakeholders can freely invest within their budgets to any chain, i.e., $\sum_{m=1}^{M} s_n^m \leq B_n$. Thus, the total payoff of stakeholder n is

$$U_n = \sum_{m=1}^{M} U_n^m = \sum_{m=1}^{M} \left(\frac{s_n^m}{s_n^m + T_m} R_m \right), \tag{4.3}$$

where $T_m = \sum_{i \in \mathcal{N}_{-n}} s_i^m$ expresses the total stakes invested in chain m by all the other stakeholders.

4.3.2 Game Theoretical Analysis

4.3.2.1 Followers' strategy

To analyze the game, we first examine the existence of the follower sub-game equilibrium in Theorem 4.3.

Theorem 4.3. *There exists at least one Nash equilibrium in the follower sub-game.*

Proof. We prove existence of the equilibrium by proving that the strategy space is convex and U_n is concave $\forall n \in \mathcal{N}$ [39]. The detailed proof can be found in Appendix B.3. □

Then, we examine the uniqueness of the equilibrium in Theorem 4.4.

Theorem 4.4. *The follower sub-game equilibrium is unique, and the convergence to the equilibrium is guaranteed.*

Proof. We prove the uniqueness by showing U_n satisfies Rosen Theorem's conditions [112]. The detailed proof is provided in Appendix B.4. □

In this game, the stakeholders can invest any number of stakes within their budgets. However, as shown in Theorem 4.5, a rational stakeholder will always invest all its budget regardless of the other stakeholders' strategies.

Theorem 4.5. *For every follower n, the strategies that invest less than its total budget, i.e., $\sum_{m=1}^{M} s_n^m < B_n$, always give lower payoffs than the strategy that invests all the budget, i.e., $\sum_{m=1}^{M} s_n^m = B_n$, regardless of other followers' strategies.*

Proof. We compare the utility functions in two cases to prove that investing all stakes always brings more profits. The detailed proof is provided in Appendix B.5. □

As a result of Theorem 4.5, the strategies which invest less than the total budget can be removed from the strategy space of every follower. Then, we can reformulate the utility function to reflect the budget constraint as follows:

$$U_n = \sum_{m=1}^{M-1} \left(\frac{s_n^m}{s_n^m + T_m} R_m \right) + \frac{B_n - \sum_{m=1}^{M-1} s_n^m}{B_n - \sum_{m=1}^{M-1} s_n^m + T_M} R_M. \tag{4.4}$$

With the existence and uniqueness guaranteed, the only question remained is how to find the equilibrium point. Interestingly, for the considered game model, we can prove the exact formula of the equilibrium in Theorem 4.6.

Theorem 4.6. *The point where every follower's strategy satisfies $s_n^{*m} = B_n \frac{R_m}{\sum_{i=1}^{M} R_i}, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}$ is the unique equilibrium of the follower sub-game.*

Proof. We prove that at $s_n^{*m} = B_n \frac{R_m}{\sum_{i=1}^{M} R_i}, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}$, all the followers can maximize their profits,, and thus this is the equilibrium. The detailed proof is provided in Appendix B.6. □

Then, we can conclude that there is a unique sub-game equilibrium for every leader strategy set, and at the equilibrium the stakeholders play their optimal strategies, i.e.,

$$s_n^{*m} = B_n \frac{R_m}{\sum_{i=1}^{M} R_i}, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}. \tag{4.5}$$

This optimal strategy only depends on the stakeholder's total budget and the ratios of block rewards between the chains. As a result, every stakeholder has a

unique optimal strategy that maximizes its profits, and thus a rational stakeholder will always invest according to this strategy. In the next stage, we will analyze the leader strategy to determine the optimal block reward for the leaders.

4.3.2.2 Leader strategy

The proposed incentive mechanism for FedChain has two main aims. The first one is to attract stakes to improve the individual chain's performance and security. The second aim is to ensure the decentralization of the system, i.e., encourage the stakeholders to distribute their stakes evenly across all the chains. For these two aims, we propose a utility function U_m for the leaders as follows:

$$U_m = \sum_{n=1}^{N} \omega_m^n s_n^{*m} - R_m = \sum_{n=1}^{N} \frac{B_n R_m}{\sum_{i=1}^{M} R_i} \ln\left(\frac{B_n R_m}{\sum_{i=1}^{M} R_i}\right) - R_m, \tag{4.6}$$

where ω_m^n is a weight factor which can be defined by $\omega_m^n = \ln(s_n^{*m})$. By using the logarithm of the stakes as the weight factor, we can achieve two main aims. In particular, from this designed utility function, a leader can attract more stakes invested to its pool by increasing its block reward. However, at a certain level, if this leader keeps increasing its block reward to get more stakes, its utility will be decreased. As a result, this utility function encourages the chain operator to set an appropriate level of block reward such that it can attract sufficient stakes to the chain while ensuring that individual stakeholders do not control too much of the network stakes. Moreover, this also discourages the chain operators from setting a too high block reward that will cause the centralization of stakes into a single chain in FedChain. Then, we proceed to find the equilibrium of the upper sub-game and the Stackelberg equilibrium of the considered Stackelberg game in Theorem 4.7.

Theorem 4.7. *The point where every leader's strategy is* $R_m^* = \frac{M-1}{M^2} \sum_{n=1}^{N} B_n \Big(1 + \ln\big(\frac{B_n}{M}\big)\Big)$ *and every follower's strategy satisfies* $s_n^{*m} = B_n \frac{R_m}{\sum_{i=1}^{M} R_i}, \forall m \in \mathcal{M}, \forall n \in$

$\mathcal{N}$ is the unique Stackelberg equilibrium of the considered game. Moreover, the convergence to the Stackelberg equilibrium is guaranteed.

Proof. We solve $\frac{\mathrm{d}U_m}{\mathrm{d}R_m} = 0$ to find R_m^*. Since R_m^* is uniquely defined by constants, the equilibrium is unique. The detailed proof is provided in Appendix B.7. □

Interestingly, the result from Theorem 4.7 shows that the optimal strategies are the same for all the chain operators. The reason is that since stakes can be transferred, the security of the whole system is as strong as that of the weakest chain. Therefore, the highest utility can only be achieved when every chain is equally secure.

4.4 Performance Evaluation

In this section, we conduct experiments and simulations to (i) show that the proposed Stackelberg game can help the stakeholders to maximize their profits, (ii) confirm our analytical results, and (iii) demonstrate that the proposed incentive mechanism can enhance FedChain's security and performance. To this end, we first examine the utility function of a stakeholder to confirm our results from Theorem 6 and show that the Stackelberg game model can help to maximize the stakeholder's profit. After that, to evaluate the security and performance of the FedChain, we implement extensive simulations under various settings. In the simulations, we first show that the rational stakeholders will act according to our proposed Stackelberg game-theoretical analysis. We will then demonstrate that the FedChain's consensus mechanism can satisfy the security properties and attain reasonable performance even under extreme adversarial scenarios. Furthermore, we will show that under the same simulation setting, the proposed dynamic reward scheme achieves better security and performance compared to those of the static reward scheme.

4.4.1 Simulation Setting

First, we examine the utility function of stakeholder 1 in a small case which consists of two stakeholders and three chains. The stakeholders have budgets $\mathbf{B} = [100, 300]$, and the chains set block rewards to be $\mathbf{R} = [10, 20, 30]$. In this experiment, the strategy of stakeholder 2 is fixed according to (4.5). Then, we simulate a system with N stakeholders and M chains under different adversarial models (static and adaptive), reward schemes (static and dynamic), and different adversarial levels (weak, medium, and strong). The simulation parameters are presented in Table 4.2.

The simulation has several steps as presented in Algorithm 4.1. In particular, at the beginning, each stakeholder has a budget $B_i \in [\text{LB}, \text{UB}]$ generated randomly with uniform distribution. Each chain operator then sets a block reward R_m based on Theorem 7's result in the case of the dynamic reward scheme. In the static reward scheme, R_m are fixed as constants based on several real-world PoS blockchain networks [98, 99, 101]. After the block rewards are set, the stakeholders make their decisions. To find the best strategies for each stakeholder, we employ the Matlab fmincon function [113], starting from stakeholder 1. Then, the newly found optimal strategy is fixed for the stakeholder, and the algorithm continues to find the best response for stakeholder 2 until stakeholder N. After that, the adversary begins to attack. In the static adversary scenario, the adversarial stakes budget B_A is constant and predetermined. In the adaptive adversary scenario, the adversary chooses a number N_A of stakeholders to corrupt, making their stakes to be adversarial stakes, i.e., the adversarial stakes budget is $\sum_{i \in \mathcal{N}_A^S} B_i$. Then, we measure the impacts of the adversary on Pr_{CP}, Pr_{CQ}, transaction confirmation time, and transaction throughput. Finally, we simulate the stake changes by randomly choosing N_Δ stakeholders and changing their budgets by $\pm\Delta_s B_n, \Delta_s \in (0, 1)$. The epoch is then ended, and the simulation moves to the next epoch until the stopping criteria are met, i.e., after n_e epochs.

Algorithm 4.1 Simulation Steps

$k \leftarrow 0$
repeat
  if reward scheme = dynamic **then** ▷ Chains set block rewards at each epoch
    for $m := 1$ to M **do**
      $R^*_m \leftarrow \frac{M-1}{M^2} \sum_{n=1}^{N} B_n \left(1 + \ln\left(\frac{B_n}{M}\right)\right)$
    end for
  end if
  for $n := 1$ to N **do** ▷ Followers make decisions
    for $m := 1$ to M **do**
      Find s_n^{*m} using fmincon
    end for
  end for
  if Adversary = Static **then** ▷ Static Adversary
    Adversary attacks with fixed B_A
  else ▷ Adaptive Adversary
    Adversary corrupts N_A stakeholders
    Adversary attacks with $B_A = \sum_{i \in \mathcal{N}_A} B_i$
  end if
  for $i := 1$ to N_Δ **do** ▷ Randomly adjust followers' budgets
    Adjust a random follower budget by $\pm \Delta_s B_n$
  end for
  $k \leftarrow k + 1$
until $k > n_e$

During the simulation, we measure several important security and performance criteria. First, we measure the stake distribution at the beginning of each epoch

Table 4.2 : Parameter setting

Parameter	**Weak Adversary**	**Medium Adversary**	**Strong Adversary**
N	100	100	100
M	3	3	3
LB	50	50	50
UB	100	100	100
Δ_s	$(0, 1)$	$(0, 1)$	$(0, 1)$
B_A	500	1000	1500
N_A	10	20	30
n_e	10	10	10

to see if the rational stakeholders invest according to our game-theoretical analysis. Then, we examine four different scenarios. In the first two scenarios, we simulate a static adversary who will try to attack the chains under the static and dynamic reward schemes. In the remaining scenarios, an adaptive adversary will try to attack the chains. For each type of adversary, we simulate three different levels of adversary capacity (low, medium, and high) as shown in Table 1.

In terms of security, we measure the CP and CQ violation probabilities. These probabilities can be determined by (B.1) and (B.3), respectively. In terms of performance, we measure how much the adversaries can negatively impact the transaction confirmation time and transaction throughput. To calculate the transaction confirmation time, for each chain, we find the value of κ such that $\Pr_{\text{CP}} < 0.1\%$. For the transaction throughput, we want to examine the case where the adversary wants to reduce the transaction processing capability of one of the chains. Specifically, the

adversary will move all its stakes to a chain and participate in the leader selection process. For every block the adversary is elected to be the leader, it creates an empty block without any transaction, thereby reducing the network's transaction throughput. In the simulation, we measure a transaction throughput reduction threshold Θ, such that the probability that the adversary can reduce the transaction throughput more than Θ is overwhelmingly low (i.e., $\Pr_{\text{CQ}} < 0.1\%$).

4.4.2 Performance Results

4.4.2.1 Economical benefits

Fig. 4.5 illustrates the utility function of stakeholder 1 in the case where stakeholder 2 invest according to (4.5). As observed from the figure, stakeholder 1 can achieve maximum utility when it also invests according to (4.5). Particularly, stakeholder 1 achieves a utility $U_1^* = 15$ with the optimal strategy $\mathbf{s}_1^* = [16.6, 33.3, 50]$. This result shows that our Stackelberg game model can help the stakeholders to achieve maximum profits. Moreover, the ratios between s_1^{*1}, s_1^{*2}, and s_1^{*3} are the same as the ratios between R_1, R_2, and R_3, which confirms our results in Theorem 6.

4.4.2.2 Stake distribution

Fig. 4.6 illustrates the stake distribution at the end of each epoch. As can be seen from the figure, although the total number of stakes vary across the epochs, the ratio of stakes invested in each chain remains unchanged in both the dynamic and static reward schemes. Moreover, we can observe that the stakes are distributed more evenly in the dynamic reward scheme, which is more beneficial to the chains' security and performance. Furthermore, the stake ratios in both schemes equal the ratio of the block rewards, which confirms our analytical results in Theorem 6.

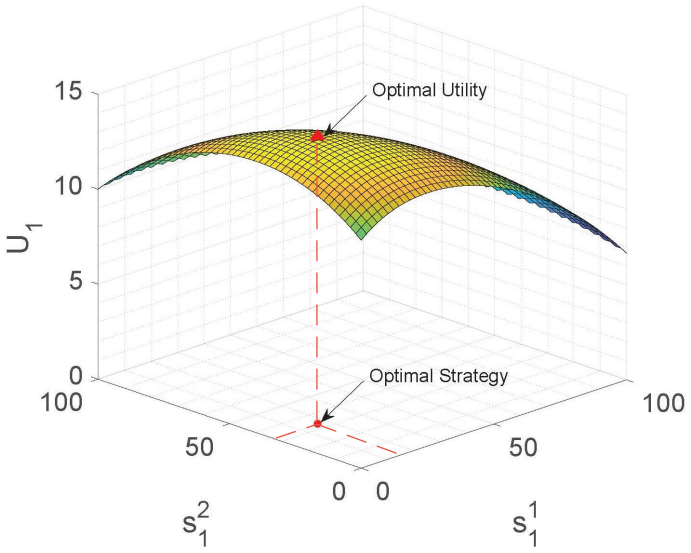


Figure 4.5 : Stakeholder's utility function.

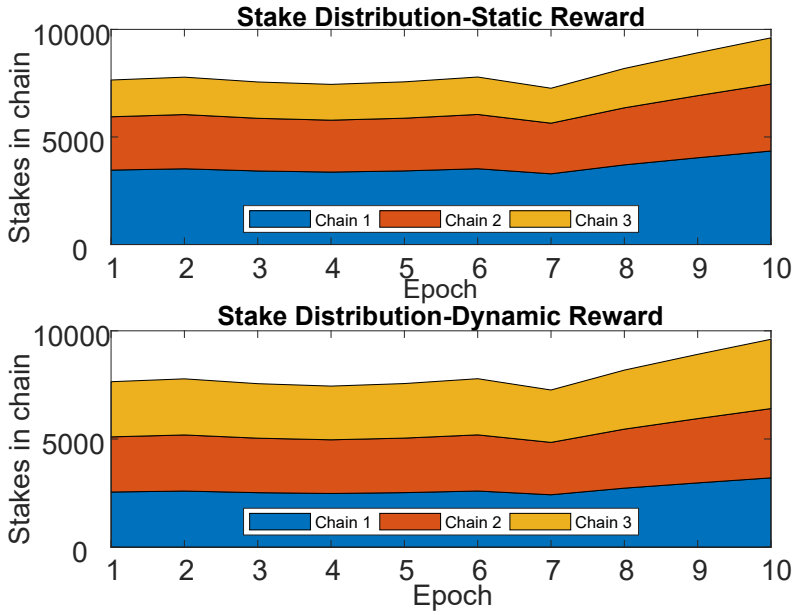


Figure 4.6 : Stake distribution.

4.4.2.3 Security properties

Fig. 4.7 and Fig. 4.8 illustrate $\Pr_{\text{CP}}$ of each chain at the end of each epoch under the static and adaptive adversary settings, respectively. From the figures, we can observe that the more stakes the adversary controls, the higher chance it can violate the security of the system. For example, in the static adversary setting, with a low budget (weak adversary), $\Pr_{\text{CP}}$ is at most 0.02%, whereas this probability increases to 1.5% in case of an adversary with a high budget (strong adversary). Secondly, the total system stakes have different effects on the chains' security under the static and adaptive adversary setting. For instance, the system has the highest stakes in the last epoch. At this epoch, $\Pr_{\text{CP}}$ achieve the lowest value under the static adversary because the static adversary has a fixed budget. However, $\Pr_{\text{CP}}$ achieve the highest value under the adaptive adversary setting because the adaptive adversary can corrupt the stakeholders with the most stakes. Therefore, it is crucial to not only attract more stakes to the system but also to incentivize more diversity, i.e., encourage the stakeholders to split their stakes across more chains. We can observe the effect of such diversity between the dynamic and the static reward schemes. Although the total network stakes are the same, the dynamic scheme, which encourages equal stakes distribution, achieves much lower $\Pr_{\text{CP}}$, e.g., at most 14% compared to 24% of the static reward scheme.

Fig. 4.9 and Fig. 4.10 illustrate $\Pr_{\text{CQ}}$ of each chain under the static and adaptive adversary settings, respectively. Similar to the $\Pr_{\text{CP}}$, we can draw several conclusions from examining $\Pr_{\text{CQ}}$. Firstly, the stronger the adversary is, the higher chance it violates system security. For example, in the weak adaptive adversary scenario, $\Pr_{\text{CQ}}$ is at most 1.2%, whereas this probability increases to 2.4% in the case of a strong adaptive adversary. Generally, $\Pr_{\text{CQ}}$ gets higher in the case of the adaptive adversary. The reason is that according to the simulation setting, the adversary can corrupt more stakes compared to B_A in the case of the static adversary. Secondly,

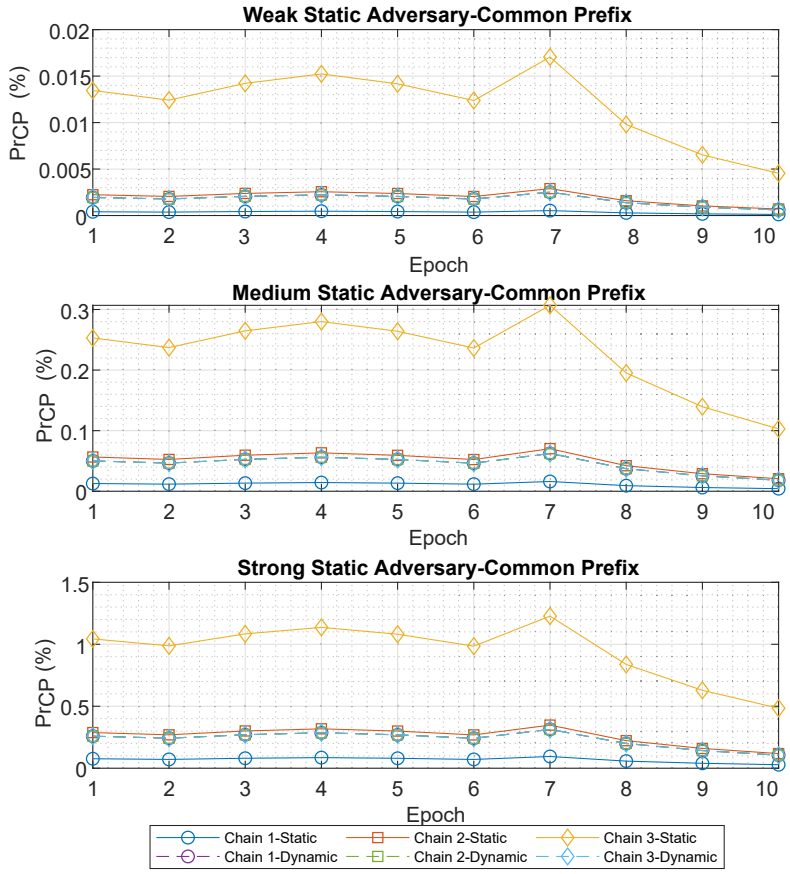


Figure 4.7 : $\mathrm{Pr}_{\mathrm{CP}}$ under static adversary settings.

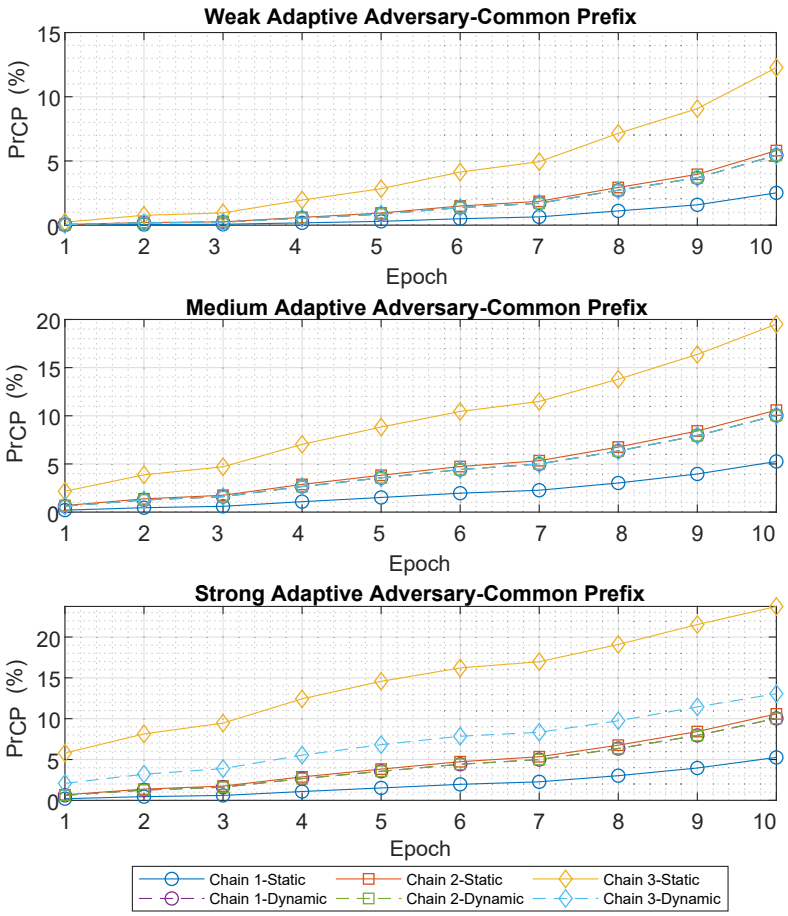


Figure 4.8 : $\mathrm{Pr}_{\mathrm{CP}}$ under adaptive adversary settings.

similar to the results of $\mathrm{Pr}_{\mathrm{CP}}$, $\mathrm{Pr}_{\mathrm{CQ}}$ is inversely proportional to the total system stakes in the case of the static adversary, and it is proportional to the total system stakes in the case of the adaptive adversary. As a result, we can observe that the dynamic scheme achieves lower $\mathrm{Pr}_{\mathrm{CQ}}$, e.g., at most 14% $\mathrm{Pr}_{\mathrm{CQ}}$ compared to 24%. Moreover, since the security of the system is only as good as that of its weakest chain (especially with the SPV proof mechanism), it can be observed that the dynamic reward scheme achieves better security compared to the static reward scheme, i.e., the chains of the dynamic reward scheme always achieve better $\mathrm{Pr}_{\mathrm{CP}}$ and $\mathrm{Pr}_{\mathrm{CQ}}$ compared to those of the weakest chain under the static reward scheme (i.e., Chain 3).

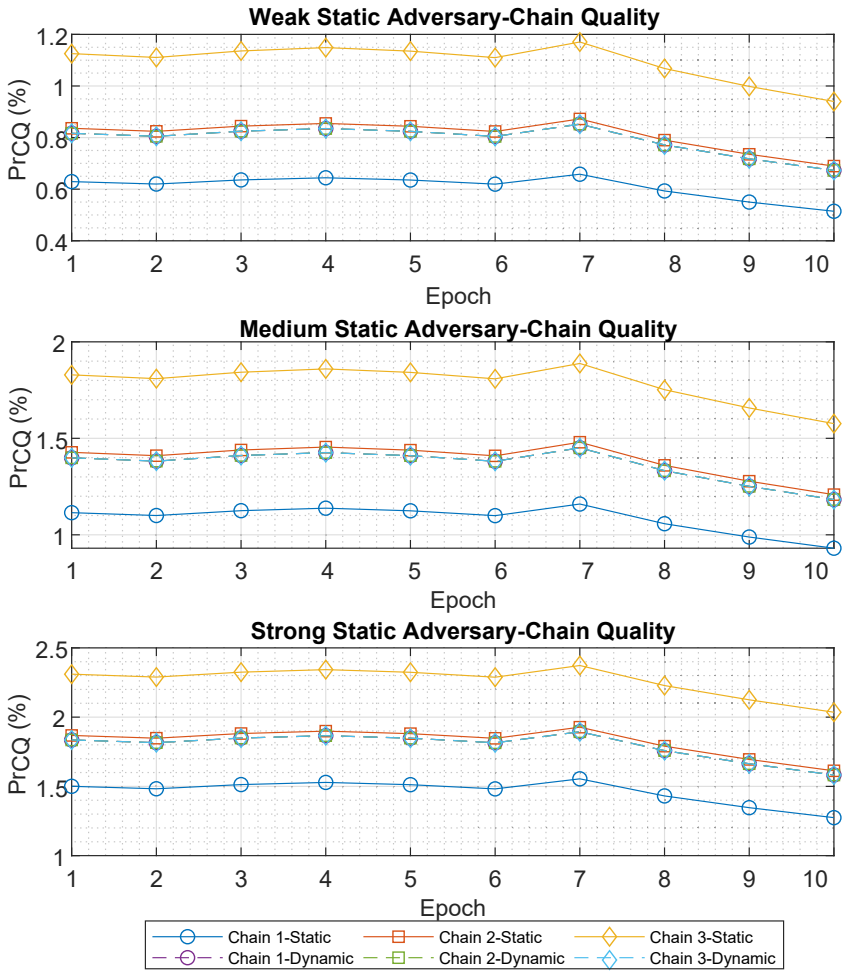


Figure 4.9 : $\mathrm{Pr}_{\mathrm{CQ}}$ under static adversary settings.

4.4.2.4 Performance properties

Fig. 4.11 and Fig. 4.12 illustrate the transaction confirmation time of each chain under the static and adaptive adversary settings, respectively. From the figures, we can observe that the stronger the adversary is, the more it can negatively affect the system performance. For example, the chains takes at most 120 seconds to confirm a transaction in case of a weak static adversary, but it takes up to 220 seconds in case of a strong static adversary. This is because the transaction confirmation time is directly related to $\mathrm{Pr}_{\mathrm{CP}}$. A stronger adversary has a higher chance to violate the CP property, and thus the users have to wait longer to confirm a transaction. Moreover,

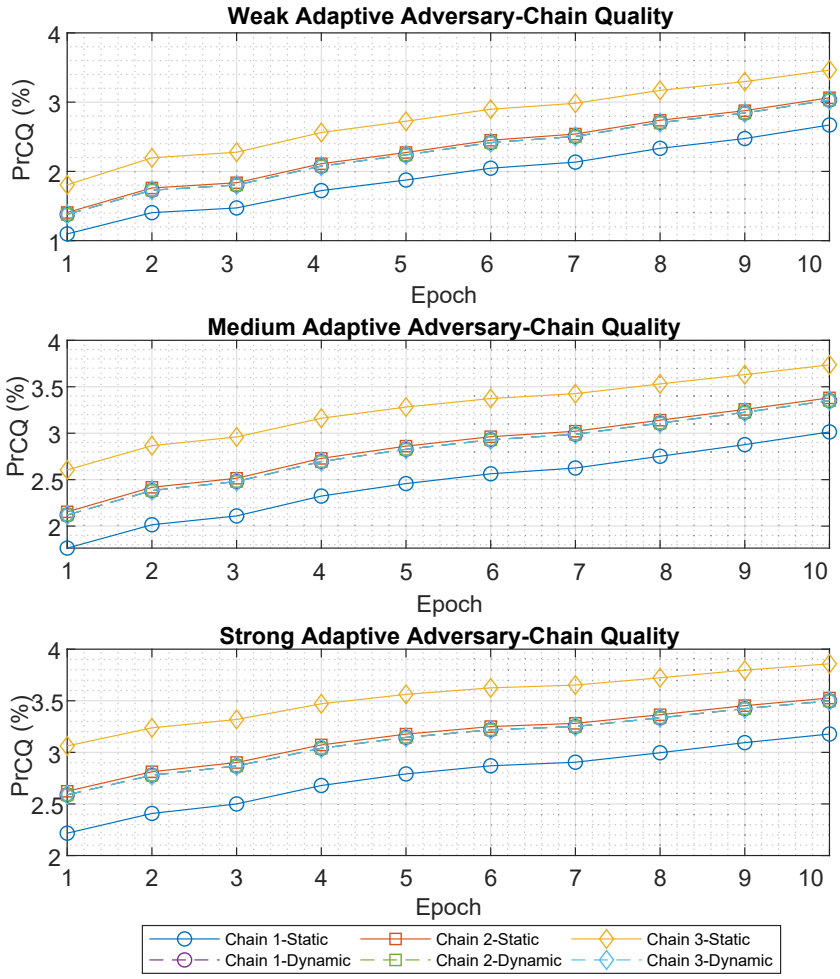


Figure 4.10 : $\mathrm{Pr}_{\mathrm{CQ}}$ under adaptive adversary settings.

we can also observe that the transaction confirmation time is inversely proportional to the total stakes of the system in the static adversary settings, whereas the opposite holds true in the adaptive adversary settings. The reason is the same as that of the $\mathrm{Pr}_{\mathrm{CP}}$ scenarios, i.e., the adaptive adversary can corrupt more stakes, whereas B_A of the static adversary is fixed. Furthermore, the transaction confirmation time of the three chains under the dynamic reward schemes is always better than at least two chains under the static reward scheme.

Fig. 4.13 and Fig. 4.14 illustrate the transaction throughput reduction percentages of each chain under the static and adaptive adversary setting, respectively.

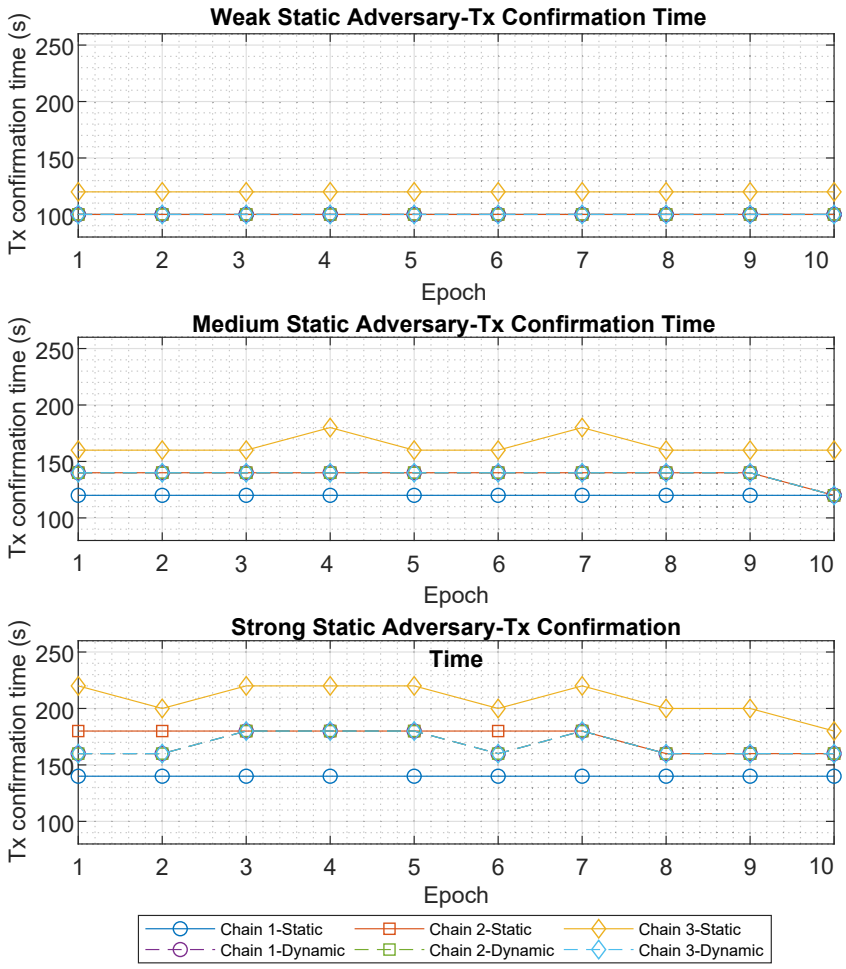


Figure 4.11 : Transaction confirmation time under static adversary settings.

Similar to the previous scenarios, we can observe that a stronger adversary can cause more negative impacts on the system performance, e.g., a weak static adversary can reduce the throughput by at most 24%, whereas the strong static adversary can reduce the throughput by nearly 50%. Moreover, it can be observed that as the system has more stakes, the static adversary becomes weaker, whereas the adaptive adversary becomes stronger, similar to the previous scenario. Finally, one can observe that the dynamic reward scheme can achieve a better overall performance compared to that of the static reward scheme (the performances of the three chains in the dynamic scheme are better than those of at least two chains in the static

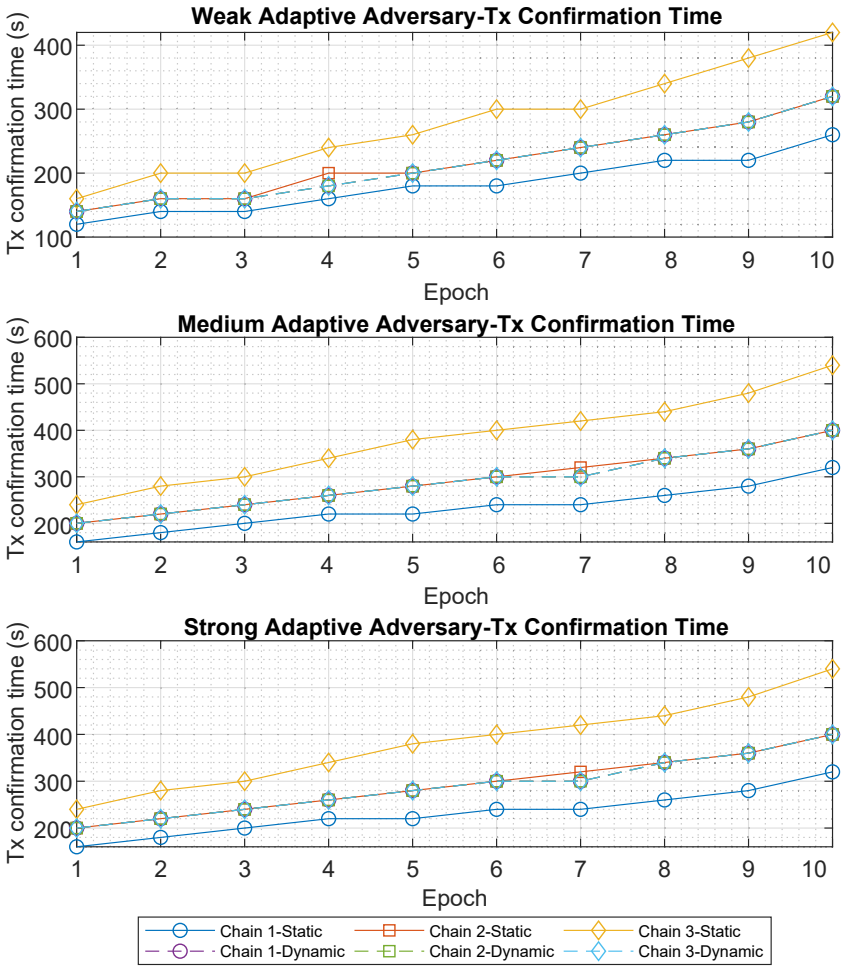


Figure 4.12 : Transaction confirmation time under adaptive adversary settings.

scheme).

4.5 Conclusion

In this chapter, we have introduced FedChain, an effective framework for federated-blockchain systems together with a cross-chain transfer protocol to facilitate the secure and decentralized transfer of tokens between the blockchains. In this framework, we have proposed a novel consensus mechanism which can satisfy the CP, CG, and CQ properties, prevent various blockchain-specific attacks, and achieve better transaction confirmation time compared to existing consensus mechanisms. Robust

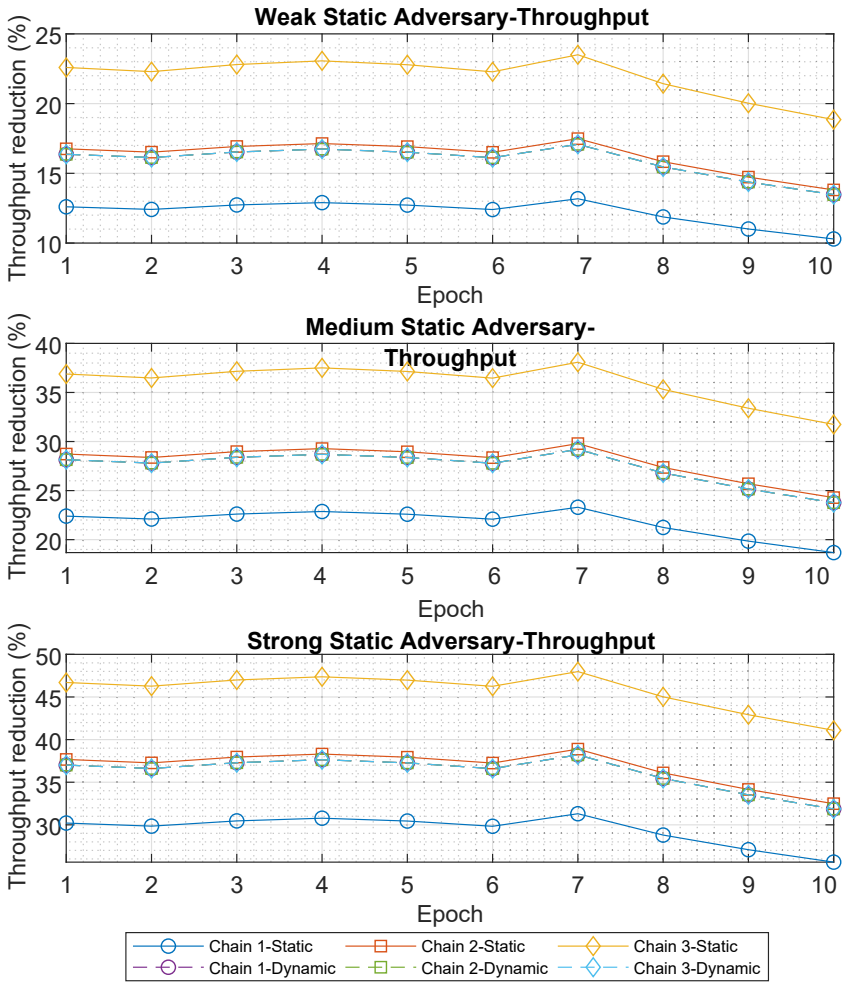


Figure 4.13 : Transaction throughput under static adversary settings.

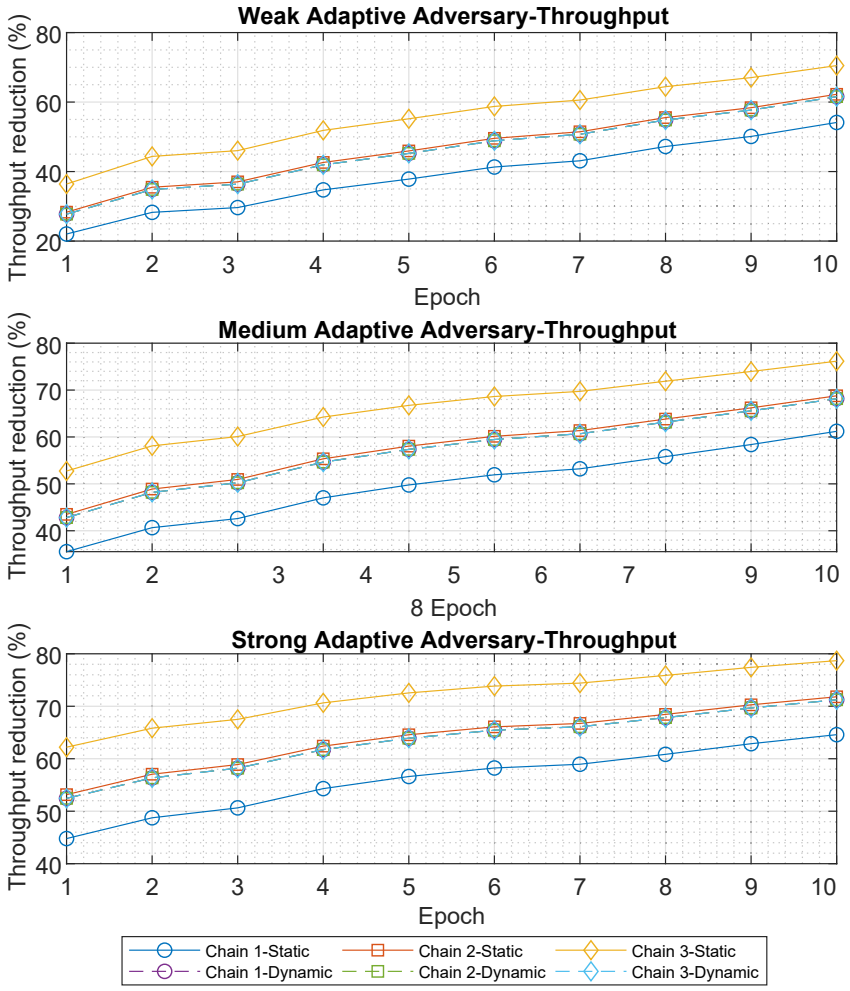


Figure 4.14 : Transaction throughput under adaptive adversary settings.

theoretical analyses have been then conducted to prove FedChain's consensus mechanism security and performance properties. After that, a Stackelberg game model has been developed to examine the interactions between the stakeholders and the blockchains managed by chain operators. This model can provide additional profits for the stakeholders and enhance the security and performance of the blockchains. Through analyses of the Stackelberg game model, we can prove the uniqueness of the Stackelberg equilibrium and find the exact formula for this equilibrium. These results are especially important for the stakeholders to determine their best investment strategies and for the chain operators to design the optimal policy, i.e., block rewards. Finally, extensive experiments and simulations have been conducted to show that our proposed framework can help stakeholders to maximize their profits and the chain operator to design appropriate parameters to enhance FedChain's security and performance.

Chapter 5

MetaShard: A Novel Sharding Blockchain Platform for Metaverse Applications

Due to its security, transparency, and flexibility in verifying virtual assets, blockchain has been identified as one of the key technologies for Metaverse. Unfortunately, blockchain-based Metaverse faces serious challenges such as massive resource demands, scalability, and security/privacy concerns. To address these issues, this chapter proposes a novel sharding-based blockchain framework, namely MetaShard, for Metaverse applications. Particularly, we first develop an effective consensus mechanism, namely Proof-of-Engagement, that can incentivize MUs' data and computing resource contribution. Moreover, to improve the scalability of MetaShard, we propose an innovative sharding management scheme to maximize the network's throughput while protecting the shards from 51% attacks. Since the optimization problem is NP-complete, we develop a hybrid approach that decomposes the problem (using the binary search method) into sub-problems that can be solved effectively by the Lagrangian method. As a result, the proposed approach can obtain solutions in polynomial time, thereby enabling flexible shard reconfiguration and reducing the risk of corruption from the adversary. Extensive numerical experiments show that, compared to the state-of-the-art commercial solvers, our proposed approach can achieve up to 66.6% higher throughput in less than 1/30 running time. Moreover, the proposed approach can achieve global optimal solutions in most experiments.

The rest of this chapter is organized as follows. Section 5.1 presents MetaShard's system overview. The proposed PoC consensus mechanism and sharding management scheme are presented in detail in Section 5.2. Section 5.3 presents the sharding

management problem and our proposed lightweight approach, and its performance is evaluated in Section 5.4. Finally, conclusions are drawn in Section 5.5.

5.1 System Overview

5.1.1 System Overview

Fig. 5.1 illustrates an overview of the proposed MetaShard framework. In this framework, there is an MSP operating a Metaverse running multiple Metaverse applications. Each Metaverse application is a self-contained environment that offers

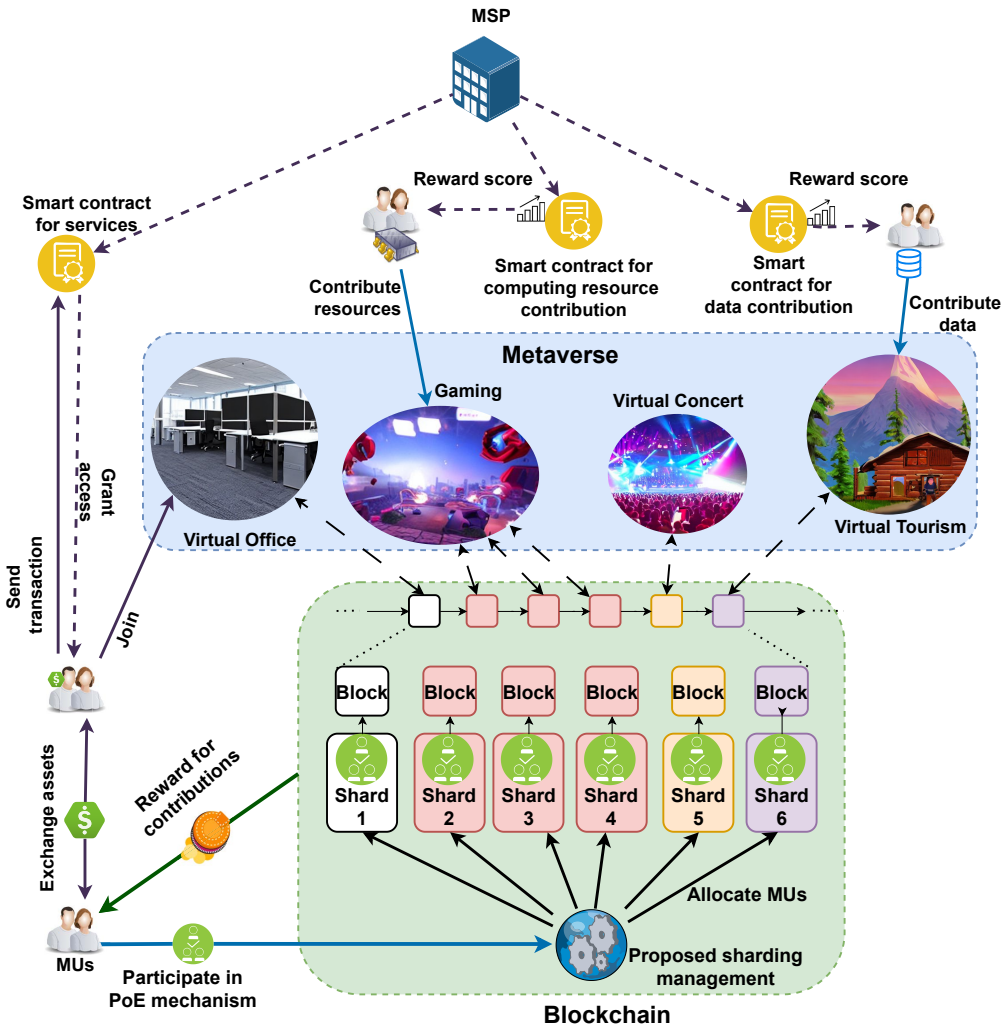


Figure 5.1 : An illustration of the proposed system

a wide range of services and experiences, e.g., virtual office, virtual concerts, gaming, and virtual tourism, for MUs. Compared to traditional virtual applications, the core difference here is that applications in the Metaverse are fully interconnected, allowing the MUs to freely and seamlessly move between different applications, e.g., Meta Horizon Worlds [114]. The MUs also have various interactions with each other and the MSP, such as exchanging assets, purchasing services and items, contributing resources, and participating in the blockchain's consensus process. A blockchain-based system can be applied to record and facilitate those interactions.

5.1.2 Metaverse Users and Metaverse Service Provider

An MU is a user that can join and use different Metaverse applications and services provided by the MSP. The MUs have unique avatars that represent them in the Metaverse, allowing them to interact with each other as well as the virtual worlds. There can be various interactions among the MUs, as well as between the MUs and the MSP. First, the MUs can easily exchange digital assets, such as Metaverse tokens and virtual items, with each other using blockchain transactions. For example, MUs who purchased virtual concert tickets (but could not attend) can sell their tickets to others. All these digital assets and transactions can be verified and stored in the blockchain, providing a secure transparent way to manage assets without the need for a central authority.

Moreover, the MUs can pay the MSP to gain access to services or buy digital items. This process can be automated by smart contracts, i.e., a user-defined program that can be automatically executed when the conditions within are met [10]. For example, the MSP can broadcast its virtual meeting options, e.g., duration, number of people, and fees, by publishing a smart contract on the blockchain. Then, MUs who want to purchase this service can send a transaction that contains the specified options to the smart contract. After the transaction is validated, the smart contract

can automatically send the MU a transaction that contains a proof for the purchase. When the MU requests to enter the virtual meeting room, the Metaverse application can query the blockchain to verify the proof and grant the involved MUs access to the room.

Furthermore, in our proposed MetaShard, MUs can also contribute data or computing resources to Metaverse applications. For example, in Metaverse virtual tourism applications, the MSP needs up-to-date 3D image/video data from tourist attractions to provide more immersive experiences to MUs. In this scenario, the MSP can encourage MUs who live near the tourist attraction to contribute the data, thereby saving costs and increasing MUs' engagement. Moreover, in compute-intensive AR/VR applications, the MSP can incentivize MUs to execute the rendering locally instead of offloading to the MSP's servers. Additionally, the MSP can offload computing tasks to MUs with idle resources to alleviate the heavy burden on the edge/cloud servers. For those contributions, MUs can be rewarded with digital assets such as Metaverse items or tokens. This can help to encourage more MUs to participate in the Metaverse and alleviate the high resource demands of Metaverse applications. Similarly, smart contracts can be utilized to provide a transparent and trusted way to reward the MUs for their contributions because the conditions written within a smart contract are visible to everyone. For example, the MSP can publish a smart contract that specifies the payment for different amounts of data contributed. When the MUs send the data to the smart contract, they can be automatically paid for their data.

5.1.3 Blockchain and Sharding

In MetaShard, the blockchain serves as a platform to store and manage MUs and applications data, interactions, and assets. Blockchain enables the MUs and the MSP to manage their identities, avatars, and digital assets in a decentralized

manner, thereby significantly enhancing transparency and trust for MUs. Moreover, smart contracts can automate and facilitate various interactions among MUs and applications. Furthermore, the blockchain can also provide a transparent way to manage and reward MUs' data and computing resources contribution, thereby creating a more engaged and motivated MUs community. However, managing such a huge amount of data and interactions for many MUs requires very high transaction processing capabilities, which conventional blockchain technology cannot handle. Particularly, most current blockchain networks are still employing the PoW consensus mechanism which consumes a huge amount of energy and has very low processing capability.

Therefore, we propose a PoS-based consensus mechanism for MetaShard. With PoS, the energy consumption is negligible, and the transaction processing capability can be significantly improved. Moreover, different from the conventional PoS that only considers the user assets (stakes), we develop a PoE consensus mechanism that will also take into account MUs' data and resources contribution and reward MUs for their engagement. In this way, PoE can not only leverage MUs' resources to alleviate the massive resource demands for the MSP but also encourage more MUs to join the Metaverse for the rewards, thereby creating a more engaged MUs community. This PoE consensus mechanism will be discussed in detail in Section 5.2. Moreover, scalability is a major constraint that hinders the applicability of conventional blockchain technology for Metaverse applications with a huge number of MUs. Therefore, we propose to employ the sharding mechanism [115, 116] for MetaShard. With sharding, the blockchain network can be divided into multiple smaller networks that allow the parallel processing of transactions and smart contracts, thereby improving scalability and processing speed and reducing the workload on individual consensus nodes. Furthermore, each Metaverse application can be adaptively allocated a different number of shards according to their processing demands. For

example, we can allocate more shards to virtual office applications during working hours and more shards to virtual concert applications at night.

Although dividing a blockchain into shards can significantly enhance the network throughput (in terms of the number of transactions successfully verified and processed per time unit), it also causes some potential risks for network security as shown in [115, 116]. Particularly, the security of a blockchain network depends on the honest majority. For example, if the adversary can control the majority (51%) of stakes in PoS, it can successfully perform various attacks, such as double-spending and transaction denial attacks [3, 6, 7], on the network. However, if the stakes are not allocated properly into the shards, then the adversary may not need too many stakes to successfully attack a shard. Therefore, it is crucial to determine the proper number of shards and MUs allocation such that the security of the whole network is still ensured. To this end, in Section 5.3, we will formulate this sharding management optimization problem and propose an efficient approach to quickly obtain solutions, thereby significantly improving the network performance and security.

5.2 Proposed PoC Consensus Mechanism and Sharding

5.2.1 Epoch and time slots

In our proposed PoE consensus mechanism, time is divided into epochs. Each epoch is then divided into time slots. During epoch e_k, our proposed sharding management process is executed to determine the number of shards and MUs allocation for epoch e_{k+1}, as illustrated in Fig. 5.2. Note that frequent and dynamic adjustment of the number of shards can be beneficial for the system, e.g., adding more shards to address the varying transaction processing demands or closing shards to reduce unnecessary communication [117]. This sharding management process is run once for each epoch, which is beneficial for network security [115, 116]. Moreover, it is also more desirable for the MUs, e.g., an MU who contributes more in this epoch

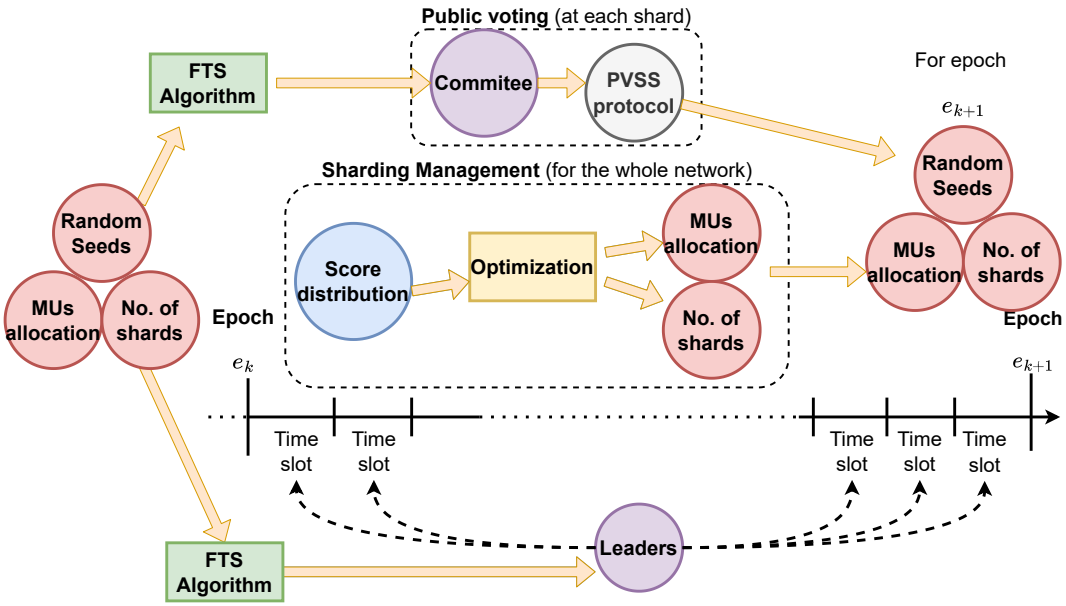


Figure 5.2 : An illustration of the proposed sharding management and election processes.

should have a higher chance to be elected as a leader and earn block rewards (e.g., in Metaverse tokens) in the next epoch.

Moreover, during the epoch, the committee members (selected from MUs who participate in the consensus process) of each shard execute the Publicly Verifiable Secret Sharing (PVSS) protocol [91] to create random seeds. The PVSS protocol is guaranteed to produce unbiased random strings, and it allows network participants to verify those strings, as long as 51% of the protocol participants are honest [91]. Therefore, the PVSS protocol can be employed to create publicly verifiable random seeds. At the beginning of each epoch, these random seeds, along with the number of shards and MUs allocation, are then used as the input of a hash function, e.g., Follow-the-Satoshi (FTS) algorithm [6], to choose the leaders for the current epoch and committee members for the next epoch. If the numbers of shards of two epochs are different, the random seeds can be used to determine which shard will create more (or fewer) random seeds. For example, if there is one more shard in the next

epoch, then a random shard in this epoch will create two seeds instead of one.

5.2.2 MU Engagement and Reward

In MetaShard, MUs are incentivized to contribute data and computing resources. To reward this contribution, MUs are given contribution scores that are stored in the blockchain. These scores are then used along with the MUs assets, e.g., Metaverse items and tokens, to determine the MUs' total engagement scores. Particularly, each MU has a data contribution score D_n, a computing resource contribution score C_n, and an amount of Metaverse tokens T_n. The data and computing resource score rewarded to the MUs can be determined by the MSP, e.g., based on the amount or frequency of resources and data contribution [118, 119]. The total engagement score of MU n can be calculated by

$$\eta_n = \alpha_D D_n + \alpha_C C_n + \alpha_T T_n, \tag{5.1}$$

where α_D, α_C, and α_D are the weight factors for data contribution, computing resources contribution, and Metaverse token, respectively. These weight factors are determined by the MSP, and they can also reflect the MSP's priority. For example, if the MSP needs more computing resource contribution, it can set α_C higher than α_T and α_D.

Every MU can choose to participate in the consensus processes to be able to earn the block rewards. Since each shard runs its own consensus process, the probability that MU n is selected to be the leader of shard s is given by:

$$\mathrm{Pr}_{\mathrm{n}}^{\mathrm{s}} = \frac{\eta_{\mathrm{n}}^{\mathrm{s}}}{\sum_{\mathrm{i}=1}^{\mathrm{N}} \eta_{\mathrm{i}}^{\mathrm{s}}}. \tag{5.2}$$

Besides the benefits of MUs' resource contribution, our proposed leader selection approach can also enhance the security of the network. The reason is that MUs who are more engaged (with high contributions and own a lot of assets) might want to protect the network more. Moreover, in existing approaches such as [49, 50, 52–55, 120, 121],

the leader is not selected based on stakes/scores (BFT-based approaches). Instead, these approaches rely only on the number of validators. However, the adversary can target those protocols by conducting Sybil attacks, i.e., creating multiple accounts, to improve their chance of being selected as validators. In contrast, the leaders are chosen based on their engagement in MetaShard, and thus creating multiple accounts with no contributions or assets cannot adversely affect the leader selection process.

5.2.3 Threat Model and Shard Security

Threat Model: In this work, we consider the type of adversary that tries to gain the majority in any shard to conduct 51% attacks. Particularly, the adversary possesses multiple accounts (adversarial MUs) in the system. These accounts, along with the other MUs' accounts, are allocated into different shards in the system. If the total score of the adversary exceeds 51% of the total score of any shard in the system, the adversary can successfully conduct various attacks, such as double-spending and transaction denial attacks [3, 6, 7], and unfairly affect the seeds generation of the PVSS protocol. Moreover, the adversary can corrupt honest MUs, but the corruption will take effect after a period of time [49, 50, 54, 115]. When an MU is corrupted, it will be controlled by the adversary, and its score will count toward the adversary's total score.

Given the above adversary model, there are two serious threats. First, when the adversary controls more than 51% of a shard, the adversary can influence the leader election process to conduct other types of attacks such as double-spending and transaction denial attacks [3, 6, 7] on the shard. Consequently, the Metaverse transactions might be reverted, or transactions from specific MUs might be blocked by the adversary. Therefore, it is crucial to allocate scores to each shard such that the adversary has a minimal chance to attack every shard. However, a major

challenge is that we do not know which MU is adversarial, and thus we can only minimize the chance that the adversary can control the majority of scores in any shard. Second, if the epoch is too long, the adversary might be able to corrupt the honest MUs during the epoch and successfully gain control of the shard. Therefore, the score allocation needs to be regularly reconfigured, e.g., Ethereum's epoch only lasts for 6.4 minutes [122].

To address these threats, we develop a sharding management approach to determine the number of shards and allocate MUs scores such that the adversary's chance to successfully attack any shard is minimal, e.g., lower than 0.1%. Moreover, the proposed approach can quickly obtain solutions, thereby reducing the time for the adversary to corrupt honest MUs. The proposed approach is presented in detail in the next section.

5.3 Sharding Management Problem and Solution

5.3.1 Problem Formulation

We first formulate the sharding management problem as follows. In the considered system, there is a set $\mathcal{N} = (1, \ldots, N)$ of MUs. Since we do not know which MU is adversarial, we can consider the total engagement score of the adversary, denoted by η_s^A, to be a sum of independent random variables. Let p_n^A denote the probability that MU n is adversarial. p_n^A can be determined based on the MUs' assets and contribution, i.e., MUs who owns more assets or contribute frequently to the Metavese are less likely to be adversarial, or using Machine Learning approaches such as those in [123–125]. The expected value of the total engagement score of the adversary in shard s can then be determined by:

$$\mathbb{E}[\eta_s^A] = \mathbb{E}[\sum_{n=1}^{N} p^A \eta_n^s] = \sum_{n=1}^{N} p^A \eta_n^s. \tag{5.3}$$

Since η_s^A is a sum of independent random variables, we want to determine the probability that η_s^A exceeds 50% of the total engagement scores in any shard, i.e., when the adversary gains the majority in a shard. To find this probability, we apply the Hoeffding bound [126] to determine the bounds on the tail distribution of η_s^A . Particularly, let $\theta_s = \sum_{n=1}^{N} \eta_n^s$ denote the total engagement score of all MUs (including the adversary) in shard s. Based on (5.3), the probability that the adversary's score exceeds 50% of the total scores in shard s can be determined by:

$$\Pr[\eta_s^A \geq 0.5\theta_s] = \Pr[\eta_s^A \geq \mathbb{E}[\eta_s^A] + t] \leq \exp\Big(\frac{-2t^2}{\sum_{n=1}^{N}(\eta_n^s)^2}\Big) \tag{5.4}$$

where t denotes the deviation from the expected value of η_s^A such that the adversary can gain majority in the shard, i.e., $\eta_s^A \geq 0.5\theta_s$. This deviation can be determined by:

$$\begin{aligned} 0.5\theta_s &= \mathbb{E}[\eta_s^A] + t, \\ \sum_{n=1}^{N} 0.5\eta_n^s &= t + \sum_{n=1}^{N} p_n^A \eta_n^s, \\ t &= \sum_{n=1}^{N} (0.5 - p_n^A)\eta_n^s. \end{aligned} \tag{5.5}$$

The inequality in (5.4) comes from Hoeffding bound [126]. To keep the probability in (5.4) lower than a certain safety threshold τ (e.g., $\tau = 0.001$), we have

$$\begin{aligned} \exp\Big(\frac{-2t^2}{\sum_{n=1}^{N}(\eta_n^s)^2}\Big) &\leq \tau, \\ -2\frac{\Big(\sum_{n=1}^{N}(0.5 - p_n^A)\eta_n^s\Big)^2}{\sum_{n=1}^{N}(\eta_n^s)^2} &\leq \ln(\tau), \\ \Big(\sum_{n=1}^{N}(0.5 - p_n^A)\eta_n^s\Big)^2 &\geq -0.5\ln(\tau)\sum_{n=1}^{N}(\eta_n^s)^2. \end{aligned} \tag{5.6}$$

This means that to make all the shards to be secured, we need to allocate the scores η_n^s of the MUs in each shard such that they satisfy (5.6). Let S denote the maximum possible number of shards*. We formulate the optimal sharding management

*In theory, we do not have the maximum possible number of shards, e.g., an MU can participate in many shards. However, in practice, this number cannot be unlimited because an MU does not

problem **(P1)** below.

$$(\mathbf{P1}) \max_{\boldsymbol{\eta},\mathbf{x},\varsigma} T\varsigma \tag{5.7}$$

$$\text{s.t. } (\sum_{n=1}^{N}(0.5-p_n^A)\eta_n^s)^2 \geq -0.5x_s \ln(\tau) \sum_{n=1}^{N}(\eta_n^s)^2, \qquad \forall s=1,\ldots,S \tag{5.8}$$

$$x_s \geq \frac{\varsigma - s + 1}{S}, \qquad \forall s=1,\ldots,S \tag{5.9}$$

$$x_s \leq \varsigma - s + 1, \qquad \forall s=1,\ldots,S \tag{5.10}$$

$$\sum_{s=1}^{S} \eta_n^s = \eta_n, \qquad \forall n \in \mathcal{N}. \tag{5.11}$$

In (**P1**), the objective (5.7) is to maximize the total network throughput, which can be obtained by multiplying the number of shards ς with the maximum number of transactions that a shard can process per second T. Constraints (5.8) follow (5.6). Note that out of these S constraints, only ς constraints are active to ensure the security for ς shards, while the constraints for the other (dummy) shards need to be inactive. To this end, we use auxiliary decision variables $\mathbf{x}$ to make the constraints active for the shards from 1 to ς, and inactive for the other shards. Particularly, constraints (5.9) and (5.10) ensure that $x_s = 1, \forall s = 1,\ldots,\varsigma$, while $x_s = 0, \forall s = \varsigma,\ldots,S$. Then, for shards from 1 to ς, the right-hand-side of constraints (5.8) become $-0.5x_s \ln(\tau) \sum_{n=1}^{N}(\eta_n^s)^2$ (active). For shards from $\varsigma+1$ to S, the right-hand-side of constraints (5.8) become zero, and thus they are always satisfied (inactive). Finally, constraints (5.11) ensure that the MUs scores are fully allocated. The reason for these constraints is that the MUs' rewards for consensus participation are proportional to their engagement scores, and thus the MUs will want to use all their scores for consensus participation.

---

want to participate in too many shards (same rewards but needs much more computational and communication resources).

From (5.8), we can observe that (**P1**) is a Mixed Integer Non-linear Programming (MINLP) problem [96, 127] which is NP-complete [128]. As later shown in Section 5.4, commercial solvers such as CPLEX [127] can only solve instances of (**P1**) with a small number of shards. For larger values of S, it becomes intractable and infeasible to obtain optimal solutions. However, the score allocation needs to be regularly reconfigured, e.g., Ethereum's epoch only lasts for 6.4 minutes [122]. Such frequent shard reconfiguration can bring various benefits. First, the MUs who contribute more resources in one epoch can have their scores updated earlier to earn more rewards in the next epoch. Moreover, if the epoch is short, the adversary will have less time to corrupt the honest MUs.

5.3.2 Proposed Hybrid Algorithm

5.3.2.1 Problem decomposition and the proposed Lagrangian approach

To address the abovementioned problems, we develop a lightweight approach based on Lagrange multipliers and binary search that can quickly obtain solutions in a very short time, thereby enabling flexible scores reallocation and improving the shards' security. To that end, we first decompose (**P1**) into multiple relaxed sub-problems (**P2**) as follows:

$$(\mathbf{P2}) \max_{\boldsymbol{\eta}} T\sigma \tag{5.12}$$

$$\text{s.t. } \Big(\sum_{n=1}^{N}(0.5 - p_n^A)\eta_n^s\Big)^2 \geq -0.5\ln(\tau)\sum_{n=1}^{N}(\eta_n^s)^2, \qquad \forall s = 1, \ldots, \sigma \tag{5.13}$$

$$\sum_{s=1}^{\sigma}\eta_n^s = \eta_n, \qquad \forall n \in \mathcal{N} \tag{5.14}$$

Particularly, in (**P2**), we fix the value of $\varsigma = \sigma$. In this way, we do not need to determine ς and $\mathbf{x}$, and thus constraints (5.8) become constraints (5.13). Moreover, constraints (5.9) and (5.10) can be omitted. Furthermore, the objective function (5.12)

becomes a constant, and thus we only need to find a feasible solution to (**P2**), instead of optimizing it. As a result, (**P2**) becomes a Nonlinear Programming (NLP) problem, which is easier to solve compared to MINLP problems [96, 127]. Then, we can solve (**P2**) for all values of $\sigma = 1, \ldots, S$, and the largest value of σ for which we can find a feasible solution is the global optimal solution of (**P1**). Nevertheless, (**P2**) is non-convex and nonlinear due to (5.13), and thus it still requires exponential time to solve [129], as later shown in Section 5.4.

To address this limitation, we reformulate the optimization problem (**P3**) as follows:

$$(\mathbf{P3}) \max_{\boldsymbol{\eta}} \sum_{s=1}^{\sigma} \left(\Big(\sum_{n=1}^{N} (0.5 - p_n^A) \eta_n^s \Big)^2 + 0.5 \ln(\tau) \sum_{n=1}^{N} (\eta_n^s)^2 \right) \tag{5.15}$$

$$\text{s.t.} \quad \sum_{s=1}^{\sigma} \eta_n^s = \eta_n, \qquad \forall n \in \mathcal{N} \tag{5.16}$$

The core idea of (**P3**) is that, instead of finding feasible solutions that satisfy (5.13) and (5.14), we try to maximize the left-hand-side of (5.13), subject to (5.14). Then, we can check the optimal solution $\boldsymbol{\eta}'$ obtained from (**P3**). Then, we adopt the Lagrange multipliers method to solve (**P3**) as follow. We first define the Lagrange function:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\eta}, \boldsymbol{\lambda}) &= f(\boldsymbol{\eta}) - \boldsymbol{\lambda} g(\boldsymbol{\eta}), \\ &= \sum_{s=1}^{\sigma} \left(\Big(\sum_{n=1}^{N} (0.5 - p_n^A) \eta_n^s \Big)^2 + 0.5 \ln(\tau) \sum_{n=1}^{N} (\eta_n^s)^2 \right) - \sum_{j=1}^{N} \lambda_j \left(\sum_{s=1}^{\sigma} \eta_n^s - \eta_n \right). \end{aligned} \tag{5.17}$$

Then, we solve the following set of equations:

$$\nabla_{\boldsymbol{\eta}, \boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\eta}, \boldsymbol{\lambda}) = 0, \tag{5.18}$$

which is equivalent to

$$\begin{aligned} \lambda_k + \ln(\tau) \sum_{n=1}^{N} (\eta_n^s) + 2 p_k^A \sum_{n=1}^{N} (0.5 - p_n^A) \eta_n^s = 0, \forall k \in \mathcal{N}, \forall s = 1, \ldots, \sigma, \\ \sum_{s=1}^{\sigma} \eta_n^s - \eta_n = 0, \forall n \in \mathcal{N}. \end{aligned} \tag{5.19}$$

Instead of solving the NLP problem (**P2**), we only need to solve (5.19) which is a set of $(\sigma+1)N$ equations with $(\sigma+1)N$ variables. Moreover, in (5.19), all the equations are linear, and thus it is a system of linear equations. As a result, this system of equations can be solved effectively in a very short period of time compared to (**P2**).

Finally, we implement Algorithm 5.1 which combines binary search and the Lagrange multiplier method to obtain optimal solutions for the original problem (**P1**). Particularly, Algorithm 5.1 first finds the optimal solution $\boldsymbol{\eta}'$ of (**P3**), using the system of equations in (5.19), with $\sigma = S$. Then, if $\boldsymbol{\eta}'$ satisfies (5.13), it is the optimal solution of (**P1**). Otherwise, we apply binary search to speed up the optimization process as illustrated in Fig. 5.3. Particularly, we first set $high = S - 1$, $low = 2$, and $\sigma' = (high + low)/2$. Then, we solve (5.19) to find $\boldsymbol{\eta}'$. Next, if $\boldsymbol{\eta}'$ satisfies (5.13) (which means σ is the best solution so far), we set $high = S - 1$, $low = \sigma$, and $\sigma' = (high + low)/2$. Otherwise, we set $high = \sigma$, $low = 2$, and $\sigma' = (high + low)/2$. In both cases, the loop is repeated until $\sigma' = high$. During the loop, the algorithm records the best solution found (that can satisfy (5.13)) in $\boldsymbol{\eta}^*$ and σ^*, and it will return $\boldsymbol{\eta}^*$ when the loop ends. With $\boldsymbol{\eta}^*$ and σ^*, x^* can be straightforwardly deduced for the original problem (**P1**) as shown in the proof of Theorem 5.1.

5.3.2.2 Optimality analysis

In Lemma 5.1, we first prove that the solution obtained from solving (5.19) is the global optimal solution of (**P3**).

Lemma 5.1. *Let $\boldsymbol{\eta}'$ denote a solution of* (5.19). *$\boldsymbol{\eta}'$ is also the global optimal solution of (**P3**).*

Proof. The detailed proof is provided in C.1. □

Then, in Lemma 5.1, we prove that for any given σ, if the solution obtained from (5.19) satisfies (5.13), it is the global optimal solution of (**P2**).

Algorithm 5.1 Proposed hybrid algorithm for (**P1**)

Input: Optimization problem (**P1**)

Output: $\boldsymbol{\eta}^*$

$\sigma \leftarrow S$.
Solve (5.19) to obtain $\boldsymbol{\eta}'$
if $\boldsymbol{\eta}'$ satisfies (5.13) **then**
    $\boldsymbol{\eta}^* \leftarrow \boldsymbol{\eta}', \varsigma^* \leftarrow S$, stop algorithm.
else
    $high \leftarrow S-1, low \leftarrow 2\sigma' \leftarrow (high+low)/2$
    repeat
        Solve (5.19) to obtain $\boldsymbol{\eta}'$
        if $\boldsymbol{\eta}'$ satisfies (5.13) **then**
            $low \leftarrow \sigma'$, $\sigma' \leftarrow (high+low)/2$
            $\varsigma^* = \sigma', \boldsymbol{\eta}^* \leftarrow \boldsymbol{\eta}'$
        else
            $high \leftarrow \sigma'$, $\sigma' \leftarrow (high+low)/2$
        end if
    until $\sigma' = high$
end if

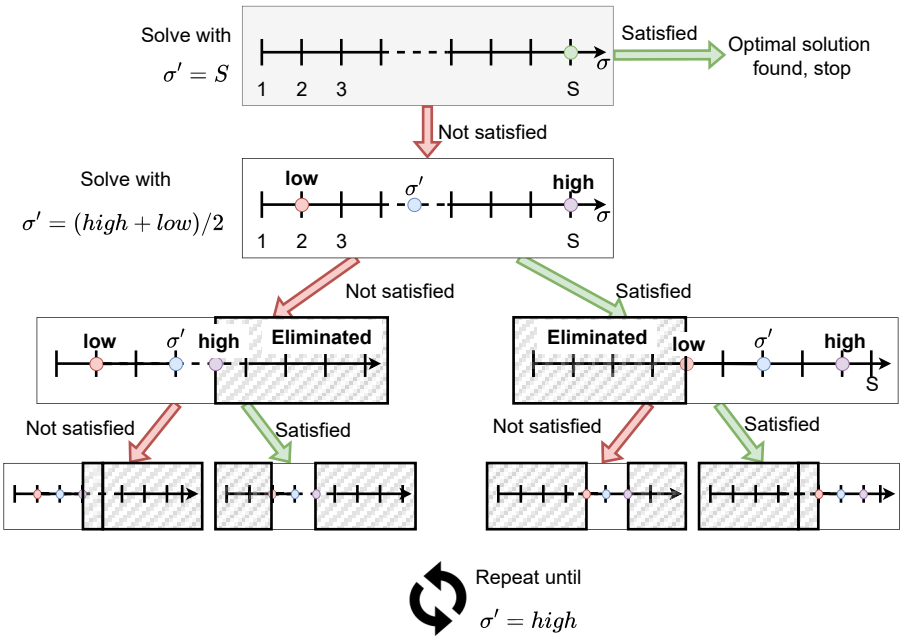


Figure 5.3 : An illustration of Algorithm 5.1.

Lemma 5.2. *If the solution* $\boldsymbol{\eta}'$ *obtained from* (5.19) *satisfies* (5.13), $\boldsymbol{\eta}'$ *is the global optimal solution of (**P2**).*

Proof. The detailed proof is provided in C.2. □

Next, in Theorem 5.1, we prove that for any given σ, if the solution $\boldsymbol{\eta}'$ obtained from solving (5.19) satisfies (5.13), then we can straightforwardly derive an equivalent feasible solution of (**P1**). Moreover, if the optimal solution of (**P3**) satisfies (5.13) in the case where $\sigma = S$, we can derive the global optimal solution of (**P1**). Note that when the optimal solution of (**P3**) cannot satisfy (5.13), it does not imply the absence of a feasible solution of (**P1**) for a given σ. Despite this limitation, the proposed Lagrangian method can still find solutions that are better than those from commercial solvers in a significantly shorter amount of time. Moreover, the proposed method can find the global optimal solution in most experiments as later shown in Section 5.4.

Theorem 5.1. *For any given σ, if the solution $\boldsymbol{\eta}'$ obtained from* (5.19) *satisfies* (5.13)*, then $\{\boldsymbol{\eta}', \mathbf{x}, \sigma\}$ is a feasible solution to (**P1**), where $\mathbf{x}$ can be straightforwardly derived from σ.*

Proof. The detailed proof is provided in C.3. □

5.3.2.3 Complexity analysis

The main component of Algorithm 5.1 is solving (5.19) to obtain $\boldsymbol{\eta}'$ in Steps 2 and 8. Using methods such as Gaussian elimination [129], each instance of (5.19) can be solved with time complexity $O\big((\sigma N + N)^3\big)$. Additionally, because we utilize binary search, (5.19) needs to be solved at most $\log(S)$ times, and thus the total time complexity of Algorithm 5.1 is $O\big(\log(S)(\sigma N + N)^3\big)$. In contrast, the time complexity of solving (**P1**) is exponential [129], and (**P1**) involves more variables. As a result, Algorithm 5.1 can be frequently deployed to reconfigure the shards, thereby reducing the risk of corruption from the adversary.

5.4 Performance Evaluation

5.4.1 Simulation Settings

To evaluate the performance of our proposed approach, we conduct various numerical experiments in five problem instances with different parameters (number of nodes, maximum difference, mean, and standard deviation (STD) of MUs score distribution) as shown in Table 5.1. Moreover, in all experiments, we set $T = 2000$ Tx/s and $\alpha_C = \alpha_D = \alpha_T = 1$. In these experiments, we compare the performance of three methods as follows:

- $SVP1$: We solve (**P1**) directly using the commercial solver CPLEX [127].
- $SVP2$: We apply an iterative algorithm similar to Algorithm 5.1. However,

Table 5.1 : Problem instance parameters.

Instance	**No. of nodes**	**Maximum difference**	**Mean**	**STD**
1	25	29	39.0	7.9
2	50	31	36.8	6.7
3	100	38	38.4	4.8
4	150	109	89.9	19.9
5	200	170	123.8	32.9

instead of solving (5.19) in Steps 2 and 8 as done in Algorithm 5.1, we solve (**P2**) using the commercial solver CPLEX [127].

- *LGRN*: We solve (**P3**) using the proposed Lagrangian approach as described in Algorithm 5.1.

In the first set of experiments, we examine the best solution found by the three methods under a limited running time (1 minute). Particularly, for each instance, we vary ς and τ to examine the best possible solution found by each method. The results show the lowest probability that the adversary can control more than 51% of a shard's score, denoted by $\text{Pr}_{51\%}$ ($\text{Pr}_{51\%}$ can be calculated using (5.4)). For each method, we record the lowest $\text{Pr}_{51\%}$ given a specific number of shards.

In the second set of experiments, we let all three methods run up to 10 minutes and then compare their running time and achievable throughput. For $SVP2$, we set the time limit of each iteration (Step 2 and Step 8 in Algorithm 5.1) to 1 minute. Moreover, we conduct experiments with different values of S to show the impact of S on the performance of the considered methods.

In the third set of experiments, we vary the values of p_n^A to study the impacts of the adversarial probability on the performance and security of the network. Particularly, we gradually increase p_n^A and examine the best achieved $\text{Pr}_{51\%}$ of the three

methods for various numbers of shards. Moreover, we also measure the highest throughput achieved by the considered methods.

Finally, we study the impact of the MUs' scores on the security of the system. In particular, for a network of 50 nodes, we randomly generate instances with different user engagement scores, as reflected by the different standard deviations and average values of engagement scores. Then, for each instance, we examine the best $\Pr_{51\%}$ achieved by the proposed $LGRN$ method to study how different distributions of scores can affect network security.

5.4.2 Simulation Results

Fig. 5.4 illustrates the best $\Pr_{51\%}$ obtained by the three methods for different numbers of shards in all problem instances. For example, in Instance-1 with 25 nodes, when we want to optimize the score allocation for 2 shards, the three methods achieve similar results, e.g., around 0.003 possibility to be attacked. However, if we want to have more shards in the system, $\Pr_{51\%}$ increases drastically if we use the $SVP1$ and $SVP2$ methods, e.g., 0.006 for 3 shards, 0.01 for 4 shards, and more than 0.01 for higher numbers of shards. In contrast, even for 20 shards, the value of $\Pr_{51\%}$ achieved by the $LGRN$ method is only around 0.003. Moreover, for all other instances, the $LGRN$ method can achieve $\Pr_{51\%}$ lower than the safety threshold (0.001) for up to 20 shards in the system. In contrast, $SVP1$ and $SVP2$ can only ensure security, i.e., $\Pr_{51\%} < 0.001$, for up to 4, 8, 10, and 11 shards in instances 2, 3, 4, and 5, respectively. Furthermore, compared to $SVP1$ and $SVP2$, $LGRN$ can achieve smaller $\Pr_{51\%}$ in all cases. Note that since the values of $\Pr_{51\%}$ achieved by $LGRN$ does not vary much compared to the other methods, it is not shown clearly in the figure. For example, in Instance-3, the values of $\Pr_{51\%}$ achieved by $LGRN$ ranges from $6\text{x}10^{-10}$ to $9\text{x}10^{-10}$, whereas those achieved by $SVP1$ ranges from $3\text{x}10^{-8}$ to 0.012.

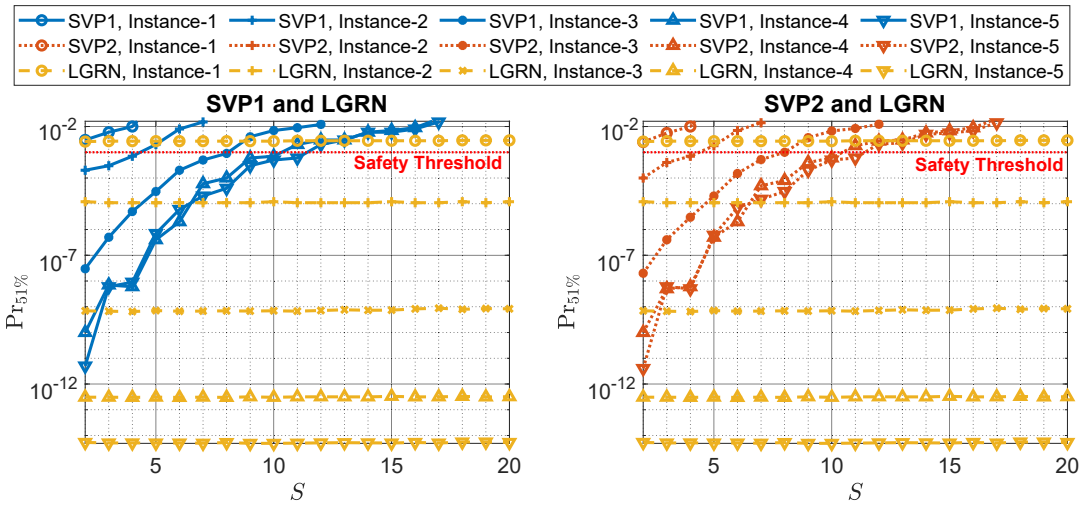


Figure 5.4 : $\Pr_{51\%}$ achieved by the three methods.

Fig. 5.5 shows the throughput achieved by the three methods for Instance-2 to Instance-5. We do not show the achieved throughput for Instance-1 because, in this instance, all three methods cannot ensure that $\Pr_{51\%}$ is lower than the safety threshold even for 2 shards, and thus the network cannot be divided into shards. For all the remaining instances, we can observe that the proposed $LGRN$ method performs better than the other methods, especially for high numbers of shards. For example, in Instance 2, $LGRN$ can achieve a throughput up to $20,000$ Tx/s, while the other methods can achieve at most $8,000$ Tx/s. Moreover, the $SVP1$ fails to find a feasible solution for $S > 5$, and thus the network cannot be divided into shards in these cases. Similarly, $LGRN$ performs better than the other methods by up to 25%, 50%, and 66.6%, in Instances 3, 4, and 5, respectively. Moreover, in Instance-2 to Instance-5, $LGRN$ can achieve global optimal solutions for all values of S, whereas $SVP1$ and $SVP2$ can not for higher values of S.

Fig. 5.6 shows the running time of the three methods in Instance-2 to Instance-5. As observed, the computational time of the proposed $LGRN$ is trivial compared to the other methods, particularly for high numbers of shards. For example, in

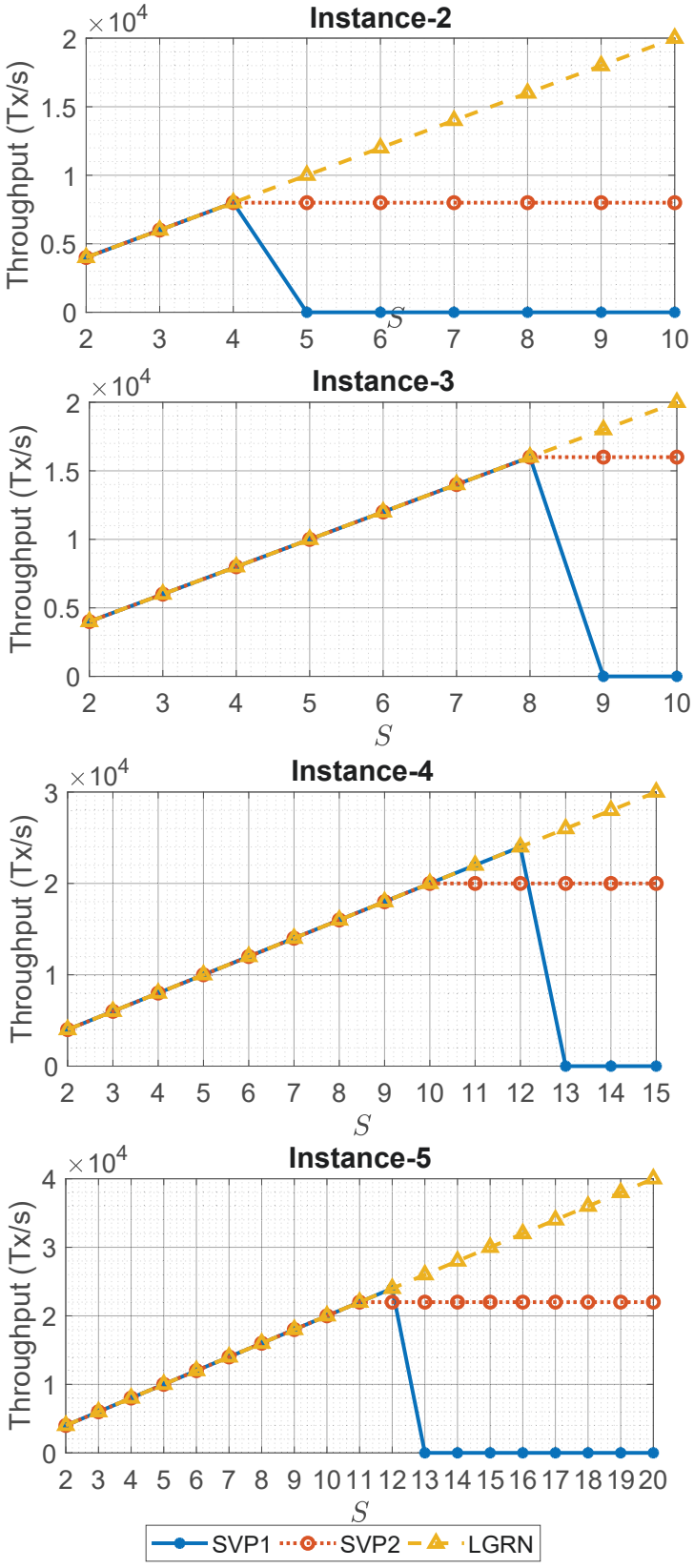


Figure 5.5 : Throughput achieved by the three methods.

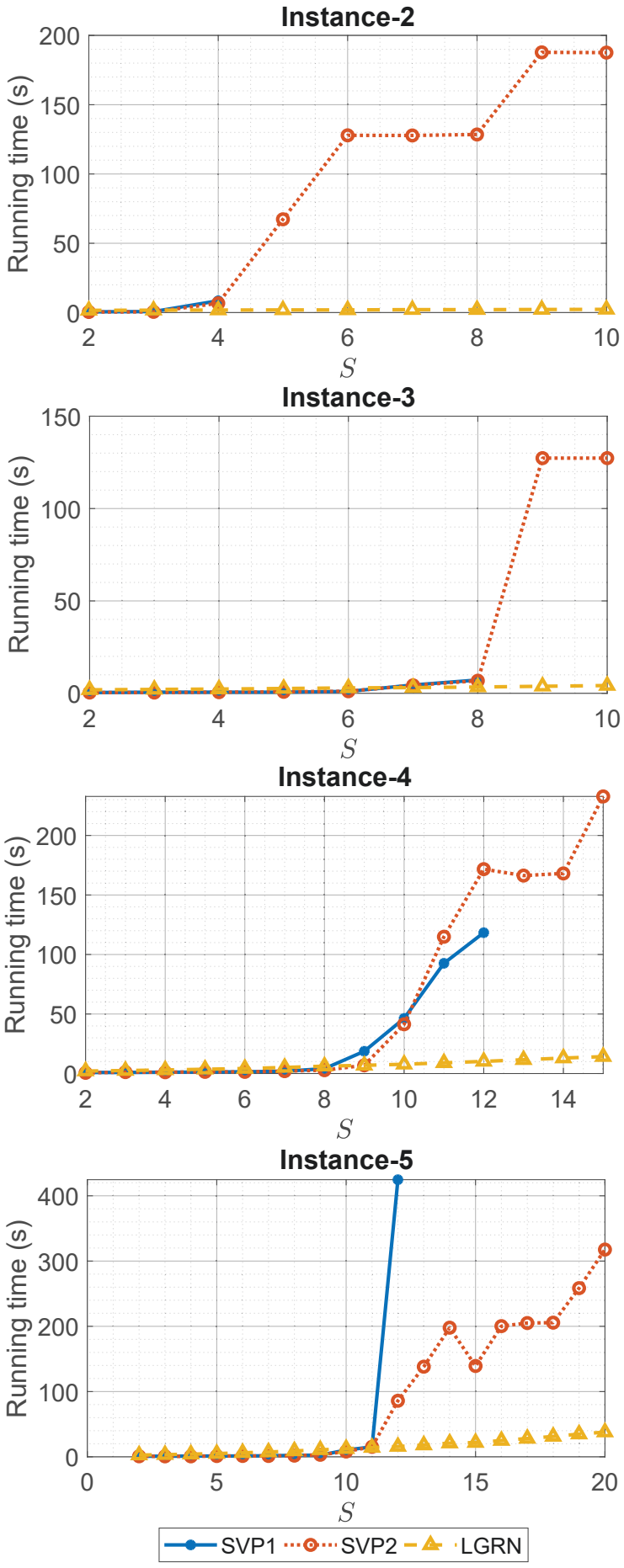


Figure 5.6 : Running time of the three methods.

Instance 2, $LGRN$ needs only 2.3 seconds to find the solution to divide the network into 10 shards. In contrast, $SVP2$ needs more than 187 seconds, whereas $SVP1$ can only find solutions for up to 4 shards. For more than 5 shards, $SVP1$ exceeds the running time limit without being able to find any feasible solution. Similarly, for the remaining instances, $LGRN$ can find better solutions in a much shorter time, i.e., more than 30, 16, and 8 times faster than $SVP2$. Meanwhile, $SVP1$ fails to find any feasible solution for 9, 13, and 12 shards in Instance-3, Instance-4, and Instance-5, respectively. Because of that, the graphs in Fig. 5.6 do not show the running time of $SVP1$ in the cases where $SVP1$ cannot find any feasible solution. Additionally, we can observe that the running time of $LGRN$ scales almost linearly with S, while the running time of the other methods increases exponentially as S increases.

Fig. 5.7 illustrates the results of the third set of experiments in terms of security, i.e., the change in $\Pr_{51\%}$ as the adversarial probability (p_n^A) increases. As observed from the figure, $LGRN$ can ensure the safety ($\Pr_{51\%} < 0.1\%$) of a network with 50 nodes and 10 shards even when p_n^A increases by nearly 150%. In contrast, if we use $SVP1$ and $SVP2$, $\Pr_{51\%}$ is nearly 0.04. Moreover, as p_n^A increases, $\Pr_{51\%}$ obtained from $SVP1$ and $SVP2$ increases drastically to over 0.1. This means that these methods cannot be employed for sharding when the adversary controls a high portion of MUs. Furthermore, we can also observe that $SVP2$ performs slightly better than $SVP1$ as the adversarial probability increases.

Fig. 5.8 shows the highest throughput achieved by the three methods as p_n^A increases. It can be observed that the highest throughput $LGRN$ can achieve is 20,000 Tx/s (global optimal) with up to 145% increase in adversarial probability. In contrast, $SVP1$ and $SVP2$ only attain a maximum of 8,000 Tx/s and their throughput decreases when p_n^A exceeds 115%. When p_n^A continues to rise, $SVP1$ and $SVP2$ fail to divide the network into shards at 120% and 140% respectively, while $LGRN$ can still sustain a throughput of 14,000 Tx/s at 155%. $LGRN$ only

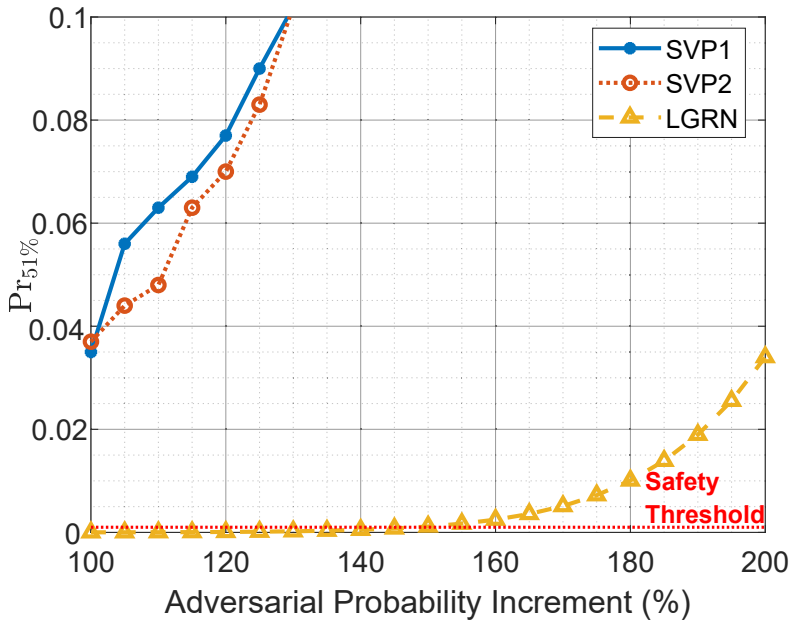


Figure 5.7 : $Pr_{51\%}$ under increasing adversarial probability.

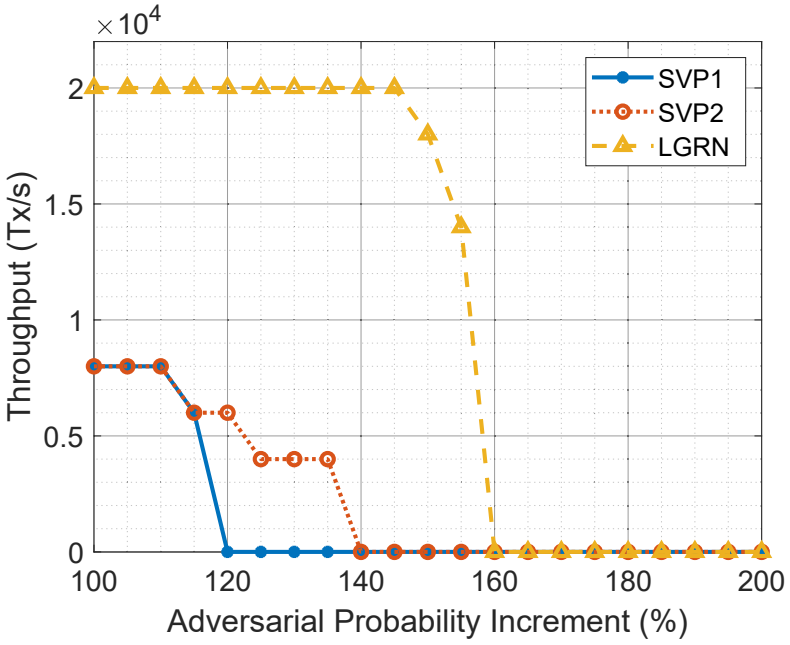


Figure 5.8 : Throughput under increasing adversarial probability.

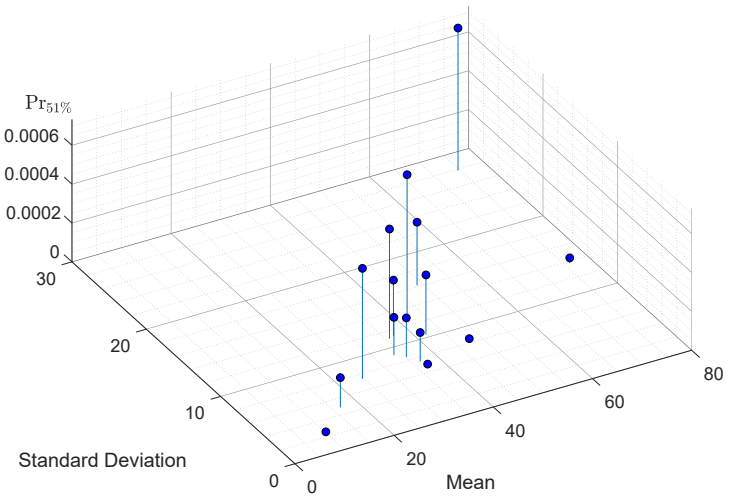


Figure 5.9 : Impacts of mean and standard deviation.

fails to find a feasible solution after p_n^A increases by more than 160%.

Fig. 5.9 illustrates the $Pr_{51\%}$ achieved by *LGRN* for different distributions of engagement scores. As observed from the figure, the more spread out the engagement scores are (i.e., the standard deviation is high), the higher the possibility of the network being attacked by the adversary. Moreover, the higher score the network has (i.e., higher mean), the more secure it becomes. For example, the instance with the largest standard deviation has the greatest likelihood of being attacked. Moreover, among instances with similar standard deviations, the ones with a higher mean (which corresponds to a higher total score) have a lower probability of being attacked. Therefore, while the network operators cannot influence the score distribution, they can try to attract more MUs to the network (thereby increasing the total score) to improve network security and performance.

5.5 Conclusion

In this chapter, we have developed a novel sharding blockchain framework for Metaverse applications. Particularly, we have developed a PoE consensus mecha-

nism that can encourage and reward MUs' resources contribution, thereby alleviating the huge resource demands for MSP and creating a more engaged MU community. Moreover, we have proposed a sharding management scheme and formulated an optimization problem to find the optimal number of shards and MUs allocation. Since the optimization problem is NP-complete, we have developed a hybrid approach that decomposes the problem (using the binary search method) into sub-problems that can be solved effectively by the Lagrangian method. As a result, the proposed approach can obtain solutions in polynomial time, thereby enabling flexible shard reconfiguration and reducing the risk of corruption from the adversary. Extensive numerical experiments have been conducted, and their results have shown that, compared to the state-of-the-art commercial solvers, our proposed approach can achieve up to 66.6% higher throughput in less than 1/30 running time. Moreover, the proposed approach can achieve global optimal solutions in most experiments. Furthermore, we have studied the impacts of key parameters on the performance of the system and shown that the proposed approach can further improve the robustness of the system.

Chapter 6

Conclusions and Potential Research Directions

6.1 Conclusion

In this thesis, we have presented our works in developing PoS-based frameworks for smart data management in mobile networks. In particular, our first work concerned with the mobile roaming data management problem. Specifically, to address the problem of roaming fraud for mobile service providers, we have proposed BlockRoam, a novel blockchain-based roaming management system which consists of our thoroughly analyzed PoS consensus mechanism and a smart-contract-enabled roaming management platform. Moreover, we have analyzed and showed that BlockRoam's security and performance can be enhanced by incentivizing more users to participate in the network. Therefore, we have developed an economic model based on Stackelberg game to jointly maximize the profits of network users, thereby incentivizing their participation. We have analyzed and determined the best strategies for the stakeholders and the stake pool. We have also proposed an effective solution that results in a unique equilibrium for our economic model. Lastly, we have evaluated the impacts of important parameters on the strategies and the equilibrium of the game. The proposed economic model can help the mobile service providers to earn additional profits, attract more investment to the blockchain network, and enhance the network's security and performance.

In the second work, we have introduced FedChain, an effective framework for federated-blockchain systems together with a cross-chain transfer protocol to facilitate the secure and decentralized transfer of tokens between the blockchains. In this

framework, we have proposed a novel consensus mechanism which can satisfy the CP, CG, and CQ properties, prevent various blockchain-specific attacks, and achieve better transaction confirmation time compared to existing consensus mechanisms. Robust theoretical analyses have been then conducted to prove FedChain's consensus mechanism security and performance properties. After that, a Stackelberg game model has been developed to examine the interactions between the stakeholders and the blockchains managed by chain operators. This model can provide additional profits for the stakeholders and enhance the security and performance of the blockchains. Through analyses of the Stackelberg game model, we can prove the uniqueness of the Stackelberg equilibrium and find the exact formula for this equilibrium. These results are especially important for the stakeholders to determine their best investment strategies and for the chain operators to design the optimal policy, i.e., block rewards. Finally, extensive experiments and simulations have been conducted to show that our proposed framework can help stakeholders to maximize their profits and the chain operator to design appropriate parameters to enhance FedChain's security and performance.

In the third work, we have developed MetaShard, a novel sharding blockchain framework for Metaverse applications. Particularly, we have developed a PoC consensus mechanism that can encourage and reward MUs' resources contribution, thereby alleviating the huge resource demands for MSP and creating a more engaged user community. Moreover, we have proposed a sharding management scheme and formulated the associated score allocation optimization problem. Since this problem is NP-complete, we have proposed a lightweight hybrid approach to efficiently solve the problem to obtain good solutions, thereby allowing frequent shards reconfiguration and improving network security. Extensive numerical experiments have shown that our proposed approach can achieve better results than other methods in terms of security, performance (up to 66.6% higher throughput), and running time (up to

30% faster). Moreover, we have studied the impacts of important parameters and shown that the proposed approach is more robust to stronger adversaries.

6.2 Future Works

Blockchain technology and PoS consensus mechanism have a great potential to address emerging problems in data management of future mobile networks, especially for privacy, security, and performance issues. However, as many new types of networks have emerged, e.g., Metaverse, decentralized web, etc., new challenges have arisen, requiring careful investigation and problem-specific modifications to be addressed.

- *PoS for Metaverse:* The recent emergence of Metaverse has brought many new challenges, especially in data management. Although existing works have been able to partially address these challenges, they have been mostly employing the PoW mechanism with significant disadvantages. Therefore, the PoS consensus mechanism can be a promising solution to address the challenges that Metaverse applications are facing. Although this thesis has proposed a PoS-based solution for Metaverse, the Metaverse applications are still in their very early stage of development. Therefore, new challenges might continue to emerge, and thus new solutions are always in need.

- *PoS for Web 3.0:* The next generation of World Wide Web, namely Web3, have recently attracted massive attention. With the advantages of decentralization, immutability, and privacy-preserving, blockchain technology has been identified as one of Web3's core enabling technology. In this context, PoS is expected to be the main consensus mechanism for blockchain in the Web3 era, as the PoW consensus mechanism cannot handle the expected massive traffic. However, the development of Web3 is still in a nascent stage, and research

on this topic is still limited. Therefore, developing PoS-based frameworks for Web3 is a future research direction with great potential.

- *Alternative consensus mechanisms:* Although the PoS mechanism has many advantages, it cannot address all the current challenges of blockchain technology such as scalability and privacy concerns. Instead, additional mechanisms such as sidechains and sharding are often integrated with PoS to address those issues, bringing extra complexity and security threats. Therefore, new or improved consensus mechanisms based on PoS that can inherently improve blockchain's scalability and privacy are necessary for the future development and applications of blockchain technology.

Appendix A

Proofs in Chapter 3

A.1 The proof of Theorem 3.1

First, we determine the probability $\Pr_{\text{CP}}$ that our proposed consensus mechanism will violate the common prefix property in the following Lemma.

Lemma A.1. *BlockRoam's consensus mechanism violates the common prefix property with probability* $Pr_{CP} = (1-\gamma)^{\kappa}$.

Proof. We will prove that our consensus mechanism can achieve a new bound of $\Pr_{\text{CP}}$ based on the following properties:

- A_1**:** An honest user will create exactly one block for each slot that the user is the leader.
- A_2**:** The list of leaders is known by every honest user at any time.
- A_3**:** An honest user, when received different forks, will adopt the longest valid fork, i.e., the longest fork that has no conflicting blocks and each block is signed by a designated leader.

Suppose properties A_1, A_2, and A_3 are satisfied, any fork created by the adversary must include all the blocks created by the honest users. This is because if an honest user does not change its block, then the adversary can either adopting the block in the fork or replace it by another block. However, as the list of leaders is known, the adversary must include the honest block in the fork. Otherwise, it will create an invalid fork that will be rejected based on property A_3. Moreover, any change in

a block's content results in a different block's hash, and the block's hash is linked to its previous block. Thus, the part of the chain from the first block to the latest honest block is confirmed by every honest user. As a result, the adversary can only create forks with κ last blocks different from the honest fork if it is elected leader for κ consecutive blocks. Since $(1-\gamma)$ is the ratio of adversarial stakes in the total network stakes, the probability that the adversary is elected leader for κ consecutive blocks is

$$\Pr_{\mathrm{CP}} = (1-\gamma)^{\kappa} = (\frac{B_{\mathcal{A}}}{B_{\mathcal{A}} + \sum_{i=1}^{N} B_i})^{\kappa}, \tag{A.1}$$

which is also the probability that the common prefix property is violated.

Properties A_1 and A_3 can be easily satisfied if all the honest users follow the consensus mechanism. Property A_2 can be ensured by conducting the coin-tossing protocol at the beginning of each epoch, instructing the honest users to broadcast the leader list of the next epoch during the current epoch, and requiring an honest user to be online at least once each epoch (this requirement is reasonable since an epoch of [23] lasts for 5 days). □

Next, we prove that our proposed consensus mechanism will always satisfy the chain growth property in the following Lemma.

Lemma A.2. *BlockRoam's consensus mechanism will always satisfy the chain growth property , i.e.,* $\Pr_{\mathrm{CG}} = 0$.

Proof. Even if a new block is not broadcast during a time slot, an empty block will be added to the chain. Therefore, a chain received at time $t + \varsigma$ will always be ς blocks longer than the chain at time t. □

Then, we determine the probability $\Pr_{\mathrm{CQ}}$ that our proposed consensus mechanism will violate the chain quality property in the following Lemma.

Lemma A.3. *BlockRoam's consensus mechanism violates the chain quality property with probability* $\Pr_{\text{CQ}} = 1 - \mathrm{e}^{\frac{(\gamma-1)\mathrm{l}\delta^2}{2}}$.

Proof. We can characterize the block adding process among the honest stakeholders and the adversary as a binomial random walk [126]. During the considered l slots, the leader election processes can be considered independent Bernoulli trials $X_1 ... X_l$ such that, for $1 \leq i \leq l$, $\Pr[X_i = 0] = \gamma$ and $\Pr[X_i = 1] = 1 - \gamma$. Then, the expected value of the trials is $\mathbb{E}[X] = \sum_{i=1}^{l}(1-\gamma)$. Applying the Chernoff bound [126], the probability that the adversary creates less than $l(1-\mu)$ blocks, i.e., $X = \sum_{i=1}^{l} X_i < l(1-\mu)$, is

$$\begin{aligned} \Pr[X < (1-\delta)\mathbb{E}[X]] &= \exp\left(\frac{-\mathbb{E}[X]\delta^2}{2}\right), \\ \Pr[X < (1-\delta)l(1-\mu)] &= \exp\left(\frac{(\mu-1)l\delta^2}{2}\right), \end{aligned} \tag{A.2}$$

where δ is any real number such that $0 < \delta \leq 1$. In the case of *ideal chain quality* [38], we have $\mu = \gamma$, and the chain quality violation probability $\Pr_{\text{CQ}}$ is

$$\Pr_{\text{CQ}} = 1 - \exp\left(\frac{l(\gamma-1)\delta^2}{2}\right) \tag{A.3}$$

□

From Lemma 1, we can observed that $\Pr_{\text{CP}}$ decreases exponentially as κ increases. Similarly, $\Pr_{\text{CQ}}$ decreases exponentially as δ and l grow as proven in Lemma 3. Thus, choosing appropriate parameter can help to significantly reduce these violation probabilities. Moreover, $\Pr_{\text{CG}}$ is always zero as proven in Lemma 2. Thus, the proof is completed.

A.2 The proof of Theorem 3.2

In a double-spending attack, the adversary attempts to revert a transaction by adding a conflicting transaction to the blockchain after the original transaction is

confirmed. It is straightforward to see that this attack cannot be successful if the common prefix property is not violated.

In grinding attacks and nothing-at-stake attacks, the adversary creates multiple blocks to influence the seeds of the leader selection process or revert some blocks in the chain. More specifically, grinding attacks target the blockchain where the seeds for leader selection are the previous blocks' headers. However, the seeds for leader selection are created by the committee in BlockRoam, and thus grinding attacks are mitigated. Moreover, although the adversary can create forks, nothing-at-stakes attacks do not affect the network's security as long as the common prefix property is not violated. Furthermore, the adversary's deposit will be confiscated if the adversary signs different blocks for the same time slot.

In bribe attacks, the adversary can bribe the leaders to create specific blocks, e.g., to support other types of attacks such as double-spending or transaction denial. In the context of roaming, bribe attacks may cause severe financial loss. For example, an adversary can perform a bribe attack to support a transaction denial attack, i.e., bribe the leaders to not include any transaction made from a certain MSP, and consequently that MSP cannot process any roaming request. In this case, the deposits will be confiscated, which significantly increases the costs of bribe attacks and transaction denial attacks.

In a long-range attack, a committee member immediately sells its stakes at the beginning of its designated epoch, and thus it can behave maliciously for the rest of the epoch without consequences. Our system can mitigate this attack by locking committee members' stakes during their designated epoch.

A.3 The proof of Theorem 3.3

In Case 1 and 2, although the follower can invest using any number of stakes within its budget, we prove in Lemma 1 that a rational follower will always invest all its budget.

Lemma A.4. *Let s_i' denote a strategy where follower i invests less than its total budget, i.e., $m_i' + p_i' < B_i$, with corresponding utility U_i', and s_i is a strategy where follower i invests all its budget, i.e., $m_i + p_i = B_i$, with corresponding utility U_i. For Case 1 and Case 2, we always have $U_i' < U_i$.*

Proof. We consider Cases 1 and Case 2 separately as follows:

- *Case 1:* When the follower only invests $p_i' < B_i$ stakes to the pool, its expected payoff $U_i'^1$ is equal to r_i^p in (3.4). Now, if the follower invests $p_i = B_i$ to the pool, its payoff can be determined as follows:

$$U_i^1 = \frac{B_i(1-\alpha)}{\sigma + \sum_{n\in\mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j} R - c\mathrm{e}^{-B_i}. \tag{A.4}$$

  Then, the difference in payoff between the two strategies is

$$U_i^1 - U_i'^1 = \frac{(B_i - p_i')(1-\alpha)}{\sigma + \sum_{n\in\mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j} R + (c\mathrm{e}^{-p_i'} - c\mathrm{e}^{-B_i}), \tag{A.5}$$

  which is always positive since $p_i' < B_i$.

- *Case 2:* When follower i only uses $m_i' < B_i$ stakes for self-mining, its payoff $U_i'^2$ is equal to r_i^m in (3.5). If the follower self-mines with all its budget, the payoff is

$$U_i^2 = \frac{B_i}{\sigma + \sum_{n\in\mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j} R - C_i. \tag{A.6}$$

  The different in payoff is then determined by:

$$U_i^2 - U_i'^2 = \frac{B_i - m_i'}{\sigma + \sum_{n\in\mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j} R, \tag{A.7}$$

  which is always positive since $m_i' < B_i$.

□

Moreover, we prove in the following Lemma that, given the same stakes to invest, Case 3 always gives a worse payoff than Case 2, and thus a rational follower will never choose Case 3.

Lemma A.5. *Let U_i^2, U_i^3 denote the payoff of Case 2 and Case 3, respectively. If follower i invests the same β stakes in these two cases, i.e., $m_i^2 = \beta$ and $m_i^3 + p_i^3 = \beta$, then Case 2 always gives a better payoff than Case 3, i.e., $U_i^2 > U_i^3, \forall \alpha, c$.*

Proof. The difference in payoff between Case 2 and 3 can be calculated by

$$\begin{aligned} U_i^2 - U_i^3 =& \frac{\beta}{\sigma + \sum_{n\in\mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j} R - C_i \\ & - \left(\frac{\beta - p_i^3 + p_i^3(1-\alpha)}{\sigma + \sum_{n\in\mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j} R - C_i - c\mathrm{e}^{-p_i^3} \right), \\ =& \frac{p_i^3 \alpha}{\sigma + \sum_{n\in\mathcal{N}_p} p_n + \sum_{j=1}^{N} m_j} R + c\mathrm{e}^{-p_i^3}, \end{aligned} \tag{A.8}$$

which is always positive. □

As a result, Case 3 can be removed from the strategy space of every follower.

In Case 4, the follower receives payoff $U_i^4 = 0$. Therefore, if follower i has budget B_i such that $r_i^p > 0$ or $r_i^m > 0$, the follower will invest stakes to the pool or to self-mining, i.e., switch to Case 1 and 2. If follower i has B_i such that $r_i^p < 0$ and $r_i^m < 0$, the follower will not participate in the consensus process, and thus it does not have any impact on the game.

Since Case 3 and Case 4 are eliminated, it follows from Lemma 4 that the best response of a stakeholder is to use its budget either to invest to the pool or for self-mining, and the proof is completed.

A.4 The proof of Theorem 3.5

We prove that there exists at least one solution of (3.12). This means that there exists at least one leader's optimal strategy. Since only the decision variable x_i is a binary number in (3.12), if we fix the value of $x_i, \forall i \in \mathcal{N}$, (3.12) becomes a Linear Programming (LP) problem. By fixing the value of x_i, we can decompose (3.12) into 2^N LP problems (there are 2^N different combinations of x_i's values). Each LP problem has the form of (3.12), except that all x_i are constants instead of decision variables. In the LP problem where $\sum_i^N x_i = 0$, the optimal objective value is 0. In each of the remaining LP problems, the feasible region is constrained by

$$\frac{B_i R\alpha}{\sum_{j=1}^{N} B_j} + c\mathrm{e}^{-B_i} = C_i, \forall i \in \mathcal{N}_p. \tag{A.9}$$

Since c

each of

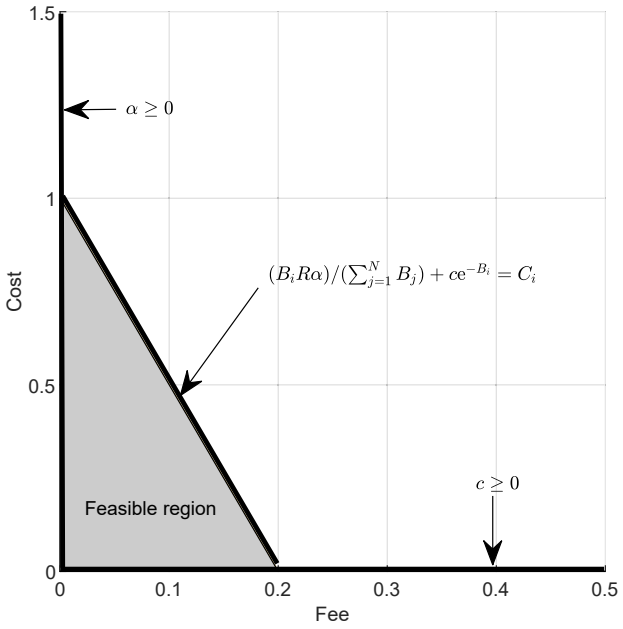


Figure A.1 : An illustration of a bounded LP's feasible region.

Since these 2^N LP problems enumerate all possible combinations of x_i and each of

these LP has at least one optimal solution, there exists at least one optimal solution of the MILP. Moreover, the existence of the best response of every follower is proven in Theorem 4. Therefore, there exists at least one Stackelberg equilibrium (U_p^*, U_i^*) with the corresponding Stackelberg solution (s_p^*, s_i^*) in this game.

A.5 The proof of Theorem 3.6

First, we will prove that two different optimal strategies of the leader, denoted by $s_p^*(\alpha^*, c^*)$ and $s_p^{'*}(\alpha^{'*}, c^{'*})$, must have two different fees, i.e., $\alpha^* \neq \alpha^{'*}$. For the sake of contradiction, we will prove that if $\alpha^* = \alpha^{'*}$, then $s_p^*(\alpha^*, c^*)$ and $s_p^{'*}(\alpha^{'*}, c^{'*})$ must be the same. Specifically, assume that $\alpha^* = \alpha^{'*}$, and as s_p^* and $s_p^{'*}$ are optimal strategies of the leader, then we have $U_p^* = U_p^{'*}$, i.e.,

$$\begin{aligned}
U_p^* &= U_p^{'*}, \\
\sum_{i\in\mathcal{N}_p}\left(\frac{p_i\alpha^*}{\sigma+\sum_{j=1}^N B_j}R + c^*\mathrm{e}^{-B_i}\right) &= \sum_{i\in\mathcal{N}_p}\left(\frac{p_i\alpha^{'*}}{\sigma+\sum_{j=1}^N B_j}R + c^{'*}\mathrm{e}^{-B_i}\right), \\
\sum_{i\in\mathcal{N}_p}\left(\frac{p_i(\alpha^*-\alpha^{'*})}{\sigma+\sum_{j=1}^N B_j}R + (c^*-c^{'*})\mathrm{e}^{-B_i}\right) &= 0, \\
\sum_{i\in\mathcal{N}_p}\left((c^*-c^{'*})\mathrm{e}^{-B_i}\right) &= 0, \\
c^* &= c^{'*},
\end{aligned} \tag{A.10}$$

which means that $s_p^* = s_p^{'*}$, i.e., the two strategies are the same.

Then, with the secondary priority to minimize α, we can define the unique optimal strategy as follows:

$$s_p^* = \{(\alpha^*, c^*) | \alpha^* = \min \alpha \in \mathbf{S}_p^*\}, \tag{A.11}$$

where $\mathbf{S}_p^*$ is the set of all strategies that maximize U_p. Since the leader's and followers' optimal strategies can be uniquely defined, the Stackelberg equilibrium is unique.

Appendix B

Proofs in Chapter 4

B.1 The proof of Theorem 4.1

We first prove that our FedChain's consensus mechanism can satisfy the CP property in Lemma 1.

Lemma B.1. *The probability that FedChain's consensus mechanism violates the common prefix property with parameter $\kappa \in \mathbb{N}$ is less than or equal to $(1-\gamma)^{\kappa}$.*

Proof. In order to violate the CP property, the adversary must have two forks with at least κ conflicting blocks, and both forks must be accepted by the honest stakeholders. However, an honest stakeholder will accept only one fork in the same time slot. Therefore, the adversary must (i) create a fork, (ii) have the honest stakeholders accept it, (iii) create another fork with a conflicting block at a later time slot, and (iv) have the honest stakeholders accept the new fork. We will prove that the adversary can only do that if it is elected to be the leader for κ consecutive blocks.

Without loss of generality, assume that the adversary is elected to be the leader at time slot $\mathrm{s}_1^{\mathrm{l}}$, $\mathrm{s}_2^{\mathrm{l}}$, and $\mathrm{s}_4^{\mathrm{l}}$. This means that at $\mathrm{s}_3^{\mathrm{l}}$ an honest stakeholder is elected to be the leader. Assume that the adversary wants to create two conflicting forks $\mathcal{C}_1, \mathcal{C}_2$. Let $\mathcal{B}_j^i$ denote the block from fork $\mathcal{C}_i$ at time slot $\mathrm{s}_{\mathrm{j}}^{\mathrm{l}}$. Firstly, at $\mathrm{s}_1^{\mathrm{l}}$ and $\mathrm{s}_2^{\mathrm{l}}$, the adversary broadcasts blocks $\mathcal{B}_1^1$ and $\mathcal{B}_2^1$. At this point, the adversary can create fork $\mathcal{C}_2$ with different blocks, i.e., $\mathcal{B}_1^2 \neq \mathcal{B}_1^1$ and $\mathcal{B}_2^2 \neq \mathcal{B}_2^1$, and has both forks accepted by the honest stakeholder (some honest stakeholders will adopt $\mathcal{C}_1$ while some will adopt $\mathcal{C}_2$).

However, at s_3^1, the honest leader will either choose one of the two forks to adopt. Assume that the leader chooses $\mathcal{C}_1$, it will add block $\mathcal{B}_3^1$ to the chain, and the fork $\mathcal{C}_2$ will be discarded by all honest stakeholders. Next, at s_4^1, the adversary is elected to be the leader and can create a fork again. At this point, the adversary can try to broadcast $\mathcal{C}_2$ to the honest stakeholder again with block $\mathcal{B}_1^2 \neq \mathcal{B}_1^1$ (e.g., to gain back the tokens spent in block $\mathcal{B}_1^1$). Nevertheless, any change in a block's content results in a different block's hash, and the block's hash is linked to its previous block. Thus, $\mathcal{B}_1^1$ cannot be changed unless block $\mathcal{B}_3^1$ is changed. However, since the leader of $\mathcal{B}_3^1$ is honest, the block will not be changed (due to consensus rules I_2 and I_3).

Moreover, the leader election is conducted at the beginning of each epoch, and as long as the PVSS protocol is secure (proven in [91]), any honest stakeholder can obtain and verify the correct leader list if that stakeholder is online at least once during the epoch (thanks to consensus rule I_1). Since the epoch is long (e.g., 5 days [23]), we can assume that every honest stakeholder will have the correct leader list. Therefore, the adversary also cannot broadcast $\mathcal{B}_3^1 \neq \mathcal{B}_3^2$ on its own since it is not the designated leader. Thus, the adversary must include $\mathcal{B}_3^1$ and every block before that. Otherwise, it will create an invalid fork that will be rejected due to due to consensus rule I_4.

As a result, the part of the chain from the first block to the latest honest block (e.g., until $\mathcal{B}_3^1$ in the above analysis) is confirmed by every honest user. Therefore, the adversary can only create forks with κ last blocks different from the honest fork if it is elected to be the leader for κ consecutive blocks. Since $(1-\gamma)$ is the ratio of adversarial stakes in the total network stakes, the probability that the adversary is elected to be the leader for κ consecutive blocks is

$$\Pr_{\text{CP}} = (1-\gamma)^{\kappa}, \tag{B.1}$$

which is also the probability that the CP property is violated. □

Then, we prove that our FedChain's consensus mechanism satisfies the CG property in Lemma 2.

Lemma B.2. *FedChain's consensus mechanism satisfies the chain growth property*

Proof. Even if a new block is not broadcast during a time slot, an empty block will be added to the chain. Therefore, the CG property will always be satisfied. □

Next, we prove that FedChain's consensus mechanism can satisfy the CQ property in Lemma 3.

Lemma B.3. *The probability that FedChain's consensus mechanism violates the ideal chain quality property with parameters l, μ over l blocks is no more than* $1 - \exp\left(\frac{l(\gamma-1)\delta^2}{2}\right)$.

Proof. We can characterize the block adding process among the honest stakeholders and the adversary as a binomial random walk [126]. During the considered l slots, the leader election processes can be considered independent Bernoulli trials $X_1, \ldots, X_l$ such that, for $1 \leq i \leq l$, $\Pr[X_i = 0] = \gamma$ and $\Pr[X_i = 1] = 1-\gamma$. Then, the expected value of the trials is $\mathbb{E}[X] = \sum_{i=1}^{l}(1-\gamma)$. Applying the Chernoff bound [126], the probability that the adversary creates less than $l(1-\mu)$ blocks, i.e., $X = \sum_{i=1}^{l} X_i < l(1-\mu)$, is

$$\begin{aligned} \Pr[X < (1-\delta)\mathbb{E}[X]] &= \exp\left(\frac{-\mathbb{E}[X]\delta^2}{2}\right), \\ \Pr[X < (1-\delta)l(1-\mu)] &= \exp\left(\frac{(\mu-1)l\delta^2}{2}\right), \end{aligned} \tag{B.2}$$

where δ is any real number such that $0 < \delta \leq 1$. In the case of *ideal CQ* [38], we have $\mu = \gamma$, and the ideal CQ violation probability is

$$\Pr_{\text{CQ}} = 1 - \exp\left(\frac{l(\gamma-1)\delta^2}{2}\right) \tag{B.3}$$

□

From Lemma 1, we have the CP violation probability $\Pr_{\text{CP}} = (1-\gamma)^{\kappa}$ which decreases exponentially as κ grows. Then, we proved in Lemma 2 that the CG property will always be satisfied. Finally, from Lemma 3, we have the CQ violation probability $\Pr_{\text{CQ}} = 1 - \exp\left(\frac{l(\gamma-1)\delta^2}{2}\right)$ which decreases exponentially as l and δ grow. Thus, all the three violation probabilities can be satisfied with overwhelming probabilities, and the proof is completed.

B.2 The proof of Theorem 4.2

We prove FedChain's consensus mechanism's ability to prevent each attack as follows:

- **Double-spending attack** To double-spend, the attacker has to either create a conflicting transaction in the same fork, i.e., create Tx1 in $\mathcal{B}_i^1$ and Tx2 in $\mathcal{B}_j^1$, or create two transactions in two forks, i.e., create Tx1 in $\mathcal{B}_i^1 \in \mathcal{C}_1$ and Tx2 in $\mathcal{B}_j^2 \in \mathcal{C}_2$. For the first approach, Tx2 is not a valid transaction and will be rejected. For the second approach, if the CP property is not violated, only one of $\mathcal{C}_1$ and $\mathcal{C}_2$ will be confirmed. Thus, this attack is prevented.

- **Grinding attack:** The seeds for leader and committee selection are created by the committee via the PVSS protocol in FedChain's consensus mechanism. Therefore, grinding attacks are prevented.

- **Nothing-at-stake attacks:** Although the adversary can create valid forks, they are not confirmed. Therefore, the vendors only have to wait until a transaction is confirmed. Thus, this attack is prevented as long as the CP property holds.

- **Bribe attacks:** In the considered adaptive adversary model, the adversary can only corrupt honest stakeholders with a delay. Since the adversary can-

not know who is the leader in advance, the adversary cannot bribe the leaders. Moreover, the leaders who are bribed will be penalized by the penalty mechanism.

- **Transaction denial attack:** With the liveness property, a transaction will eventually be included in a block created by an honest leader. As a result, this attack can be prevented as long as the liveness property (or the CG and CQ properties) holds.

- **Long-range attack:** FedChain's consensus mechanism can prevent this attack by locking committee members' stakes during their designated epochs.

B.3 The proof of Theorem 4.3

Let $\mathcal{S}_n$ denote the strategy space of follower n. Then, any strategy $\mathbf{s}_n = [s_n^1, \ldots, s_n^M]$ that satisfies

$$\sum_{m=1}^{M} s_n^m \leq B_n, \tag{B.4}$$

is a feasible strategy of follower n, i.e., $\mathbf{s}_n \in \mathcal{S}_n$. We first prove $\mathcal{S}_n$ to be compact and convex $\forall n \in \mathcal{N}$ in Lemma 4.

Lemma B.4. *$\mathcal{S}_n$ is compact and convex $\forall n \in \mathcal{N}$.*

Proof. Let $\mathbf{s}_n$ and $\mathbf{s'}_n$ be any two different strategies in $\mathcal{S}_n$. To prove $\mathcal{S}_n$ is convex, we prove that any convex combination of $\mathbf{s}_n$ and $\mathbf{s'}_n$ is in $\mathcal{S}_n$, i.e.,

$$\begin{aligned} \lambda\mathbf{s}_n + (1-\lambda)\mathbf{s'}_n = [\lambda s_n^1 + (1-\lambda)s_n^{'1}, \ldots, \lambda s_n^M + (1-\lambda)s_n^{'M}] \in \mathcal{S}_n, \\ \forall \lambda \in (0,1), \forall \mathbf{s}_n, \mathbf{s'}_n \in \mathcal{S}_n. \end{aligned} \tag{B.5}$$

Since $\lambda s_n^m + (1-\lambda)s_n^{'m} \leq max\{s_n^m, s_n^{'m}\}, \forall m \in \mathcal{M}$, we have

$$\sum_{m=1}^{M} \left(\lambda s_n^1 + (1-\lambda)s_n^{'1}\right) \leq max\left\{\sum_{m=1}^{M} s_n^m, \sum_{m=1}^{M} s_n^{'m}\right\} \leq B_n. \tag{B.6}$$

From (B.6), all convex combinations of $\mathbf{s}_n$ and $\mathbf{s'}_n$ satisfy (B.4), and thus they all lie in $\mathcal{S}_n$. As a result, $\mathcal{S}_n$ is convex. Moreover, since $\mathcal{S}_n$ is closed and bounded, it is compact. □

Then, we prove that U_n is concave in Lemma 5.

Lemma B.5. *U_n is concave over S_n, $\forall n \in \mathcal{N}$.*

Proof. We have:

$$\frac{\partial^2 U_n^m}{\partial (s_n^m)^2} = \frac{-2R_m T_m}{(s_n^m + T_m)^3} \leq 0. \tag{B.7}$$

Thus, U_n^m is concave over $\mathcal{S}_n$. Then, $U_n = \sum_{m=1}^{M} U_n^m$ is also concave over $\mathcal{S}_n$. □

According to [112], if S_n is compact and convex and U_n is quasi-concave $\forall n \in \mathcal{N}$, there exists at least one Nash equilibrium. It follows from Lemma 4 and 5 that the follower sub-game satisfies these conditions, and thus the proof of this Theorem is complete.

B.4 The proof of Theorem 4.4

According to Rosen's theorem [112], a sufficient condition to guarantee the uniqueness of the equilibrium and the convergence to the equilibrium is that the matrix $[\mathbf{G}(\mathbf{s},\omega) + \mathbf{G}^T(\mathbf{s},\omega)]$ is negative definite for a fixed $\omega > 0$. $\mathbf{G}(\mathbf{s},\omega)$ can be calculated by:

$$\mathbf{G}(\mathbf{s},\omega) = \begin{bmatrix} \omega_1 \frac{\partial^2 U_1}{\partial s_1^1 \partial s_1^1} & \omega_1 \frac{\partial^2 U_1}{\partial s_1^2 \partial s_1^1} & \cdots & \omega_1 \frac{\partial^2 U_1}{\partial s_1^M \partial s_N^1} \\ \omega_1 \frac{\partial^2 U_1}{\partial s_1^1 \partial s_1^2} & \omega_1 \frac{\partial^2 U_1}{\partial s_1^2 \partial s_1^2} & \cdots & \omega_1 \frac{\partial^2 U_1}{\partial s_1^M \partial s_N^2} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_N \frac{\partial^2 U_N}{\partial s_N^1 \partial s_1^M} & \omega_N \frac{\partial^2 U_N}{\partial s_N^2 \partial s_1^M} & \cdots & \omega_N \frac{\partial^2 U_N}{\partial s_N^M \partial s_N^M} \end{bmatrix} \tag{B.8}$$

Let $\omega_n = 1, \forall n \in \mathcal{N}$, the entries of $\mathbf{G}(\mathbf{s}, \omega)$ can then be calculated as follows:

$$\Phi_n^m = \frac{2R_m}{\sum_{n=1}^{N} B_n}\left(\frac{-T_m^2}{(s_n^m + T_m)^3}\right), \tag{B.9}$$

and

$$\phi_n^m = \frac{2R_m}{\sum_{n=1}^{N} B_n}\left(\frac{s_n^m T_m}{(s_n^m + T_m)^3}\right). \tag{B.10}$$

From (B.9) and (B.10), we can calculate

$$\Delta_n^m = \Phi_n^m - \phi_n^m = \frac{2R_m}{\sum_{n=1}^{N} B_n}\left(\frac{-T_m(s_n^m + T_m)}{(s_n^m + T_m)^3}\right), \tag{B.11}$$

which is negative. Then, $\mathbf{G}(\mathbf{s}, \omega)$ can be expressed as a sum of 2 matrices $\mathbf{G} = \mathbf{D}+\mathbf{E}$, where:

- $\mathbf{D}$ is similar to $\mathbf{G}$, except that all the diagonal entries of $\mathbf{D}$ are ϕ_n^m instead of Φ_n^m. Then, $\mathbf{D}$ has identical columns (columns i and $M + i$ are identical), and thus it is negative semi-definite.
- $\mathbf{E}$ is a diagonal matrix with entries equal to Δ_n^m. Thus, $\mathbf{E}$ is negative definite

As a result, $\mathbf{G}(\mathbf{s}, \omega)$ is the sum of a negative semi-definite matrix and a negative definite matrix ($\mathbf{E}$). Thus, $\mathbf{G}(\mathbf{s}, \omega)$ is negative definite. Therefore, $[\mathbf{G}(\mathbf{s}, \omega) + \mathbf{G}^T(\mathbf{s}, \omega)]$ is negative definite, and the proof is completed.

B.5 The proof of Theorem 4.5

Assume that follower n is employing strategy $\mathbf{s}_n$ which invests less than the available budget, i.e., $\sum_{m=1}^{M} s_n^m < B_n$. The utility function in this case is given in (4.3). Without loss of generality, if the follower chooses a strategy $\mathbf{s}'_n$ which invests the remaining budget Δs_n^j into a chain j, its utility function becomes:

$$U'_n = \sum_{m \in \mathcal{M}_{-j}} \left(\frac{s_n^m}{s_n^m + T_m} R_m\right) + \frac{s_n^j + \Delta s_n^j}{s_n^j + \Delta s_n^j + T_j} R_j, \tag{B.12}$$

where $\mathcal{M}_{-j}$ is the set of all chains except chain j. Then, the difference in the utilities between the two strategies is:

$$U'_n - U_n = \frac{s_n^j + \Delta s_n^j}{s_n^j + \Delta s_n^j + T_j} R_j - \frac{s_n^j}{s_n^j + T_j} R_j, = \frac{\Delta s_n^j \sum_{k \in \mathcal{N}_{-n}} s_k^j}{(s_n^j + \Delta s_n^j + T_j)(s_n^j + T_j)}, \quad \text{(B.13)}$$

which is always positive. This means that $\mathbf{s}_n$ always gives a lower payoff than $\mathbf{s}'_n$ regardless of the other followers' strategies, and the proof is completed.

B.6 The proof of Theorem 4.6

To prove this Theorem, we prove that at the point where every follower's strategy satisfies $s_n^m = B_n \frac{R_m}{\sum_{i=1}^M R_i}, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}$, every follower's strategy maximizes its utility ($\frac{\partial U_n}{\partial s_n^m} = 0$). Therefore, no rational follower will deviate from this point, and thus this is the Nash equilibrium of this game [40]. Substitute $s_n^m = B_n \frac{R_m}{\sum_{i=1}^M R_i}$ into $\frac{\partial U_n}{\partial s_n^m}$, we have

$$\begin{aligned}
\frac{\partial U_n}{\partial s_n^m} =& \frac{R_m T_m}{(s_n^m + T_m)^2} - \frac{R_M T_M}{(s_n^M + T_M)^2}, \\
=& \frac{\sum_{j \in \mathcal{N}_{-n}} B_j \frac{R_m^2}{\sum_{i=1}^M R_i}}{\sum_{n=1}^N (B_n \frac{R_m}{\sum_{i=1}^M R_i})^2} - \frac{\sum_{j \in \mathcal{N}_{-n}} B_j \frac{R_M^2}{\sum_{i=1}^M R_i}}{\sum_{n=1}^N (B_n \frac{R_M}{\sum_{i=1}^M R_i})^2}, \\
=& \sum_{j \in \mathcal{N}_{-n}} B_j \left(\frac{\sum_{i=1}^M R_i}{\sum_{n=1}^N (B_n)^2} - \frac{\sum_{i=1}^M R_i}{\sum_{n=1}^N (B_n)^2} \right), \\
=& 0, \forall m \in \mathcal{M}, \text{and } \forall n \in \mathcal{N}.
\end{aligned} \quad \text{(B.14)}$$

The proof is now completed.

B.7 The proof of Theorem 4.7

To find the equilibrium of this upper sub-game, we first find the best response R_m^* for each leader, i.e., the strategy that maximizes U_m when the strategies of the

other leaders are fixed [40].To this end, we first take the derivative of U_m:

$$\frac{\partial dU_m}{\partial dR_m} = \sum_{n=1}^{N} \frac{B_n \sum_{i\in\mathcal{M}_{-m}} R_i \left(1 + \ln\left(\frac{B_n R_m}{\sum_{i=1}^{M} R_i}\right)\right)}{\left(R_m + \sum_{i\in\mathcal{M}_{-m}} R_i\right)\right)^2} - 1. \tag{B.15}$$

To find R_m^*, we solve $\frac{\mathrm{d}U_m}{\mathrm{d}R_m} = 0$, i.e.,

$$\sum_{n=1}^{N} \frac{B_n \sum_{i\in\mathcal{M}_{-m}} R_i \left(1 + \ln\left(\frac{B_n R_m}{\sum_{i=1}^{M} R_i}\right)\right)}{\left(R_m + \sum_{i\in\mathcal{M}_{-m}} R_i\right)\right)^2} - 1 = 0. \tag{B.16}$$

Since the leaders' utility functions are the same, we have $\sum_{i\in\mathcal{M}_{-m}} R_i = (M-1)R_m$. Then, (B.16) becomes

$$\begin{aligned}
\frac{\partial U_m}{\partial R_m} = \sum_{n=1}^{N} \frac{B_n \sum_{i\in\mathcal{M}_{-m}} R_i \left(1 + \ln\left(\frac{B_n R_m}{\sum_{i=1}^{M} R_i}\right)\right)}{(R_m + \sum_{i\in\mathcal{M}_{-m}} R_i)^2} - 1 = 0, \\
\sum_{n=1}^{N} \frac{B_n \sum_{i\in\mathcal{M}_{-m}} R_i \left(1 + \ln\left(\frac{B_n R_m}{\sum_{i=1}^{M} R_i}\right)\right)}{(R_m + \sum_{i\in\mathcal{M}_{-m}} R_i)^2} = 1, \\
\sum_{n=1}^{N} \frac{B_n (M-1) R_m \left(1 + \ln\left(\frac{B_n R_m}{M R_m}\right)\right)}{(M R_m)^2} = 1, \\
\sum_{n=1}^{N} \frac{B_n (M-1) \left(1 + \ln\left(\frac{B_n}{M}\right)\right)}{M^2 R_m} = 1, \\
\frac{M-1}{M^2} \sum_{n=1}^{N} B_n \left(1 + \ln\left(\frac{B_n}{M}\right)\right) = R_m.
\end{aligned} \tag{B.17}$$

Thus,

$$R_m^* = \frac{M-1}{M^2} \sum_{n=1}^{N} B_n \left(1 + \ln\left(\frac{B_n}{M}\right)\right), \tag{B.18}$$

is the optimal strategy of leader m. Since R_m^* is uniquely defined by constants, i.e., M and B_n, the equilibrium of this upper sub-game exists and is unique. Moreover, the convergence to this equilibrium is also guaranteed. Since the uniqueness and

the convergence to the follower's sub-game are guaranteed, the considered Stackelberg game admits a unique Stackelberg equilibrium, and the convergence to the Stackelberg equilibrium is also guaranteed.

Appendix C

Proof in Chapter 5

C.1 The proof of Lemma 5.1

We will prove that $\boldsymbol{\eta}'$ satisfies the Karush–Kuhn–Tucker (KKT) conditions for non-convex optimization problems [129]. Moreover, since (5.15) and (5.16) are differentiable and satisfy linearity constraint qualification, strong duality holds, and thus $\boldsymbol{\eta}'$ is the global optimal solution of (**P3**). Next, we prove that $\boldsymbol{\eta}'$ satisfies the KKT conditions as follows. The first condition is:

$$f_i(\boldsymbol{\eta}') \leq 0. \tag{C.1}$$

This condition is always satisfied since there is no inequality constraint ($f_i(\cdot)$) in (**P3**). The second condition is:

$$h_k(\boldsymbol{\eta}') = 0 = \sum_{s=1}^{\sigma}(\eta_k^s) - \eta_k, \forall k \in \mathcal{N}. \tag{C.2}$$

The second condition is always satisfied since it is included in (5.19). The third condition is

$$\lambda_k \geq 0, \forall k \in \mathcal{N}. \tag{C.3}$$

From (5.19), we have

$$\lambda_k = -\ln(\tau)\sum_{n=1}^{N}(\eta_n^s) - 2p_k^A\sum_{n=1}^{N}(0.5 - p_n^A)\eta_n^s = \big(-\ln(\tau) - p_k^A\big)\sum_{n=1}^{N}(\eta_n^s) + 2p_k^A\sum_{n=1}^{N}(\eta_n^s). \tag{C.4}$$

Since $\tau \leq 0.001$ and $p_k^A < 1$, we have $\lambda_k > 0$, and thus the third condition is satisfied. The fourth condition is:

$$\lambda_k h_k(\boldsymbol{\eta}') = 0. \tag{C.5}$$

Similar to (C.2), this condition is always satisfied since there is no inequality constraint in (**P3**). The fifth condition is

$$\begin{aligned} \nabla f_o(\boldsymbol{\eta}') + \sum_{k=1}^{N} \lambda_k \nabla h_k(\boldsymbol{\eta}') = 0, \\ \lambda_k + \ln(\tau) \sum_{n=1}^{N} (\eta_n^s) + 2p_k^A \sum_{n=1}^{N} (0.5 - p_n^A)\eta_n^s = 0, \forall k \in \mathcal{N}, \forall s = 1, \dots \sigma, \end{aligned} \tag{C.6}$$

which is included in (5.19). As a result, $\boldsymbol{\eta}'$ satisfies all KKT conditions, and thus the proof is completed.

C.2 The proof of Lemma 5.2

It follows from Lemma C.2 that $\boldsymbol{\eta}'$ is the global optimal solution of (**P3**), and thus it satisfies (5.16). Moreover, constraints (5.16) and (5.14) are identical. Therefore, $\boldsymbol{\eta}'$ satisfies (5.14). Furthermore, the objective function (5.12) of (**P2**) is constant. As a result, if $\boldsymbol{\eta}'$ satisfies (5.13), $\boldsymbol{\eta}'$ is the global optimal solution of (**P2**). The proof is completed.

C.3 The proof of Theorem 5.1

First, we prove that for any specific σ, we can straightforwardly derive $\mathbf{x}$. Substituting $\varsigma = \sigma$ into (5.9), we have

$$x_s \geq \frac{\sigma - s + 1}{S}, \forall s = 1, \dots, S. \tag{C.7}$$

This means that $x_s > 0$, $\forall s \leq \sigma$. Then, substituting $\varsigma = \sigma$ into (5.10), we have

$$x_s \leq \sigma - s + 1, \forall s = 1, \dots, S. \tag{C.8}$$

This means that $x_s \leq 0$, $\forall s > \sigma$. Moreover, since $\mathbf{x}$ are binary, we have $x_s = 1$, $\forall s = 1, \dots, \sigma$ and $x_s = 0$, $\forall s > \sigma$.

As a result, (5.8) becomes

$$\left(\sum_{n=1}^{N} (0.5 - p_n^A)\eta_n^s \right)^2 \geq -0.5 \ln(\tau) \sum_{n=1}^{N} (\eta_n^s)^2, \forall s \leq \sigma, \tag{C.9}$$

and

$$\left(\sum_{n=1}^{N}(0.5-p_n^A)\eta_n^s\right)^2 \geq 0, \forall s > \sigma. \tag{C.10}$$

Since $\boldsymbol{\eta}'$ satisfies (5.13), it also satisfies (C.9). Moreover, since $p_n^A < 0.5$ and $\eta_n^s > 0$, (C.10) is always satisfied. As a result, $\{\boldsymbol{\eta}', \mathbf{x}, \sigma\}$ satisfy all constraints of (**P1**), and thus it is a feasible solution to (**P1**). The proof is now completed.

Bibliography

[1] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no.3, pp. 2084-2123, Jan. 2016.

[2] "Global Charts," *CoinMarketCap*. [Online]. Available: `https://coinmarketcap.com/charts/`. Accessed on: 03-Nov-2018.

[3] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[4] GSMA Intelligence, "The Mobile Economy 2019," GSM Association, 2019. [Online]. Available: `https://www.gsmaintelligence.com/research/?file=b9a6e6202ee1d5f787cfebb95d3639c5&download`. [Accessed: 16-Aug-2019]

[5] L. Papachristou, "Report: US$32.7 Billion Lost in Telecom Fraud Annually," *Organized Crime and Corruption Reporting Project*. [Online]. Available: https://www.occrp.org/en/27-ccwatch/cc-watch-briefs/9436-report-us-32-7-billion-lost-in-telecom-fraud-annually. [Accessed: 16-Aug-2019].

[6] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities," in *IEEE Access*, vol. 7, pp. 85727–85745, Jun. 2019.

[7] Y. Xiao, N. Zhang, W. Lou and Y. T. Hou YT, "A survey of distributed con-

sensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 2, no. 22, pp. 1432-1465, Jan, 2020.

[8] A. Back et al. (Oct. 2008). Enabling blockchain innovations with pegged sidechains. [Online]. Available: http://kevinriggen.com/files/sidechains.pdf

[9] L. H. Lee et al., "All one needs to know about Metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda,", Oct. 2021 *arXiv preprint arXiv:2110.05352*. [Online]. Available: https://arxiv.org/abs/2110.05352

[10] L. Luu, D. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making Smart Contracts Smarter," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS16*, Vienna, Austria, Oct. 2016, pp. 254-269.

[11] G. Macia-Fernandez, P. Garcia-Teodoro, and J. Diaz-Verdejo, "Fraud in roaming scenarios: an overview," in *IEEE Wireless Communications*, vol. 16, no. 6, pp. 88–94, Dec. 2009.

[12] 3GPP, "3GPP TS 22.031 V15.0.0," Technical Specification 22.031, Jun-2018.

[13] GSMA, "GSMA Speeds Up The Transfer Of Roaming Call Records," *Newsroom*, 21-Mar-2012. [Online]. Available: https://www.gsma.com/newsroom/press-release/gsma-speeds-up-the-transfer-of-roaming-call-records/. [Accessed: 13-Nov-2019].

[14] Starhome Mach, "Starhome Mach: Operator's Roaming Fraud Losses Can Reach €40,000 Per Hour," *PR Newswire: press release distribution, targeting, monitoring and marketing*, 29-Jun-2018. [Online]. Available: https://www.prnewswire.com/news-releases/starhome-mach-operators-roaming-fraud-losses-can-reach-40000-per-hour-598836021.html. [Accessed: 16-Sep-2019].

[15] IBM, "Reimagining telecommunications with blockchains," *IBM Institute for Business Value*. [Online]. Available: https://www.ibm.com/thought-leadership/institute-business-value/report/blockchaintelco. [Accessed: 16-Aug-2019].

[16] Deutsche Telekom AG, "Deutsche Telekom and SK Telecom pave the way for the future," *Deutsche Telekom*, 26-Feb-2019. [Online]. Available: https://www.telekom.com/en/media/media-information/archive/deutsche-telekom-and-sk-telecom-pave-the-way-for-the-future-564180. [Accessed: 16-Aug-2019].

[17] M. Boddy, "EEA Publishes Blockchain Uses for T-Mobile and Other Major Telecoms," *Cointelegraph*, 30-Aug-2019. [Online]. Available: https://cointelegraph.com/news/enterprise-ethereum-alliance-publishes-on-blockchain-uses-in-telecoms. [Accessed: 20-Sep-2019].

[18] S. Nakamoto. (May 2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[19] "Bitcoin Energy Consumption Index," *Digiconomist*. [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption. [Accessed: 13-Nov-2019].

[20] A. Kulichevskiy. (03-Oct-2017). *Bubbletone blockchain white paper*. [Online]. Available: `https://icos.icobox.io/uploads/whitepaper/2017/10/59e8dcfa89537.pdf` [Accessed: 16-Aug-2019].

[21] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake (extended abstract)," *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 3, pp. 34–37, Dec. 2014.

[22] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," *International Conference on Financial Cryptography and Data Security*.

Springer, pp. 142–157, 2016.

[23] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol," *Advances in Cryptology – CRYPTO 2017 Lecture Notes in Computer Science*, pp. 357–388, 2017.

[24] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, pp. 51–68, 2017.

[25] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv preprint arXiv:1710.09437*, 2017.

[26] J. Kwon, "Tendermint: Consensus without mining," *Draft v. 0.6, fall*, 2014.

[27] Y. Xiao, N. Zhang, W. Lou and Y. T. Hou, "A Survey of Distributed Consensus Protocols for Blockchain Networks," 2020, *arXiv preprint arXiv:1904.04098*. [Online]. Available: https://arxiv.org/abs/1904.04098

[28] J. Dilley et al., "Strong federations: an interoperable blockchain solution to centralized third-party risks," 2016, *arXiv:1612.05491*.

[29] S. Lerner. (Jan. 2016). Drivechains, Sidechains and Hybrid 2-way Peg Designs [Online]. Available: `https://docs.rsk.co/Drivechains_Sidechains_and_Hybrid_2-way_peg_Designs_R9.pdf`.

[30] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, pp. 1-32, 2014.

[31] A. Garoffolo and R. Viglione, "Sidechains: Decoupled consensus between chains," 2018, *arXiv:1812.05441*.

[32] P. Gaži, A. Kiayias and D. Zindros, "Proof-of-Stake Sidechains," in *2019 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, May 18-19, 2019, pp. 139-156.

[33] A. Garoffolo, D. Kaidalov and R. Oliynykov, "Zendoo: a zk-SNARK verifiable cross-chain transfer protocol enabling decoupled and decentralized sidechains," 2020, *arXiv:2002.01847*.

[34] G. Wood. (Nov. 2016). POLKADOT: Vision for a heterogeneous multi-chain framework. [Online]. Available: `https://polkadot.network/PolkaDotPaper.pdf`

[35] J. Kwon and E. Buchman. (2019). Cosmos Whitepaper. [Online]. Available: `https://v1.cosmos.network/resources/whitepaper`

[36] M. Herlihy, B. Liskov and L. Shrira, "Cross-chain deals and adversarial commerce," *VLDB Journal*, Aug. 2021, https://doi.org/10.1007/s00778-021-00686-1.

[37] V. Zakhary, D. Agrawal, A. El Abbadi, "Atomic commitment across blockchains", 2019, *arXiv:1905.02847*

[38] J. Garay, A. Kiayias, and N. Leonardos, "The Bitcoin Backbone Protocol: Analysis and Applications," in *Advances in Cryptology - EUROCRYPT 2015 Lecture Notes in Computer Science*, vol. 9057. E. Oswald, M. Fischlin, Eds. Berlin: Springer, 2015, pp. 281–310.

[39] P. Daian, R. Pass, E. Shi, "Snow white: Robustly reconfigurable consensus and applications to provably secure proof of stake," in *International Conference on Financial Cryptography and Data Security*, Saint Kitts and Nevis, Feb. 18-22, 2019, pp. 23-41.

[40] Z. Han, D. Niyato, W. Saad, T. Basar, and A. Hjørungnes, *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge: Cambridge University Press, 2012.

[41] H. Xu, Z. Li, Z. Li, X. Zhang, Y. Sun and L. Zhang, "Metaverse native communication: A blockchain and spectrum prospective," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, Seoul, South Korea, May 16-20, 2022, pp. 7-12.

[42] T. R. Gadekallu et al., "Blockchain for the metaverse: A review," 2022, *arXiv:2203.09738*.

[43] Q. Yang, Y. Zhao, H. Huang, Z. Xiong, J. Kang and Z. Zheng, "Fusing Blockchain and AI With Metaverse: A Survey," *IEEE Open Journal of the Computer Society*, vol. 3, pp. 122-136, July 2022.

[44] H. J. Jeon, H. C. Youn, S. M. Ko and T. H. Kim, "Blockchain and AI meet in the Metaverse," in *Advances in the Convergence of Blockchain and Artificial Intelligence*. T. M. Fernadez-Carames and P. Fraga-Lamas, Eds. London, UK:Intechopen, 2022.

[45] J. Ryu, S. Son, J. Lee, Y. Park and Y. Park, "Design of secure mutual authentication scheme for Metaverse environments using blockchain," *IEEE Access*, vol. 10, pp. 98944-98958, Sep. 2022.

[46] M. Ersoy and R. Gürfidan, "Blockchain-based asset storage and service mechanism to metaverse universe: Metarepo" *Transactions on Emerging Telecommunications Technologies*. vol. 34, no. 1, pp. 4658-4675, Jan. 2023.

[47] T. Maksymyuk, J. Gazda, G. Bugár, V. Gazda, M. Liyanage and M. Dohler, "Blockchain-Empowered service management for the decentralized Metaverse of things," *IEEE Access*, vol. 10, pp. 99025-99037, Sep. 2022.

[48] J. Chen, H. Xiao, M. Hu, and C. M. Chen, "A blockchain-based signature exchange protocol for metaverse," *Future Generation Computer Systems*, vol. 142, no. 1, pp. 237-247, Jan. 2023.

[49] M. Zamani, M. Movahedi and M. Raykova "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, Toronto, Canada, Oct. 15-19, 2018, pp. 931-948.

[50] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2016, pp. 17-30.

[51] H. Dang, T. T. Dinh, D. Loghin, E. C. Chang, Q. Lin and B. C. Ooi, "Towards scaling blockchain systems via sharding," in *Proceedings of the 2019 international conference on management of data*, June 2019, pp. 123-140.

[52] X. Cai et al., "A Sharding Scheme-Based Many-Objective Optimization Algorithm for Enhancing Security in Blockchain-Enabled Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7650-7658, Nov. 2021.

[53] C. Huang et al., "RepChain: A Reputation-Based Secure, Fast, and High Incentive Blockchain System via Sharding," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4291-4304, Mar. 2021.

[54] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta and B. Ford, "OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding," in *2018 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, May 20-24, 2018, pp. 583-598.

[55] D. Tennakoon and V. Gramoli, "Dynamic blockchain sharding," in *5th International Symposium on Foundations and Applications of Blockchain 2022 (FAB 2022)*, California, USA, Jun. 3, 2022, pp. 1-17.

[56] Cardano, "Ouroboros Proof of Stake Algorithm," *Cardano*. [Online]. Available: `https://cardanodocs.com/cardano/proof-of-stake/`. [Accessed: 14-Mar-2019].

[57] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Towards Secure Blockchain-enabled Internet of Vehicles: Optimizing Consensus Management Using Reputation and Contract Theory," *arXiv preprint arXiv:1809.08387*, 2018.

[58] M. Li, L. Zhu, and X. Lin, "Efficient and Privacy-preserving Carpooling using Blockchain-assisted Vehicular Fog Computing," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[59] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, 2018.

[60] B. Leiding and W. V. Vorobev. "Tezos-based Vehicular Ad Hoc Blockchains." 2018.

[61] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2016.

[62] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2017.

[63] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling Localized Peer-to-Peer Electricity Trading Among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, 2017.

[64] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Lsb: A lightweight scalable blockchain for iot security and privacy," *arXiv preprint arXiv:1712.02969*, 2017.

[65] D. Ivan, "Moving toward a blockchain-based method for the secure storage of patient records," *ONC/NIST Use of Blockchain for Healthcare and Research Workshop. Gaithersburg, Maryland, United States*, 2016.

[66] K. Peterson, R. Deeduvanu, P. Kanjamala, and K. Boles, "A blockchain-based approach to health information exchange networks," *Proc. NIST Workshop Blockchain Healthcare*, vol. 1, pp. 1-10, 2016.

[67] G. Baxendale, "Can Blockchain Revolutionise EPRs?," *Itnow*, vol. 58, no. 1, pp. 38–39, 2016.

[68] A. Sudhan and M. J. Nene, "Employability of blockchain technology in defence applications," *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, 2017.

[69] A. McAbee, M. Tummala, and J. McEachen, "Military Intelligence Applications for Blockchain Technology," *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[70] W. Wang, D. Niyato, P. Wang, and A. Leshem, "Decentralized Caching for Content Delivery Based on Blockchain: A Game Theoretic Perspective," *2018 IEEE International Conference on Communications (ICC)*, 2018.

[71] N. Herbaut and N. Negru, "A model for collaborative blockchain-based video delivery relying on advanced network services chains," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 70–76, 2017.

[72] "Hashrate Distribution An estimation of hashrate distribution amongst the largest mining pools," . [Online]. Available: `https://www.blockchain.com/pools?`. Accessed on: 03-Nov-2018.

[73] S. King, "Primecoin: Cryptocurrency with prime number proof-of-work," Self-published Papers, Jul. 2013. [Online]. Available:`http://primecoin.io/bin/primecoin-paper.pdf`. Accessed on: 03-Nov-2018.

[74] A. Shoker, "Sustainable blockchain through proof of exercise," *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA, pp. 1–9, Oct. 2017.

[75] M. Ball, A. Rosen, M. Sabin, and P. N. Vasudevan, "Proofs of Useful Work," *IACR Cryptology ePrint Archive*, 2017.

[76] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz, "Permacoin: Repurposing bitcoin work for data preservation," *2014 IEEE Symposium on Security and Privacy*, pp. 475–490, May 2014.

[77] H. Kopp, C. Bosch, and F. Kargl,"Koppercoin – a distributed file storage with financial incentives," *12th International Conference on Information Security Practice and Experience*, Zhangjiajie, China, pp. 79–93, Nov. 2016.

[78] Protocol Labs, "Filecoin: A decentralized storage network," Protocol Labs, Tech. Rep., Aug. 2017.

[79] A. Miller, A. Kosba, J. Katz, and E. Shi, "Nonoutsourceable Scratch-Off Puzzles to Discourage Bitcoin Mining Coalitions," *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS 15*, 2015.

[80] P. Daian, I. Eyal, A. Juels, and E. G. Sirer, "(Short Paper) PieceWork: Generalized Outsourcing Control for Proofs of Work," *Financial Cryptography and Data Security Lecture Notes in Computer Science*, pp. 182–190, 2017.

[81] S. Park, K. Pietrzak, J. Alwen, G. Fuchsbauer, and P. Gazi, "Spacecoin: A cryptocurrency based on proofs of space," MIT, Tech. Rep., Jun. 2015.

[82] J. Blocki and H.-S. Zhou, "Designing Proof of Human-Work Puzzles for Cryptocurrency and Beyond," *Theory of Cryptography Lecture Notes in Computer Science*, pp. 517–546, 2016.

[83] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," Self-published Paper, Aug. 2012. [Online]. Available: `https://peercoin.net/assets/paper/peercoin-paper.pdf`

[84] I. Bentov, R. Pass, and E. Shi, "Snow white: Provably secure proofs of stake," *IACR Cryptology ePrint Archive*, vol. 2016, p. 919, Sep. 2016.

[85] C. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin meets strong consistency," *Proceedings of the 17th International Conference on Distributed Computing and Networking - ICDCN 16*, 2016.

[86] R. Pass and E. Shi, "Hybrid Consensus: Efficient Consensus in the Permissionless Model," *31st International Symposium on Distributed Computing (DISC 2017)*, vol. 91, Vienna, Austria, pp. 39:1– 39:16, Oct. 2017.

[87] D. Bluetower, "Cardano (ADA) – Ouroboros hydra and Cardano scalability to Visa level TPS," *ELEVENEWS*, 19-Jan-2019. [Online]. Available: `https://elevenews.com/2019/01/19/cardano-ada-ouroboros-hydra-and-cardano-scalability-to-visa-level-tps/`. [Accessed: 14-Mar-2019].

[88] Infinity Stones, "Overview of Tezos' Economics Model," Medium, 04-Sep-2018. [Online]. Available:https://medium.com/infinity-stones/overview-of-tezos-economical-model-e197f773c6c4?fbclid=IwAR1O6qawaVSUPbK5X2B0pCnErWy97X_35rsqQkPqawLQSsNgUFYwdoAxf1U. [Accessed: 14-Mar-2019].

[89] "Tezos (XTZ) Review – True Decentralized Governance for Blockchain," *ChainBits*, 31-Jan-2019. [Online]. Available: https://www.chainbits.com/reviews/tezos-review/?fbclid=IwAR36FPU7vavgPq__qDs-oCWwlODTcprjBLOVtb3UdAO-LoZ_8BJyE4CrIwg. [Accessed: 14-Mar-2019].

[90] Interchain Foundation, "A Beginner's Guide to Ethermint," *Cosmos Blog*, 27-Oct-2017. [Online]. Available:https://blog.cosmos.network/a-beginners-guide-to-ethermint-38ee15f8a6f4?fbclid=IwAR0OpLh7Spzf_-afrOF69U0RYmqE2aSPyDL3EXLe8NdAVHWjf9xALyLrNRY. [Accessed: 14-Mar-2019].

[91] B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," *Annual International Cryptology Conference*, Santa Barbara, California, USA, Aug. 15-19, 1999, pp. 148-164.

[92] Jotunn, "How Many Stake Pools?," *Cardano Forum*, 21-Sep-2018. [Online]. Available: https://forum.cardano.org/t/how-many-stake-pools/16132/12. [Accessed: 25-Sep-2019].

[93] "Stakecube," *Crypto Shib*. [Online]. Available: https://cryptoshib.com/stakecube/. [Accessed: 13-May-2019].

[94] "Earn profits by holdings cryptoassets," *MyCointainer*. [Online]. Available: https://www.mycointainer.com/. [Accessed: 13-May-2019].

[95] Max and Max, "Our Fee Structure.," *Medium*. [Online]. Available: `https://medium.com/brunchpool/our-fee-structure-5d951bc16976`. [Accessed: 26-May-2019]

[96] M. X. Goemans, *Advanced algorithms*. Massachusetts Institute of Technology. Laboratory for Computer Science, 1994.

[97] F. Glover, "Improved Linear Integer Programming Formulations of Nonlinear Integer Problems," in *Management Science*, vol. 22, no. 4, pp. 455–460, Dec. 1975.

[98] Cardano, "Incentives and staking in Cardano," *Incentives and staking in Cardano*. [Online]. Available:`https://staking.cardano.org/`. [Accessed: 13-May-2019].

[99] StakingRewards, "Algorand," *Digital Asset Research Platform for Staking & Dividends*. [Online]. Available: https://stakingrewards.com/asset/algo. [Accessed: 16-Aug-2019].

[100] StakingRewards, "Cosmos," *Digital Asset Research Platform for Staking & Dividends*. [Online]. Available: https://stakingrewards.com/asset/atom. [Accessed: 16-Aug-2019].

[101] StakingRewards, "Tezos," *Digital Asset Research Platform for Staking & Dividends*. [Online]. Available: https://stakingrewards.com/asset/xtz. [Accessed: 16-Aug-2019].

[102] StakingRewards, "NEM," *Digital Asset Research Platform for Staking & Dividends*. [Online]. Available: https://stakingrewards.com/asset/xem. [Accessed: 16-Aug-2019].

[103] A. Singh et al., "Sidechain technologies in blockchain networks: An examination and state-of-the-art review," *Journal of Network and Computer Applications*, vol. 149, Jan. 2020, doi: https://doi.org/10.1016/j.jnca.2019.102471

[104] S. Gupta, J. Hellings, M. Sadoghi, "Fault-Tolerant distributed transactions on blockchain," *Synthesis Lectures on Data Management*, vol. 16, no.1, pp.1-268, Feb. 2021.

[105] W. Sun, J. Liu, Y. Yue and P. Wang, "Joint Resource Allocation and Incentive Design for Blockchain-Based Mobile Edge Computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 6050-6064, Sept. 2020.

[106] J. Kang, Z. Xiong, D. Niyato, P. Wang, D. Ye and D. I. Kim, "Incentivizing Consensus Propagation in Proof-of-Stake Based Consortium Blockchain Networks," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 157-160, Feb. 2019.

[107] M. S. Iftikhar, N. Javaid, O. Samuel, M. Shoaib and M. Imran, "An Incentive Scheme for VANETs based on Traffic Event Validation using Blockchain," *2020 International Wireless Communications and Mobile Computing (IWCMC)*, Limassol, Cyprus, 2020, pp. 2133-2137.

[108] J. Kang, Z. Xiong, D. Niyato, S. Xie and D. I. Kim, "Securing Data Sharing from the Sky: Integrating Blockchains into Drones in 5G and Beyond," *IEEE Network*, vol. 35, no. 1, pp. 78-85, Feb. 2021.

[109] Z. Xiong, J. Kang, D. Niyato, P. Wang and H. V. Poor, "Cloud/Edge Computing Service Management in Blockchain Networks: Multi-Leader Multi-Follower Game-Based ADMM for Pricing," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 356-367, Apr. 2020.

[110] S. Guo, Y. Dai, S. Guo, X. Qiu and F. Qi, "Blockchain Meets Edge Computing: Stackelberg Game and Double Auction Based Task Offloading for Mobile Blockchain," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5549-5561, May 2020.

[111] C. T. Nguyen, D. N. Nguyen, D. T. Hoang, H. Pham, N. H. Tuong and E. Dutkiewicz, "Blockchain and Stackelberg Game Model for Roaming Fraud Prevention and Profit Maximization," *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, Seoul, South Korea, 2020, pp. 1-6.

[112] J. B. Rosen, "Existence and Uniqueness of Equilibrium Points for Concave N-Person Games," *Econometrica*, vol. 33, no. 3, p. 520, 1965.

[113] MathWorks, "Fmincon," *Mathworks*. [Online]. Available: `https://www.mathworks.com/help/optim/ug/fmincon.html`.

[114] "Jump in and be a part of Meta Horizon Worlds," *Meta*. [Online]. Available: https://www.meta.com/au/horizon-worlds/. [Accessed: 12-Feb-2023].

[115] G. Yu, X. Wang, K. Yu, W. Ni, J. A. Zhang and R. P. Liu, "Survey: Sharding in Blockchains," *IEEE Access*, vol. 8, pp. 14155-14181, Jan. 2020.

[116] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu and Y. Liu, "A Survey on the Scalability of Blockchain Systems," *IEEE Network*, vol. 33, no. 5, pp. 166-173, Oct. 2019.

[117] Tennakoon D and Gramoli V "Dynamic blockchain sharding," in *5th International Symposium on Foundations and Applications of Blockchain*, Oakland , California, USA, June 3, 2022, pp. 1-17.

[118] H. Liu, Y. Zhang and T. Yang, "Blockchain-enabled security in electric vehicles cloud and edge computing," *IEEE Network*, vol. 32, no. 3, pp. 78-83, June 2018.

[119] I. Martinez, S. Francis and A. S. Hafid, "Record and reward Federated Learning contributions with blockchain," in *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, Guilin, China, Oct. 17-19, 2019, pp. 50-57.

[120] Z. Hong, S. Guo and P. Li, "Scaling blockchain via layered sharding," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3575-3588, Dec. 2022.

[121] Z. Cai et al., "Benzene: Scaling blockchain with cooperation-based sharding," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 2, pp. 639-654, Feb. 2023.

[122] C. Beekhuizen , "Validated, staking on eth2: #3 - sharding consensus," *Bloomberg*, Mar. 27, 2020. [Online]. Available: `https://blog.ethereum.org/2020/03/27/sharding-consensus.` [Accessed: 12-Feb-2023].

[123] T. Ghosh, A. Roy and S. Misra, "B2H: Enabling delay-tolerant blockchain network in healthcare for Society 5.0," *Computer Networks*, vol. 210, pp. 108860-70, June. 2022.

[124] J. Gridley and O. Seneviratne, "Significant digits: Using large-scale blockchain data to predict fraudulent addresses," 2023, *arXiv:2301.01809*.

[125] M. Masmoudi, C. A. Zayani, I. Amous and F. Sèdes, "A new blockchain-based trust management model", *Procedia Computer Science*, vol. 192, pp. 1081-91, Jan. 2021.

[126] M. Mitzenmacher and E. Upfal, *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge: Cambridge University Press, 2017.

[127] J. Kronqvist, D. E. Bernal, A. Lundell and I. E. Grossmann, "A review and comparison of solvers for convex MINLP" *Optimization and Engineering*. vol. 20, no. 2, 20(2):pp. 397-455, June, 2019.

[128] R. M. Karp, "Reducibility among combinatorial problems ," in *50 Years of Integer Programming 1958-2008*. M. Junger et al, Eds. Heidelberg, Germany:Springer, 2010.

[129] S. P. Boyd and L. Vandenberghe, *Convex optimization*.Cambridge, UK: Cambridge University Press, 2011.

[130] "Stats," *Bitcoin Block Explorer*. [Online]. Available: `https://btc.com/stats`. Accessed on: 03-Nov-2018.

[131] L. M.GOODMAN, "Tezos—a self-amending crypto-ledger White paper," 2014. [Online]. Available: `https://www.tezos.com/static/papers/white_paper.pdf`. Accessed on: 03-Nov-2018.

[132] J. Kleinberg and Tardos Eva, "Algorithm design". Harlow: Pearson India, 2014.

[133] "PEER-TO-PEER," *BTCPOP*. [Online]. Available: `https://btcpop.co/home.php`. [Accessed: 13-May-2019].

[134] R. Henry, A. Herzberg and A. Kate, "Blockchain access privacy: Challenges and directions," *IEEE Security & Privacy*, vol. 16, no. 4, pp. 38-45, Aug. 2018.

[135] Q. Feng, D. He, S. Zeadally, M. K. Khan and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, pp. 45-58, Jan. 2019.

[136] StakingRewards, "Cardano," *Digital Asset Research Platform for Staking*

& Dividends. [Online]. Available: https://stakingrewards.com/asset/ada. [Accessed: 16-Aug-2019].

[137] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734-4746, Apr. 2020.

[138] H. Chen, S. A. Asif, J. Park, C. C. Shen and M. Bennis, "Blockchained Federated Learning with Model Validation and Proof-of-Stake Inspired Consensus," Jan. 2021, *arXiv:2101.03300*.

[139] "Buy & sell Crypto in minutes," . [Online]. Available: https://www.binance.com/en. Accessed on: 04-Nov-2020.

[140] "Bitcoin & Cryptocurrency Exchange," . [Online]. Available: https://www.kraken.com/. Accessed on: 04-Nov-2020.

[141] Selfkey, "A Comprehensive List of Cryptocurrency Exchange Hacks," *Selfkey*. [Online]. Available: https://selfkey.org/list-of-cryptocurrency-exchange-hacks/. Accessed on: 04-Nov-2020.

[142] S. Kovach, "Next for the Metaverse: Convincing you it's not just for kids," *CNBC*, Dec. 22, 2021. [Online]. Available: https://www.cnbc.com/2021/12/22/here-are-the-companies-building-the-Metaverse-meta-roblox-epic.html. [Accessed: 24-Dec-2021].

[143] Nissan, "Invisible-to-Visible (I2V)," *Nissan Motor Corporation* [Online]. Available: https://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/i2v.html. [Accessed: 14-Mar-2021].

[144] M. Rosenfeld "Analysis of bitcoin pooled mining reward systems," Dec.2011, *arXiv preprint arXiv:1112.4980*. [Online]. Available: https://arxiv.org/abs/1112.4980

[145] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, H. Pham, N. H. Tuong and Dutkiewicz E, "Blockchain-based Secure Platform for Coalition Loyalty Program Management," *2021 IEEE Wireless Communications and Networking Conference*, Nanjing, China, Mar. 29 - Apr.1 , 2021, pp. 1-6.

[146] C. T. Nguyen et al., "FedChain: Secure Proof-of-Stake-based Framework for Federated-blockchain Systems," *IEEE Transactions on Services Computing*, 2023, Early Access.