UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Blockchain Meets IoT: What Needs To Be Addressed

by

Guangsheng Yu

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2021

Certificate of Authorship/Originality

I, Guangsheng Yu 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 reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. I certify that the work in this thesis has not been previously submitted for a degree nor has it been submitted as a part of the requirements for other degree except as fully acknowledged within the text. This research is supported by the Australia Government Research Training Program.

Signatures: Production Note:
Signature removed prior to publication.

Date: May 31, 2021

© Copyright 2021 Guangsheng Yu

Dedication

To my parents Jiuhong Yu and Lixian Wang, and my wife, Aiyun Luo, for their endless support and love.

To my supervisors, for the academic guidance.

To my friends, for their encouragement.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor Prof. J. Andrew Zhang and Prof. Ren Ping Liu for the continuous support of my Ph. D study and research, for their patience, motivation, enthusiasm, and immense knowledge. My deepest thanks also goes to Dr. Wei Ni in CSIRO. Their guidance helped me in all the time of research and writing of this thesis. My research would have been impossible without their support and supervision.

My sincere thanks also goes to Dr. Xu Wang and Dr. Kan Yu, for their encouragement, insightful comments, and hard questions. A very special gratitude goes out to my mates Xuan Zha, Ping Yu, and Lizhang Tianyi, for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the awesome time with you guys. I would also like to thank everyone in GBDTC, UTS, and UCOT Australia who supported me along the way.

Last but not least, many thanks goes to my family: my parents Jiuhong Yu and Lixian Wang, and my wife, Aiyun Luo, for supporting me spiritually throughout my life.

Guangsheng Yu

Sydney, Australia, 2021

List of Publications

Published Journal Papers

J-1. **G. Yu**, X. Zha, X. Wang, W. Ni, K. Yu, P. Yu, J. A. Zhang, R. P. Liu and Y. J. Guo, "Enabling Attribute Revocation for Fine-Grained Access Control in Blockchain-IoT Systems," *IEEE Transactions on Engineering Management*, pp. 1-20, Feb, 2020.

J-2. **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.

J-3. **G. Yu**, X. Zha, X. Wang, W. Ni, K. Yu, J. A. Zhang and R. P. Liu, "A Unified Analytical Model for Proof-of-X Schemes," *Elsevier Computers & Security*, Jun, 2020.

J-4. **G. Yu**, L. Zhang, X. Wang, K. Yu, W. Ni, J. A. Zhang and R. P. Liu, "A Novel Dual-Blockchained Structure for Contract-Theoretic LoRa-based Information Systems," *Elsevier Information Processing and Management*, Jan, 2021.

J-5. X. Wang, **G. Yu**, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, K. Zheng and X. Niu, "Capacity of Blockchain based Internet-of-Things: Testbed and Analysis," *Elsevier Internet of Things*, vol. 8, Dec, 2019.

Published Conference Papers

C-1. **G. Yu**, X. Wang, X. Zha, J. A. Zhang and R. P. Liu, "An Optimized Round-Robin Scheduling of Speakers for Peers-to-Peers-based Byzantine Faulty Tolerance," *Proc. IEEE Globecom Workshops*, 2018.

C-2. X. Wang, X. Zha, **G. Yu**, W. Ni, R. P. Liu, Y. J. Guo, X. Niu and K. Zheng, "Attack and Defence of Ethereum Remote APIs," *Proc. IEEE Globecom Workshops*, 2018.

C-3. X. Wang, P. Yu, **G. Yu**, X. Zha, W. Ni, R. P. Liu and Y. J. Guo, "A High-Performance Hybrid Blockchain System for Traceable IoT Applications," *Network and System Security, 13th International Conference*, Dec, 2019.

Submitted Papers

C-1. X. Wang, **G. Yu**, R. P. Liu, et al, "Blockchain-Enabled Fish Provenance and Quality Tracking System," *IEEE Internet of Things Journal*.

C-2. P. Yu, W. Ni, **G. Yu**, H, Zhang, R. P. Liu and Q. Wen, "Efficient Anonymous Data Authentication for Vehicular Ad-Hoc Networks," *Hindawi Security and Communication Networks*.

C-3. X. Wang, W. Ni, X. Zha, **G. Yu**, R. P. Liu, N. Georgalas and A. Reeves, "Capacity Analysis of Public Blockchain," *Elsevier Computers & Security*.

ABSTRACT

Blockchain Meets IoT: What Needs To Be Addressed

by

Guangsheng Yu

The connection between Blockchain and Internet of Things (IoT) has no longer been futuristic. However, the research of Blockchain-based IoT is challenging. The traditional Blockchain technologies become gradually incapable of satisfying the growing market of IoT networks, and demand for significant improvements. This research proposes a variety of novel approaches, aiming to point out and address the key challenges from different aspects, i.e., consensus algorithms, Blockchain scalability, privacy/access control, and integration of the system.

The main contributions of this thesis are summarized as follows.

- This thesis proposes a Markov model explicitly capturing the weighted resource distribution of Proof-of-X (PoX) schemes in large-scale networks and unifying the analysis of different PoX schemes. The new model leads to the development of three new unified metrics for the evaluation, namely, *Resource Sensitivity*, *System Convergence*, and *Resource Fairness*, accounting for security, stability, and fairness, respectively. The generality and applicability of our model are validated by simulations in the context of the proposed metrics.

- This thesis proposes detailed comparison and quantitative evaluation of major Blockchain-based sharding mechanisms in a systematic and comprehensive way. Specifically, the contents include our insights analyzing the features and restrictions of the existing solutions. We also provide theoretical upper-bound of the throughput for each considered sharding mechanism. The remaining challenges and future research directions are also reviewed.

- This thesis proposes a new Blockchain-based IoT system which is compatible with attribute-based encryption (ABE) technique, and fine-grained access control is implemented with the attribute update enabled by integrating Chameleon Hash (CH) algorithms into the Blockchains. We design, and implement a new verification scheme over, a multi-layer Blockchain architecture to guarantee the tamper-resistance against malicious and abusive tampering. We also provide analysis and simulations showing that our system outperforms other solutions in terms of overhead, searching complexity, security, and compatibility.

- This thesis proposes a novel Dual-Blockchain-based Long Range (LoRa) system providing global cross-validated security, as a case study of integration between Blockchain and IoT. The rational behaviours of participators, the state-of-the-art contract-theoretic incentive mechanism, and the newly designed flow control protocol, can be secured by the tamper-resistance of Blockchains. Being part of the proposed incentive mechanism, the self-driven flow control scales both the Dual-Chain system and the LoRa network. We provide analysis and simulations showing that the system motivates the self-deployed LoRa Gateways in a more secure way, thus optimize the utilization of coverage while improving the Blockchain scalability and flexibility.

Contents

List of Figures

List of Tables

Abbreviation

ABE: Attribute-based encryption

BFT: Byzantine Fault Tolerance

BGP: Byzantine Generals Problem

BW: Bandwidth

CH: Chameleon Hash

CoSi: Collective signing

CR: Code rate

CP-ABE: Ciphertext-policy ABE

DAG: Directed acyclic graph

DDoS: Distributed Denial of Service

GHOST: Greedy Heaviest Observed Subtree

IC: Incentive Compatibility

IoT: Internet of Things

IR: Individual Rationality

KP-ABE: Key-policy ABE

MPT: Merkle Patricia Tree

LoRa: Long Range

LPWAN: Low-Power Wide-Area Network

NB-IoT: Narrowband IoT

NFV: Network-Function-Virtualization

NGN: Next Generation Network

P2P: Peer-to-peer

PBFT: Practical BFT

PoW: Proof-of-Work

PoS: Proof-of-Stake

PoTO: Proof-of-Task-Overhead

PoX: Proof-of-X

PVSS: Publicly verifiable secret sharing

SDN: Software-Defined-Network

SF: Spreading factor

SPOF: Single point of failure

SPV: Simple Payment Verification

TTN: The Things Network

UTXO: Unspent transaction output

VANETs: Vehicular ad hoc networks

VRF: Verifiable Random Function

VDF: Verifiable Delay Function

WAN: Wide area network

Chapter 1

Introduction

Internet of Things (IoT) has been emerged in both academia and industry since early 2000s as one of the core technologies and ecosystems of the Next Generation Network (NGN). Along with massive numbers of IoT sensors widely spread and huge IoT data streams transmitted, researchers have put more efforts into addressing the security issue due to centralized architecture. The Blockchain technology has been showing its strong performance on decentralized security when integrating with IoT network, i.e, Blockchain-based IoT. However, this can be challenging. Traditional Blockchain technologies can hardly handle large-scale IoT networks, while the inherent properties of IoT networks might also deteriorate the performance of Blockchains. More studies on solving these challenges and enhancing the integration are needed.

The thesis abstracts the aforementioned general challenges into the following four problems.

- The lack of models which are capable of evaluating consensus algorithms in large-scale networks discourages the participants who have no ability to quantify the profits.
- The trilemma of scalability-security-decentralization in Blockchains hinders the practical implementation of Blockchain-based IoT networks where high scalability is desired.
- The transparency of Blockchains and the lack of access control reveal the weak

point of being implemented in the context of permissioned IoT networks.

- The compatibility of existing architecture of Blockchain-based IoT is not good enough, e.g., the redundant IoT data reduce the throughput of Blockchains.

The thesis addresses each of the challenges, by proposing novel models, designs, and comprehensive comparison, as well as carrying out the corresponding numerical and simulation.

This chapter first introduces the background of IoT and Blockchain, as well as Blockchain-based IoT. This chapter then illustrates research topics and key research points, followed by the overview of this thesis.

1.1 Background

1.1.1 Overview of IoT

IoT plays a core role in NGN and digitizational society [61, 127]. As the name implies, IoT is more than just computer-to-computer or phone-to-phone communication. It means the connection and communication among things, indicating two important definitions. Firstly, the core and foundation of IoT is still the Internet, but with extension and expansion based on current internet. Secondly, clients have been extended to any physical items no matter nano and sophisticated machines or even living things with nano-chip transplanted inside the body. As the result, it has been successfully achieved the information exchange and communication between "things". IoT has been widely used in the integration of network by means of intelligent perception, identification technology and pervasive computing communication, etc, and therefore known as the third surge of the global development of information posterior to computers and internet.

According to the analysis report of International Data Corporation (IDC) updated in 2020 [87], the global spending regarding IoT technologies and services was

reached US$0.75 trillion in 2020, and would reach US$1.1 trillion by 2023. The statistical data includes intelligent and embedded systems, connectivity services, infrastructure, customized platforms of IoT, applications, security, analytics and IoT professional services. It predicted that by the end of 2023, the population of IoT-related equipment would reach approximately 13.15 billion [86]. Such a mass of equipment is driven by, to some extent, the intelligence system, and the data generated between clients and service providers. There still will have been a long way to go to widen the scope of implementation of IoT, however, it is expected that IoT will have a significant impact in many realms while IoT devices keep growing and become seamless.

Ranged from powerful end-devices (e.g., smartphones and vehicular ad hoc networks (VANETs)), to the cellular-based low-power wide-area networks (LPWAN) (e.g., Narrowband IoT (NB-IoT) and Category M1 (CAT-M1)) and non-cellular-based LPWAN (e.g., Long Range (LoRa) and Sigfox), it has been shown that IoT is taking a great impact on the many realms in diversified ways [195, 112, 114], including but not limited to:

- **Food and Beverage:** IoT can be used to track the entire process of consumer goods, from processing, sales to quality and storage ability. Given an example, the inspection and quarantine of outgoing beef from Australia to China require IoT technologies to be the foundation to collect and analyze the big data of even individual cattle from born to death.

- **Healthcare:** IoT technologies can be used to collect huge amount of healthcare data in an end-to-end manner, including event monitors, clinical research data, emergency response and home advanced care. This can increase the size of data, the efficiency of data collection, and enhance the treatment based on the better accuracy of prediction derived from the data, e.g., implementing

IoT technologies to defense against COVID-19 [155, 168].

- **Smart Transportation and Logistics:** Along with the rapid development of VANETs, vehicles have increasingly powerful capabilities of sensing, communicating and data processing. IoT technologies can be used to enhance these capabilities and track the product status throughout the path. Specifically, this information, which is shared among manufacturer, warehouse, retailer, and customers, includes the geography and time series tracking of location and operational status of vehicles.

- **Smart Home/Grid/City:** IoT technologies can be used in individual homes to enhance the quality of life by serving environment-friendly, intelligent, and automated facilities. Similarly, the smart grid and smart city are widely used to replace traditional infrastructure to provide reliable and efficient energy (e.g., electricity, gas, water) services.

Existing studies have dug deep into centralized architecture, such as a centralized management model [203] or a centralized global IoT Platform [182], and cloud-based architecture, such as a service-oriented hierarchical cloud architecture for IoT [75] or an IoT-fog-cloud based architecture [60]. Software-Defined-Network (SDN)-based IoT architecture [124] and Network-Function-Virtualization (NFV)-based architecture [164] are also recently proposed. These architectures aim to separate the data and control panels in a centralized manner, in order to reduce the data loading in centralized management. With the development of cloud, IoT cloud platform has become the mainstream architecture [157].

However, in order to collect and analyze the data transmitted among the exorbitant numbers of gateways and sensors in real-time context, it becomes difficult to have the centralized infrastructure placed on cloud or private data centre, and process the data in a collective way. There are two issues to which the existing

studies do not pay enough attention. Firstly, millions of end-devices constitute huge amount of data that a purely centralized infrastructure will be tough to guarantee the efficiency and scalability. Secondly, IoT security becomes an inevitable problem including the scalability, privacy, and authentication, which results in several vulnerability making an IoT network prone to failures and attacks, such as the single point of failure (SPOF) and Distributed Denial of Service (DDoS) [199, 26, 190, 76]. There have been found that the following critical issues exist in live environments:

- Amazon Cloud Service, Microsoft Azure, and Alicloud suffered from the service outage from 2017 to 2019, leading to the huge losses of data service for customers [20, 134, 204].
- The public cloud, the private cloud, and the hybrid cloud hardly tolerate the Byzantine Failure [89]. The Byzantine Failure takes malicious nodes into account, which is the most complicated failure mode in a distributed system [104].
- The cloud services lack trustworthy tamper-resistance for data storage and access control [122].

A centralized IoT architecture on cloud is not enough to ensure the IoT security. This leads to the core objective of the thesis, implementing a reliable and trustworthy Blockchain-based architecture for IoT networks. The architecture considers the cloud service as an optional infrastructure, on top of which a Blockchain system is built. The decentralized security of Blockchain has been thought as a disruptive solution to the challenges that existing IoT networks are facing.

1.1.2 Overview of Blockchains

1.1.2.1 Decentralization and the Byzantine Generals Problem

The typical decentralized security problem, namely the Byzantine Generals Problem (BGP), was first proposed by Lamport, Shostak and Pease in 1982. More specifically, this problem can be described as follows [104]:

There are three generals encircle an enemy city with one of them being the leader among them. They can only decide whether to attack or retreat. In hope of a plan for attacking the city being eventually finalized, each of them wishes to reach the consensus prior to the action. However, one of them being the traitor, targets on confusing others and disturbing the consistency of the final command. In order to solve this issue, the following requirements must hold:

- The loyal generals must follow the leader, if the leader is loyal;
- Any two loyal generals of them must finalize the same command, if the leader is a traitor.

In the case of the leader being likely a traitor, the general *Bob* cannot finalize a command of "attack" even when he receives an "attack" from others. Thus *Bob* needs to inquire of the other generals about the command they receive from the leader. However, it turns out to be a dilemma for *Bob* to recognize whether the leader or one of the generals is the traitor, if *Bob* receives a "attack" directly from the leader but a "retreat" from one of the other generals. The traitor in this case is also called a Byzantine failure, which is defined as a node either sending malicious messages or pretending to be a normal timeout failure [104]. They also show that there does not exist a solution for N generals, where $0 < N \leq 3$, being able to achieve the consistency in any situation in the presence of a single traitor. Yet, the problem is shown solvable for $N \geq 4$ where any general could find the majority in a consensus

under the premise of a responsive leader in that round, i.e., weak-synchronous; see detailed proof in [104]. Concretely, for any N that satisfies $N \geq 3f + 1$, where f denotes the number of traitors, there always exists a deterministic algorithm to reach the consensus.

1.1.2.2 Blockchain as a solution

Blockchain is one of the practical implementations. It is a decentralized cryptographic ledger that consists of blocks in a chained form by the hash value linking backwards, which is the kernel of Bitcoin first proposed in 2008 by Satoshi Nakamoto [135]. It runs in a peer-to-peer (P2P) network where each piece of data is shared among the network by data duplicately recorded on each of the nodes, without having to rely on a central authority and previously established trust relationship. Such a system can be trustworthy because a decentralized consensus algorithm is implemented to solve the Byzantine Generals Problem, leading to the tamper-resistance that prevents the double-spending attack. Specifically, different decentralized consensus algorithms exist depending on the type of Blockchain and on the requirements of the use cases. They are mainly categorized into two types, Nakamoto-based and Byzantine Fault Tolerance (BFT)-based consensus algorithms. The game theory and incentive are applied in Nakamoto-based consensus algorithms (e.g., Proof-of-Work (PoW)) to ensure to reach the consensus in a probabilistic manner in a complexity of $O(N)$, while BFT-based consensus algorithms (e.g., Practical BFT (PBFT) [40]) ensure a deterministic result by exchanging additional messages among the participants.

The cryptographic data structure (e.g., the attached digital signature and the irreversible hash function) and the decentralized consensus algorithm make the recorded data costly to be modified retroactively. This promotes Blockchain to be a suitable role for data recording, data storage, and identity management [137].

Ranged from cryptocurrency to IoT networks, Blockchain has shown its great potential to tackle critical security and trust challenges in various applications [135, 140], along with several open-source platforms being widely developed, e.g., Ethereum [38], EOS [30], and HyperLedger-Fabric [14].

1.1.3 Blockchain-based IoT: benefits and challenges

1.1.3.1 Benefits

A conventional IoT network is likely to involve multiple business parties who have an intent of ensuring secure and transparent data recording and sharing with each other. It is thus normal that a central authority needs to be involved to establish the trust relationship among all parties, which is severely prone to the SPOF, DDoS, and potential malicious authorities. The decentralized trustworthiness of Blockchain highly encourages the widely adoption of large-scale IoT applications in the sense of the tamper-resistance being surely guaranteed. It is also worth restating that the concept of Blockchain-based IoT is not opposed to the cloud-based IoT; see Section 1.1.1. A Blockchain system itself, which is composed of a decentralized P2P network, can be seamlessly implemented on top of any devices or infrastructures including the cloud-based platforms; see Figure 1.1. The flexibility also empowers the uses of Blockchain-based IoT systems.

1.1.3.2 Challenges

The studies of Blockchain-based IoT systems have been growing at a fast pace, however, revealing an increasing number of challenges when integrating the Blockchain and IoT technologies.

IoT networks require high scalability in order to handle huge amount of data stream transmitted among relatively weak performance of hardware devices. In hopes to overcome the aforementioned critical IoT security and trust issues, a

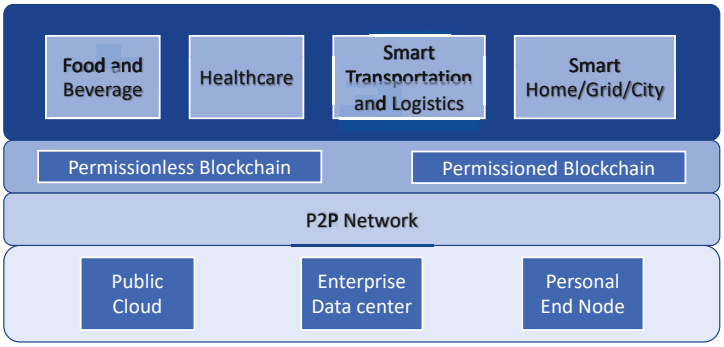


Figure 1.1 : The system architecture shows that Blockchain-based networks can be built on top of cloud-based platforms.

Blockchain-based IoT system also needs to delivery feasible solutions to meet different requirements, such as authentication service and self-motivated deployment, in a decentralized manner.

However, the trilemma of scalability-security-decentralization in Blockchain systems exists [37], stating that the challenge to achieve high scalability while at the same time featuring the decentralized security. This is because typical decentralized consensus algorithms can only satisfy two out of three in this trilemma. Typical Nakamoto-based consensus, e.g., PoW in Bitcoin, can only handle up to approximately 10 transactions per second with its maximum block size of 1MB and average 10 minutes block period [179] in a large-scale network under decentralized security. Typical BFT-based consensus, e.g., PBFT, can achieve a throughput of 1000 transactions per second by sacrificing the decentralization and limiting the block generators within a small-size committee. This severely hinders the use of Blockchains in the high-frequency data processing and trading, making Blockchains far from practical requirements.

This thesis focuses on the following four aspects:

- **Analytical models:** A decentralized architecture implies the needs of novel

evaluation approaches to be established such as an analytical model. IoT practitioners need a unified analytical model to conduct the cost-risk assessment to maintain a Blockchain-based IoT system or even a single node within the system.

- **Scalable protocols and designs:** A highly scalable Blockchain has been one of the most crucial requirements for IoT networks in a live context. This aspect corresponds to investigating novel protocols and designs in which the trilemma can be balanced.

- **Access control and privacy:** Enterprises have been playing a important role in IoT ecosystems. An effective and flexible access control policy to be applied in a Blockchain-based IoT system in order to protect the privacy of customers of theirs becomes necessary. It turns out that the inherent property of transparency in Blockchains conflicts with the requirement, which is the focus of this aspect.

- **Optimized architecture:** There have been a number of studies investigating and presenting the integration of Blockchain and IoT technologies. However, a simple architecture without dedicated optimizations cannot take full advantage of both technologies. On the contrary, Blockchains might incur negative impacts from the inherent properties of IoT networks in a straightforward integration.

1.2 Research Objectives

The challenges of implementing Blockchain-based IoT networks exist, based on which this thesis focuses on four categories, **analytical models**, **scalable protocols and designs**, **access control and privacy**, and **optimized architecture**. The research targets, key research points, and the structure of this thesis are illus-

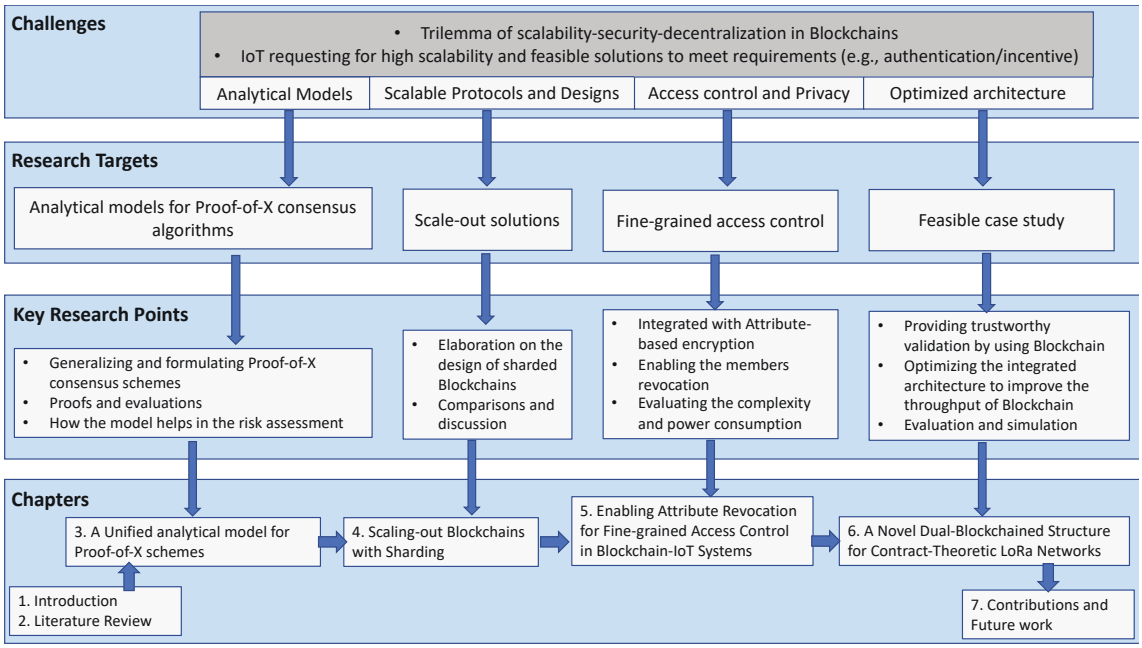


Figure 1.2 : Research targets, key research point, and chapters.

trated in Figure 1.2.

Analytical models - In regard to fulfilling the lack of a unified analytical model of PoX consensus scheme, what model/metrics should be used to reflect the generic properties?

Different from BFT-based consensus algorithms which were originally dedicated to solve the BGP and have been systematically learned, unified, and optimized (ranged from PBFT [40] to Zyzzyva [102] and Hotstuff [201]), the first Nakamoto-based consensus algorithms was first derived from Bitcoin [135], i.e., PoW. PoW utilizes the game theory and incentive to simplify the network conditions and provide a practical solution to BGP in large-scale networks. PoW has been extended to other virtual-mining-based variations (e.g., Proof-of-Stake (PoS)) and subsequently generalized to Proof-of-X (PoX)-based consensus algorithms [180, 186]. However, none of existing studies was able to provide a unified analytical model for PoX schemes. Such a model would be important to enable each practitioner to estimate its profit

against competitors based on its source. This is important for the traditional mining industry [118] and any public IoT services based on permissionless Blockchains in large-scale networks. Therefore, our research target for this aspect is to provide a unified analytical model specifically for PoX consensus schemes, which is discussed in Chapter 3. Mathematical proofs and evaluations are also presented in the chapter to elaborate on how the proposed model helps in the cost-risk assessment.

Scalable protocols and designs - Scale-out Blockchain technologies are promising to break the trillema, is there a generic framework to learn them, thus an explicit future can be pointed?

Scalability is important for IoT networks. The weak scalability of traditional Blockchain technologies severely affect the wide adoption due to the well-known trillema of decentralization-security-scalability in Blockchains. In regards to this issue, a number of solutions have been proposed, targeting to boost the scalability while preserving the decentralization and security. They range from modifying the on-chain data structure [152] and consensus algorithms [201] to adding the off-chain technologies [150]. Therein, one of the most practical methods to achieve horizontal scalability along with the increasing network size is sharding. As a scale-out solution, sharding partitions network into multiple shards so that the overhead of duplicating communication, storage, and computation in each full node can be avoided, leading to a boost of the scalability of Blockchains. Chapter 4 presents a comprehensive comparison and evaluation among existing major sharding mechanisms, in hopes of revealing the pain points of sharding technologies and pointing out the future direction.

Access control and privacy - The content needs to be editable to fulfill the changes of attributes in an ABE-based Blockchain that provides fine-grained access control, how the contradiction can be solved between this

requirement and the immutability of Blockchains?

Fine-grained access control is important in IoT networks in which the increasing number of devices and users contribute to the massive data needed to be accessible in a well-organized manner. Attribute-based encryption (ABE) has been considered to achieve fine-grained access control in Blockchains. However, members revocation which requires to modify the records conflicts with the immutability of Blockchains. Chapter 5 proposes a novel multilayer Blockchain-IoT data service system, which enables secure attribute updates in an ABE-based fine-grained access control mechanism for Blockchains. Numberical results are given to evaluate the complexity and power consumption.

Optimized architecture - How dedicated optimizations could be designed upon a Blockchain-based IoT application such that the combination could be natural, smooth, and effective?

Researchers have been doubting with a flood of designs presenting simple and stilted integration between Blockchain and IoT technologies. Without dedicated optimization, a LoRa system as a typical case study, the duplicated data (which is helpful for redundancy) becomes the cause of the throughput loss in Blockchains. Thus, a novel Dual-Chain-based LoRa system is proposed in Chapter 6 as a feasible case study to fill the gap. The system deliveries a trustworthy cross-validated-security in a decentralized manner, while a new incentive mechanism to improve the Blockchain throughput. A Monte-Carlo simulation is carried out to prove the feasibility.

1.3 Thesis Organization

This thesis is based on the research results of 4 journal papers. The thesis is organised as follows:

- *Chapter 1*: This chapter provides an overview of this research. The research background is presented in the first place to point out the challenges, followed by the research targets and key research points for implementing Blockchain-based IoT networks. This chapter also illustrates the structure of this thesis.

- *Chapter 2*: This chapter gives a comprehensive literature review of existing studies regarding the challenges of Blockchain-based IoT networks from the four highlighted aspects.

- *Chapter 3*: This chapter proposes a Markov model which captures explicitly the weighted resource distribution of PoX schemes in large-scale networks and unifies the analysis of different PoX schemes. The new model leads to the development of three new unified metrics for the evaluation, namely, Resource Sensitivity, System Convergence, and Resource Fairness, accounting for security, stability, and fairness, respectively. The generality and applicability of our model are validated by simulation results, revealing that among typically non-Fairness-oriented PoX schemes (such as PoW and PoS), the strongly restricted coinage-based PoS with a Pareto-distributed resource can offer the best performance on Resource Sensitivity, while Proof-of-Publication (PoP) with normal-distributed resource performs the best on System Convergence. Our simulations also reveal the important role of carefully designed Resource Fairness parameter in balancing Resource Sensitivity and System Convergence and improving the performance compared with other non-Fairness-oriented PoX schemes.

- *Chapter 4*: This chapter presents a survey focusing on sharding in Blockchains in a systematic and comprehensive way. We provide detailed comparison and evaluation of major sharding mechanisms, along with our insights analyzing the features and restrictions of the existing solutions. The remaining challenges

and future research directions are also reviewed.

- *Chapter 5*: This chapter proposes a new Blockchain-based IoT system which is compatible with ABE technique, and fine-grained access control is implemented with the attribute update enabled by integrating Chameleon Hash (CH) algorithms into the Blockchains. We design and implement a new verification scheme over a multilayer Blockchain architecture to guarantee the tamper resistance against malicious and abusive tampering. The system can provide an update-oriented access control, where historical on-chain data can only be accessible to new members and inaccessible to the revoked members. This is distinctively different from existing solutions, which are threatened by data leakage toward the revoked members. We also provide analysis and simulations showing that our system outperforms other solutions in terms of overhead, searching complexity, security, and compatibility.

- *Chapter 6*: This chapter proposes a novel Dual-Chain-based LoRa system providing global cross-validated security. Behaviors, including the incentive mechanism, and new flow control protocol, can be secured by the tamper-resistance of Blockchains. Being part of the proposed incentive mechanism, the new self-driven flow control scales both the Dual-Chain system and the LoRa network. We also provide analysis and simulations showing that our system can optimize the utilization of coverage while improving the Blockchain scalability and flexibility with the new incentive mechanism.

- *Chapter 7*: The research and contributions of this thesis and are summarized in this chapter, with the discussion of the future works ends the thesis.

Chapter 2

Literature Survey

A comprehensive literature survey is given in this chapter by summarizing recent research breakthroughs regarding the four aspects identified in Chapter 1. They are, respectively, analytical models for PoX consensus algorithms, scale-out solutions, fine-grained access control, and Blockchain-based LoRa networks as a feasible case study.

2.1 Analytical Models for PoX Consensus Algorithms

There have been several studies proposing analytical models to evaluate the consensus engine of Nakamoto protocol, focusing on PoW from the beginning. Garay et al. [68] proposed a model with negligible network delay and constant total mining power for PoW. Miller and LaViola proposed an analytical model for PoW in terms of the faulty tolerance within a reliant synchronization network [131]. Gervais and et al. [70] proposed a specific security model regarding the adversarial strategies (selfish mining) considered in [165]. Also, in [146], a security analysis of PoW based on a partially synchronous network was proposed in terms of both the consistency and network partition. On top of that, Zhang and Preneel [208] thoroughly discussed the security issue of PoW, which mainly focuses on natural/malicious consistency problems due to the considerable block propagation time. Apart from a few papers claiming the randomized consensus [73] and the PoX schemes [180, 186] from which the concept of PoX-based consensus algorithms originate, there are not as many as papers generalizing the PoW/PoS consensus algorithms. They tend to be a model where only PoW, PoS, or any other variants are compared [29, 56, 113].

The above models focus on attacks based on the weakness of incentive schemes due to natural/malicious network partitions caused by the considerable block propagation time, such as the selfish-mining-attack, eclipse-attack and computational double-spending attack in PoW [208], and nothing-at-stake attack and long-range attack in PoS [56]. None of them focuses on the resource distribution, to evaluate how much different settings of the weighted system resource distribution will impact the long-term steady-state, and provides an analytical model to each individual miner for a long-term risk assessment, i.e., the amount of profits can be earned if being a miner to pay the system resource.

It is worth noting that, authors of [208] proposed the pitfalls in existing security models that the unrealistic parameters range may prevent the vulnerabilities from being discovered in the first place and mislead researches into only focusing on a single attack strategy and incentive. This is indeed acceptable, nevertheless, the resource distribution can be analyzed separately from all the other parameters caused by the network delay and non-zero block propagation time; refer to Algo. 4 in [68]. In our model proposed in Chapter 3, we simplify our scenario and focus on the security only impacted by the resource distribution without taking the network delay ([68] considers the same assumption) and any corresponding attack strategies and incentives caused by the delay into account.

2.2 Scale-out Solutions

The scalability issue, which is derived from the discussed Blockchain trilemma, has been considered to consist of two major factors, i.e., throughput and latency. Before digging into the depth, the following introduces how exactly the scalability issue impacts on the Blockchains, and how the scale-out solutions are used to solve the scalability issue by improving the throughput of a large-scale Blockchain system.

The throughput of a Blockchain system, defined as the number of processed

transactions per second of the Blockchain, is far from practical requirements and has become a crucial limitation stopping Blockchain from being widely adopted [183]. For example, Bitcoin can only handle up to approximately 10 transactions per second with its maximum block size of 1MB and average 10 minutes block period [179], which severely hinders the use of Blockchains in the high-frequency trading. To handle a great number of transactions, Blockchain has been considered as a secure base-layer (or a settlement center for cryptocurrencies) where transactions are processed off-chain and then settled in the Blockchain. For example, Lightning network and Raiden network (referring to the state-channel technology) support off-chain payments and broadcast a summary of a batch of off-chain payments to the Blockchain [151, 1]. Plasma (referring to the sidechain technology) builds various applications on the top of Ethereum [150]. These methods, known as the Layer-2 scaling, minimize the interaction with the Blockchain to reduce the latency from the users' perspective but do not improve the throughput of Blockchains [90].

In contrast, the Layer-1 scaling is designed for improving the throughput of Blockchains from the systematic perspective. A Blockchain system can be optimized in the following ways to handle a growing amount of work.

- reducing the communication and computation overhead;
- adding resources to a single node, i.e., vertical scaling;
- adding more nodes to the Blockchain, i.e., horizontal scaling [41].

Reducing overhead: New Blockchain consensus protocols have been developed for high Blockchain throughput by reducing the overhead. For example, every PoW winner (i.e., a miner) is eligible for several blocks rather than a single block in Bitcoin-NG [62] and its variations [25, 94]. The traditional PBFT consensus protocol has been developed and optimized to reduce the communication overhead

and achieve high throughput in large-scale networks [132, 202, 101, 71]. However, $O(n)$ (n is the number of participating miners) is the lower bound that this type of technologies can reduce the overhead at most, as every participating miners have to exchange and store messages during every consensus round regardless of the route of transactions.

Vertical scaling: Bitcoin tried to improve throughput by vertical scaling methods. For example, increasing the number of allowed transactions in a single block and/or reducing the block period can improve the throughput of Bitcoin but consume more resources, e.g., storage, computation, and bandwidth, of Bitcoin nodes [50, 69, 192, 119]. Beyond this, The Greedy Heaviest Observed Subtree (GHOST) [171] is implemented by Ethereum to organize blocks in a tree instead of a chain of blocks and obtain a higher throughput [188]. The GHOST is subsequently extended to the directed acyclic graph (DAG). The DAG is adopted to organize transactions where every transaction contains hash values pointing to existing transactions [152, 48, 23, 170, 172, 109]. The DAG structure allows transactions to be confirmed in parallel and thus improves the network utilization ratio given the resources of a node, which improves the throughput of the entire distributed system. However, the vertical scaling methods cannot infinitely improve the throughput, as a Blockchain system is designed to run in a decentralized and homogeneous network where the security is closely dependent on the consensus across the entire network. The larger-scale the network is, the more bandwidth is needed to achieve the network synchronization, while the bandwidth is the resource that cannot be indefinitely added [50]. This leads to the vertical scaling being compromised to the throughput of resources-limited nodes.

Horizontal scaling: Sharding technology, dividing a whole Blockchain into multiple shards and allowing participating nodes to process and store transactions of a few shards (i.e., only parts of the Blockchain), holds the key to horizontal

scaling, also known as the *scale-out* technology. By taking advantage of the sharding technology that allows partial transactions processing and storage on a single node, the whole Blockchain can achieve a linearly increasing throughput with the growing number of nodes. This is important for the adoption of Blockchains providing high quantity and quality of services to the public in large-scale networks with infinite growth, which has attracted the interest of researches regarding the improvement of the Blockchain scalability.

A number of studies have proposed new sharding mechanisms. Surveys of Blockchain scalability which used to only focus on **Reducing overhead** and **Vertical scaling** have been gradually taking the sharding technology into account. However, none of them was able to focus on sharding and systematically introduce the challenges of sharding, features and restrictions of the existing solutions, and the future trends.

2.3 Fine-grained Access Control

Recently, several access control mechanisms have been proposed for IoT networks based on cloud services [15, 200, 46]. Therein the ABE technology has attracted much attention in cloud computing to provide fine-grained access control [161, 64, 110]. For example, authors of [161] applied ABE over multiple cloud servers; and authors of [64] provided a distributed and scalable structure for an ABE-based IoT network. The ABE allows users to asynchronously decrypt a ciphertext when the users' attributes match the prespecified access rules of the ciphertext with no need for IoT devices staying online all the time [27]. ABE schemes can be divided into Key-Policy Attribute-Based Encryption (KP-ABE) [103] and Ciphertext-Policy Attribute-Based Encryption (CP-ABE). In KP-ABE schemes, a ciphertext is associated with a set of attributes, and users' secret keys are based on an access policy. CP-ABE schemes use access policies to encrypt data, and users' secret keys are gen-

erated over a set of attributes. However, only relying on cloud services may lead to a loss of reliability and traceability of data [20, 134, 204]. Thus Blockchain-based access control becomes increasingly attractive.

In [143, 122], it is shown that Blockchains allow the manager or gateway of each domain network to be part of the consensus process in a Blockchain-based Key Management Cloud. This can ensure the service of key management and distribution for users in a transparent manner. Nevertheless, such access control mechanisms are incapable of providing fine-grained access control in the presence of a device, or even system breakdown, caused by a burst of encryption request messages at the IoT devices. To address this issue, there have been studies on Blockchain-based ABE schemes [185, 156, 154]. For example, in [156], an ABE scheme is employed in a Blockchain-based IoT system to encrypt sensory data into transactions. However, as revealed in [185], few existing Blockchain-based ABE schemes can support a practical mechanism for attribute update and thus can hardly be adopted in dynamic IoT applications. One of those providing a vague discussion about attribute update is [154] (the technological detail is not given), where the smart contract is used to support attribute updates in a Blockchain-based ABE system. The scheme in [154] can potentially allow the data to be directly exposed to members with revoked attributes. This is because the history of smart contract is recorded along with the past blocks, and the revoked members can directly access outdated versions of the ciphertext associated with their outdated ABE private keys. A potential remedy to this issue is to introduce a redactable Blockchain by using CH algorithms [17, 81, 84]. Our system proposed in Chapter 5 achieves a better performance than that of [154] in terms of overhead, searching complexity, and security.

2.4 Blockchain-based LoRa networks: A case study

We take Blockchain-based LoRa networks as a case study in this thesis because of the following two factors.

- LoRa networks allows the data sent from an end-device to be received by multiple LoRa Gateways, i.e., the duplicated data, which is helpful for data redundancy in the context of LPWAN wherein the reliable connection is normally being pruned due to the limited resource. However, the duplicated data uploaded to the Blockchains, on the other hand, compromises the Blockchain throughput.

- Self-motivated deployment is the key strategy of the rollout of LoRa technology. Proper incentive can play an important role in encouraging the private deployment of LoRa and increase coverage. A global validation which can secure the incentive process in a decentralized manner becomes crucial.

Recent studies have investigated the integration between the Blockchain and LoRa technologies [145, 54, 139, 115]. They integrate an existing traditional Blockchain platform (e.g., Ethereum) with a LoRa network, aiming to take advantage of smart contracts for data storage. Ozyilmaz and Yurdakul [145] introduced an Ethereum Blockchain for data storage and access, where either end-devices or gateways are in charge of the block generator. Danish and et al. [54] realized the end-devices incur large overhead with the design in [145], thus introducing a separate Ethereum Blockchain in which multiple agent nodes can provide data storage and access services for either gateways and controllers. Authors of [139, 115] involved an Ethereum Blockchain as a decentralized database providing data storage and access services for all nodes in the network. However, an Ethereum Blockchain or other traditional Blockchain technologies have been revealed that the vulnerable scalability in the

context of Low-Power Wide-Area Network (LPWAN) [163], and the one-device-to-many-gateway property of LoRa networks with ALOHA access even compromises the scalability. Again, an incentive mechanism needed to motivate the LoRa network deployment still lacks a reliable Blockchain-based solution. None of the existing technologies take advantage of both technologies, and are able to deliver a Blockchain-based solution to LoRa Networks where the Blockchain and LoRa technologies can complement each other to address the above issues. By using our new Dual-Chain-based LoRa network, the scalability issue (throughput loss and huge storage) of Blockchains can be relieved without modifying the original LoRa transmission protocol, and the self-motivation to deploy the LoRa networks can be fairly incentivized and secured by Blockchains.

2.5 Chapter Conclusion

A comprehensive literature related to this thesis was reviewed in this chapter, laying a solid foundation for the following chapters. To be specific, this chapter summarized and compared the latest research progress about, 1) the analytical models for PoX consensus algorithms; 2) varieties of solutions to improve the scalability issue and how the scale-out solutions outperform others in terms of throughput; 3) ABE as a solution to provide feasible fine-grained access control in Blockchains; and 4) the integration between Blockchains and LoRa networks. This chapter also identified the gap between existing studies and our research targets, leading to the key research points of this thesis.

Chapter 3

A Unified analytical model for Proof-of-X schemes

As a solution to reach the consensus in the context of a large-scale IoT network, PoX schemes have been considered appropriate and have been learned from various perspectives, including presenting many variations, e.g., Proof-of-Activity (PoA) [24] and Proof-of-Publication (PoP) [180], and the corresponding analytical models. However, there has been to date in lack of a unified analytical model for each participant to evaluate its steady-state profit against the competitors. The selection of mathematical model and metrics which could properly reflect the phenomenon requires more attraction. Under this background, this chapter proposes a Markov model which captures explicitly the weighted resource distribution of PoX schemes in large-scale networks and unifies the analysis of different PoX schemes.

3.1 Introduction

Nakamoto protocol in Bitcoin [135] was proposed to address the Byzantine Generals Problem [104] other than the traditional BFT protocol. The consensus engine of the Nakamoto protocol was first proposed as PoW, and has been extended to other virtual-mining-based variations (e.g., PoS) and subsequently generalized to PoX-based consensus algorithms [180, 186]. PoX schemes take advantage of probabilistic consensus algorithms, and introduce a publicly Verifiable Random Function (p-VRF) with only communication overhead of $O(N)$ (N is the number of miners). Along with the PoX schemes have been widely adopted in a variety of applications (such as Proof-of-Collaboration (PoC) [193] and Proof-of-Distribution (PoD) [106] in IoT systems), as well as the comparison between BFT schemes and PoX schemes

becoming attractive [183, 209, 188], the studies on PoX schemes become increasingly interesting for large-scale networks.

In PoX schemes, the computation resource used in PoW can be replaced using any other publicly-verifiable system resources with customized parameters (e.g., account balance/coinage in PoS and task progress in PoC), as long as the p-VRF can hold. Several recent studies analyzed the PoW, PoS and their variations [131, 146, 73, 208], but none of them was able to evaluate the long-term steady state of the system and the impact of the distribution of system resource on the state. There also lacks a general analytical model for PoX schemes. Such a model would be important to enable each miner to estimate its profit against competitors based on its source. This is important for the traditional mining industry [118] and any public services based on permissionless Blockchains.

In this chapter, we propose a new unified analytical model which is able to quantify the profit of individual miners in any of the popular permissionless PoX-based Blockchains. By applying the model, miners can estimate their profit against their resources under different PoX schemes. The new model captures proposed changes in the system resource distribution of PoX schemes by designing an infinite-dimensional Markov chain. A set of expressions is established to efficiently evaluate the mining probability of a miner, given the amount of system resource owned by the miner. The type and distribution of system resources can be customized in line with system requirements. We develop a new general presentation to unify a variety of system resource distributions in PoX schemes, such as PoW, PoS, and Proof-of-Publication (PoP). Specifically, we characterize probabilistically the system resource owned by a miner. The instantaneous probability with which the miner can mine a block at any instant is generalized to be captured by two new configurable functions respectively accounting for the specific fairness measures of a PoX scheme and the dependence of mining success on the resource distributions in the scheme.

We also design three new performance metrics, namely, *Resource Sensitivity*, *System Convergence*, and *Resource Fairness*, to evaluate the different PoX-based consensus algorithms systematically and consistently. The metrics are quantifiable based on the average mining probabilities that the proposed infinite-dimensional Markov model is able to derive under the new unified measure of system resource distributions.

As revealed by our analysis, in PoX-based consensus algorithms where the monopoly of block generation is prevented and diversity is maintained, miners can maximize the profits with strong double-spending-resistance and controllable cost-risk assessment, thereby contributing to a healthy and sustainable mining ecosystem. Specifically, the system resource has the weakest impact on the average mining probability for each participating miner when a configurable function delivering positive correlation takes effects, which leads to the best *Resource Sensitivity*. Better *System Convergence* can be achieved in PoX schemes with normal-distributed system resource than that with a Pareto-distributed system resource, unless the schemes are designed to restrict the monopoly of block generation. Good fairness or balanced resources play important roles in fast convergence. The proposed fairness function can be implemented in a distributed manner, to improve the fairness between miners and speed up the convergence.

The rest of this chapter is organized as follows. Section 3.2 discusses the preliminary knowledge. In Section 3.3, a Markov analytical model is presented. The considered network setting is also discussed in Section 3.4, followed by the simulation and analysis on different existing PoX-based consensus algorithms in terms of three proposed metrics in Section 3.5. In Section 3.6, conclusions are drawn.

3.2 Preliminary

In this section, the security model considered in Bitcoin is discussed to illustrate the relationship between Bitcoin's security model and our proposed metric, *Resource Sensitivity*, as will be shown in Section 3.3.4.1. The PoX scheme is also recapped as the basis to our proposed model.

3.2.1 Bitcoin's Security Model

In Bitcoin's model [135], security is measured by the probability δ with which an attacker can catch up with the loyal miners to dominate the block generation. δ is considered to be subject to the Poisson Distribution, as given by

$$\delta = 1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!}(1 - (\frac{q}{p})^{z-k}),$$

where p denotes the probability of a loyal node being the current block generator and q denotes the probability of a malicious node being the current block generator. $p > q$, $\lambda = z\frac{q}{p}$, and $k \leq z$, where z denotes how much the malicious node falls behind the loyal miners in terms of block height.

3.2.2 PoX-based Consensus Algorithms

PoW is generalized to the PoX scheme. The PoX scheme describes a system where a unique miner is elected to generate a new block based on a publicly verifiable system resource ratio. The PoX scheme proposed in [180, 186] can be described as follows,

$$\mathcal{P}_{i,\gamma}^{win} = \mathbb{F}\left(\frac{\omega_i}{\sum \omega}, \gamma_i\right), \quad \gamma_i \to \omega_i, \tag{3.1}$$

where $\mathcal{P}_i^{win}$ is the probability of Node-i being elected as the generator of the block, and ω_i is the amount of system resource owned by Node-i. $\sum \omega$ is the total amount

of system resources across the network, e.g., computation power, token balance, etc. $\frac{\omega_i}{\sum \omega}$ denotes the ratio of system resource owned by Node-i over the total amount of resource in the network.

On the other hand, γ_i and $\mathbb{F}(\cdot)$ can be customized to meet different requirements. γ_i is used to adjust how much impact is ω_i having on $\mathcal{P}_i^{win}$ (denoted as $\gamma_i \rightarrow \omega_i$). For example, a hybrid of PoW and PoP [106] defines γ to be the number of packets that a particular miner has distributed and forwarded, which can accordingly decrease the difficulty to win the puzzle-solving race, i.e., increasing the value of $\frac{\omega_i}{\sum \omega}$. Function $\mathbb{F}(\cdot)$ denotes a p-VRF applied during the consensus process to randomly select the block generator, subject to the distribution of $\sum \omega$, as given in (3.1).

In order to select the block generator in a large-scale network with an unknown network size in practice, a PoX-based algorithm can be any consensus algorithm subject to the Longest Chain Rule [186] that leverages a p-VRF based on any verifiable system resource. We consider two different types of system resources. They are 1) system resources which are independent to its transmission bandwidth (typically, this type implies that the the considered system resource ratio in (3.1) does not take the transmission bandwidth into account, and this type of resource can be considered independently without network connection); and 2) network resource (typically, the transmission bandwidth corresponds to the network performance). Such algorithm includes, but is not limited to, the consensus algorithms listed in Table 3.1. For example, authors of [106] considered ω_i as the total amount of system resource that a specific miner i owns, part of which is γ_i in the form of the number of distributed packets. $\mathbb{F}(\cdot)$ can be a height-oriented factor in Proof-of-Stake-Velocity (PoS-Velocity) [3], e.g., a function with variables ω_i and $\gamma_i \rightarrow h$ (indicating $\mathcal{P}_i^{win}$ is height-oriented with positive correlation). In other words, the longer it has been since the last time a miner was elected as the block generator, the more likely the miner is elected as the block generator in the current round.

Table 3.1 : Expressions of (non-)Fairness-oriented PoX schemes

Non-Fairness-oriented, let $\Upsilon(\omega_i) = \omega_i$				
The PoX schemes		**Types of the System Resource**	**Distribution of ω_i** [†]	$f_{i,h}$
Proof-of-Work [135]		Computation power	PD[⋆]	$\alpha\omega_i$
Proof-of-Activity (PoA) [24]		Online duration		
Proof-of-Publication (PoP) [180]	Proof-of-Memory [82]	Memory	ND[‡]	$\alpha\omega_i$
	Proof-of-Storage [100]	Disk Storage		
	Proof-of-Distribution [106]	Packets Forwarding		
Proof-of-Stake (Coinage, Strong restriction) [96, 193]		Account Coinage	PD	$\alpha\omega_i \min\{h-1, H\}$
Proof-of-Stake (Coinage, Weak restriction)				$\alpha\omega_i \min\{h, H\}$

Fairness-oriented, let $\Upsilon(\omega_i) = \zeta(\omega_i)$, where $\zeta(\omega_i)$ can be defined to be partitioned			
The PoX schemes	**Types of the System Resource**	**Distribution of ω_i** [†]	$f_{i,h}$
Proof-of-Stake-Velocity [3]	Account Coinage	PD	$\alpha\zeta_i(\omega_i)\mu(\min\{h, H\})$

[†] The type of distribution a set of system resource expected to follow is dependent to the considered PoX scheme shown in the first column (see the detail in Section 3.5.1).

[⋆] Pareto distribution. It describes an 80/20-rule-based wealth inequality (see the detail in Section 3.5.1).

[‡] Normal distribution.

ω_i denotes the amount of system resource owned by Node-i

$f_{i,h}$ denotes the per-slot mining probability of Node-i; see more details in Section 3.3.3.2

The following model is proposed to generalize (3.1) based on an infinite-dimensional Markov chain. Under the premise of the above-mentioned existing studies failing to evaluate PoX schemes in terms of steady states, the propose model, as far as we know, is the first one that can be used to describe the PoX-based consensus algorithms in a long-term stable system, in order to evaluate the distribution of the mining winners, and predict the cost-benefit ratio.

3.3 New Infinite-Dimensional Markov Chain Model for PoX Schemes

In this section, we first provide an overview of the proposed model along with its network settings followed by the detail of the model. The proposed metrics, *Resource Sensitivity*, *System Convergence*, and *Resource Fairness* will be elaborated.

3.3.1 Overview

For illustration convenience, we consider large-scale synchronous Blockchain networks with reference to the settings of [68] (that at most one miner can successfully mine a block within a time slot). We further consider a sparse Blockchain system in which the value of $\frac{t}{T}$ is sufficiently small and negligible with t denoting the block propagation delay and T denoting the block period. Also, we consider attack strategies which are resource-oriented, where attackers mainly leverage the double spending attacks and selfish mining to maliciously rollback the history based on their current dominated system resource ratio.

The analytical model is designed for analyzing the long-term steady-state (i.e., the probability of each state can be predicted as the system becomes stable) of PoX-based consensus algorithms. This is achieved by considering the resource distribution that is weighed by a Degree function and a Fairness function (which will be defined in Section 3.3.3.2). The analytical model features an infinite-dimensional

Markov chain to investigate the long-term steady state. We also propose three metrics, namely, *Resource Sensitivity*, *System Convergence*, and *Resource Fairness* (which will be defined in Section 3.3.4), for evaluation and simulation. To be specific, a generalized form of $\mathcal{P}_{i,\gamma}^{win}$ (the probability of Node-i elected to be the block generator in a particular round) can be obtained, i.e., $\mathcal{P}_i$. By using $\mathcal{P}_i$, we can evaluate the PoX-based consensus algorithms in terms of the proposed metrics - *Resource Sensitivity*, *System Convergence*, and *Resource Fairness*.

3.3.2 System Model - Small-slotted mechanism

In this section, we describe the system model. The notations used are listed in Table 3.2.

Our proposed analytical model starts with a small-slotted system, where the period of any miner mining a block is denoted as a "round", while a "miner" denotes any node participating in the race to win for the block generator of each round. Each round refers to a block height number and is divided into many small time slots. The number of time slots contained in a round depends on the expected block period, i.e., T. Each of the slots lasts a constant short time. The gap between two consecutive slots can be reduced to satisfy the assumption referred to [68]. This assumption is reasonable as we can make the slots arbitrarily small; see Fig. 3.1.

Each miner can potentially generate a new block on block height n, based on the amount of its system resources, as can be done by evaluating (3.1). In some PoX schemes, such as coinage-based PoS and PoS-Velocity, (3.1) can be affected by the awaiting gap h of each miner, with an upper bound H. The awaiting gap is the gap between the considered miner being the elected generator from the last round to present. There exist the following three possible scenarios for a miner within a slot.

Table 3.2 : Parameters of the analytical model

Symbols	**Description**
h	Awaiting gap that is miner-specific, the gap since the last round a designated miner being the winner until it wins again.*
$\Phi(\cdot)$	A Degree function measuring the impact of h on $f_{i,h}$
$\Upsilon(\cdot)$	A Fairness function defining whether the monopoly of system resource can be avoided
ω_i	The amount of system resource owned by Node-i
N	The number of miners among the entire network
H	The upper bound of h
α	A constant network parameter, normalizing the mining probability $f_{i,h}$ in terms of the size of a time slot
$\Pr(x\|y)$	The transition probability from awaiting gap y to awaiting gap x
R	The mining probability of the entire network per slot
$f_{i,h}$	The mining probability of Node-i per slot at awaiting gap-h
$\pi(h)$	The steady probability of a miner at awaiting gap-h in an arbitrary slot
$\mathcal{T}_i$	The average number of awaiting gap for Node-i being the winner since its last winning
$\mathcal{P}_i$	A generalized form of $\mathcal{P}^{win}_{i,\gamma}$ in (3.1) based on the proposed model
t	The block propagation time
T	The expected value of block period (round)

* This can be any level of grain. For example, it can be block-height-oriented (in terms of the block height), as shown in Section 3.5 or time-oriented [193]. Alternatively, it can be a customized level of grain can replace the block height or time to meet specific requirements.

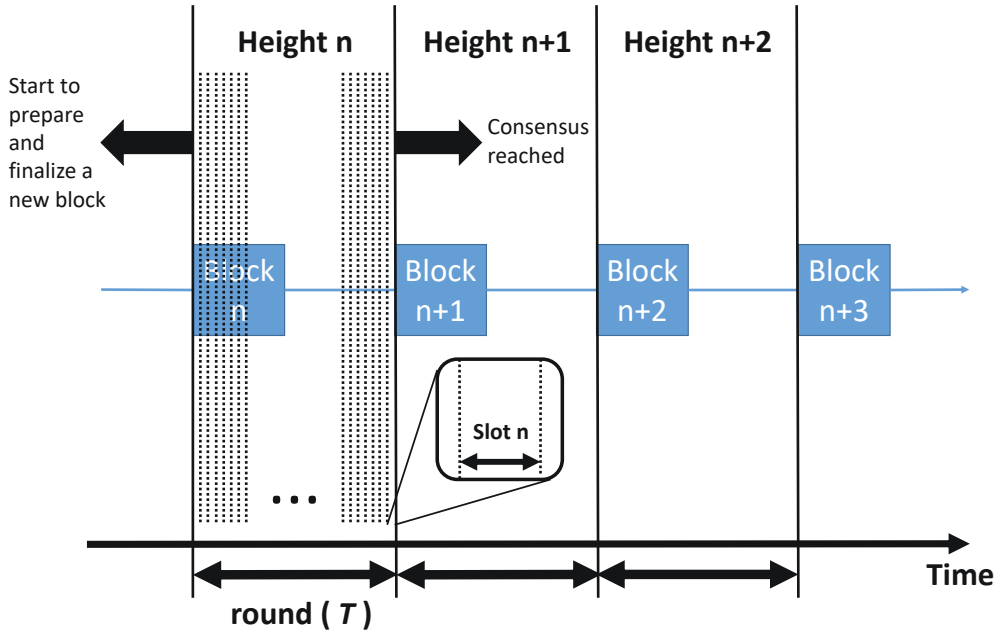


Figure 3.1 : The small-slotted mechanism divides a round into multiple slots. The number of slots contained in a round is subject to the expected value of the block period T.

1. *Scenario 1:* None of the miners mines a valid block in the network.

2. *Scenario 2:* This miner does not mine its own block but accepts a block mined by another miner;

3. *Scenario 3:* This miner mines a block, and the block is immediately accepted by other miners at the beginning of next slot, prior to the mining for the next round;

3.3.3 The Proposed Analytical Model

To clarify *Scenarios 1-3* in Section 3.3.2, we present an infinite-dimensional Markov chain. For simplicity, $\Pr(\cdot)$ denotes the simple form of $\Pr(i,\cdot)$ for Node-i; $\pi(\cdot)$ denotes the simple form of $\pi_i(\cdot)$ for Node-i.

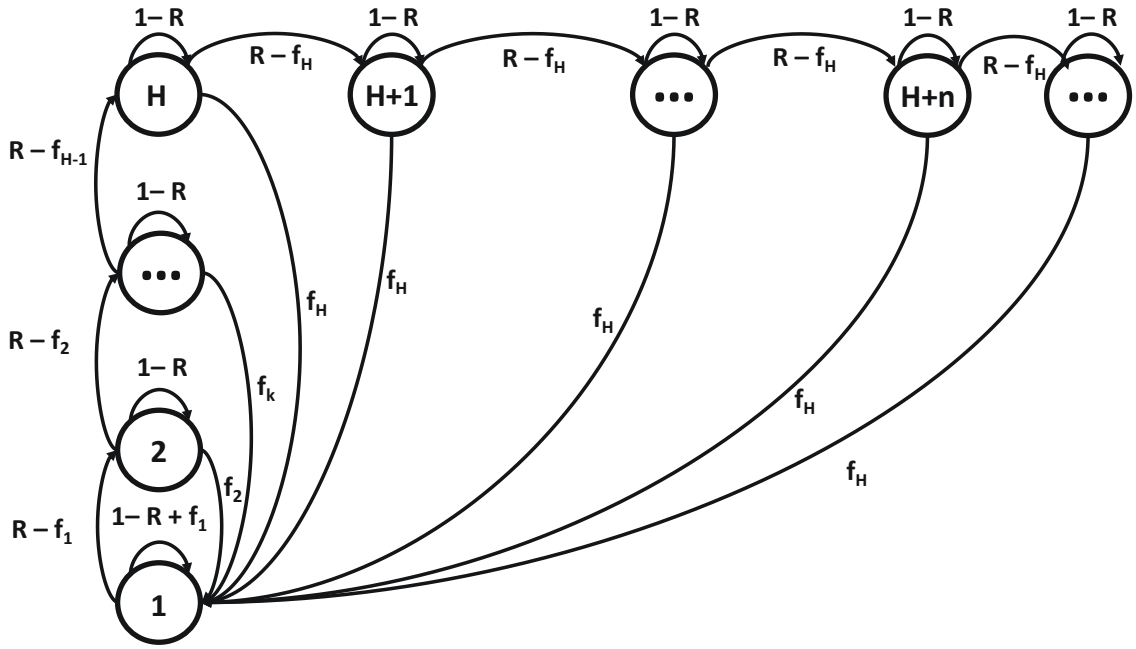


Figure 3.2 : The state transition of the proposed infinite-dimensional Markov chain at Node-i. $f_{i,h}$ is denoted as f_h for simplicity.

3.3.3.1 The infinite-dimensional Markov chain

Let $\Pr(x|y)$ denote the transition probability of an individual node/miner from awaiting gap y to the awaiting gap x at two consecutive slots. The transition probability at Node-i can be given by

$$\Pr(h|h) = 1 - R; \tag{3.2a}$$

$$\Pr(h+1|h) = R - \Pr(1|h); \tag{3.2b}$$

$$\Pr(1|h) = \begin{cases} f_{i,h}, & \text{if} \quad 1 < h \leq H; \\ f_{i,H}, & \text{if} \quad h > H; \\ 1 - R + f_{i,h}, & \text{if} \quad h = 1; \\ 0, & \text{otherwise.} \end{cases} \tag{3.2c}$$

In (3.2), R denotes the mining probability of the whole network at a slot. We consider R is consistent over time, as it must take some k slots for miners to generate a new block for a specific round, while k depending on a constant T can also be considered to be constant in the long term. Recall that h is miner-specific (and is the simple form of h_i). h is not the actual height of the chain, but the awaiting gap which can be block-height-oriented (see Table 3.2), between the previous round where a miner being the block generator and the current round where the same miner being selected again.

In (3.2), a miner running at an awaiting gap h within the considered slot behaves either in the following way.

- Eq. (3.2a) refers to *Scenario 1*. It provides the transition probability that no new block is mined in the network during this time slot. Thus, the miner is still at the awaiting gap h in the next slot.

- Eq. (3.2b) refers to *Scenario 2*. It provides the transition probability that this miner does not generate a new block, and a new block generated by another miner is finalized. Thus its awaiting gap h increases by 1, with a probability of $R - \Pr(1|h)$.

- Eq. (3.2c) refers to *Scenario 3*. It provides the transition probability that the miner is elected as the block generator to finalize a new block. Thus, its awaiting gap h returns to 1, with a probability of $f_{i,h}$ if $1 < h \leq H$. Otherwise, $f_{i,H}$ remains unchanged if $h > H$ with an upper bound H set.

It is apparent that the process above is stateless, in the sense that the probability distribution of any state-$(h+1)$ would only be determined by the current state-h. Moreover, a state can transit forward to $\mathbb{H}$[1] along with the decreased zero-lower-

[1] $\mathbb{H} \geq H$, denoting the gap between the chain tip and state-1, could theoretically be infinity.

bounded steady probability as $h \to \mathbb{H}$. It is thus reasonable to use an infinite dimensional Markov Chain to capture changes of the awaiting gap in our system.

With an increasingly comparable network latency t, the probability of finalizing a valid block per unit time (the slot time) decreases. This leads to a larger average number of slots contained in a round due to the increasing likelihood of forks per slot time, which results in a slower block period; see (3.2). The deviation (decreased probability of the generation of a valid block per slot time) implicitly captures the impact of network delay in practical asynchronous networks on the overall mining process and block miming at block-n. Intuitively, the deviation depends on the value of $\frac{t}{T}$. The probability of finalizing a valid block per slot time incur deviation away from the estimated one derived from our model as $\frac{t}{T}$ increases. In such way, the deviation can correspond to the value of $\frac{t}{T}$.

To simplify and satisfy the small-slotted mechanism, we consider a small $\frac{t}{T}$ where t denotes the propagation delay and T denotes the expected block period (more details in Section 3.4.1). Thus, the miners avoid needing to consider forking (i.e., multiple blocks with identical height being found), and have sufficient time to mine a potential unique block at the same block height-n among all received blocks at height-(n-1). With the help of a game-theoretic incentive scheme [135], the miners are willing to be consistent with each other about the finalized block for the current round (the block is broadcast and accepted by all miners), e.g., based on the difficulty defined in Bitcoin [135] or Ethereum [38]. In other words, this situation can be interpreted to a small-slotted mechanism with sufficiently small and negligible $\frac{t}{T}$, i.e., the first generated block can be finalized and accepted by all miners immediately to reach the consensus, and consistency can be satisfied by the end of this round. Thus, the infinite-dimensional Markov Chain satisfies our proposed small-slotted mechanism, hence *Scenarios 1-3* as defined above can hold. As a consequence for the results of our calculation and simulation, a small $\frac{t}{T}$ minimizes the impact of

propagation latency.

3.3.3.2 The per-miner-per-slot mining probability $f_{i,h}$

According to (3.2c), $f_{i,h}$ provides the per-slot mining probability of Node-i at awaiting gap-h. $f_{i,h}$ depends on other nodes. It reflects the system resource distribution of a specific node, Node-i, at the state with the awaiting gap h. We propose the following expression for $f_{i,h}$ to unify it for a range of popular PoX schemes:

$$f_{i,h} = \alpha \Upsilon(\omega_i)\Phi(h, H) \leq R. \tag{3.3}$$

Here, $\Upsilon(\cdot)$ and $\Phi(\cdot)$ are defined as the Fairness function and Degree function, respectively.

- *Fairness function* indicates whether the weighted resource distribution is Fairness-oriented, and how *Resource Fairness* affects the weighted resource distribution if any.
- *Degree function* can be customized in terms of the awaiting gap h, along with an upper bound H. It measures how much the awaiting gap impacts on the resource distributed to each miner, in turn, the probability that a miner is elected to be the block generator.

For example, $\Upsilon(\omega_i) = \omega_i$ indicates an inactive Fairness function, while $\Upsilon(\cdot)$ can be a partition function if $f_{i,h}$ is Fairness-oriented (so that the impact from a high ω_i to $f_{i,h}$ can be restrictive). α is a network-level parameter normalizing the probability and accounting for the size of a time slot. $\Phi(\cdot)$ can be either equal to 1 (an inactive Degree function), or customized depending on the awaiting gap h and an upper bound H. For example, $\Phi(\cdot) = \min\{h, H\}$ (a positive Degree function that delivers a positive

correlation of $f_{i,h}$ and h), outputting the minimum value between the current h and H, is used in the existing PoS consensus algorithm [96]. The expressions for $f_{i,h}$ under currently popular PoX schemes are shown in Table 3.1.

3.3.3.3 Steady-state probability

In this section, the steady-state probability of a miner at awaiting gap-h at an arbitrary slot is evaluated. The steady-state probability, denoted by $\pi(h)$, can be calculated in three cases, i.e., $h = 1$, $1 < h \leq H$ and $h > H$, subject to the rule of "steady state", $\pi(h) = \sum_i \Pr(h'|h)\pi(h), \forall h : \sum \pi(h) = 1$. We consider that any $f_{i,h}$ with awaiting gap $\forall h > H$, equals to $f_{i,H}$. $\pi(h)$ can be derived based on (3.2) as follows.

In the case of $1 < h \leq H$:

The steady-state probability $\pi(h)$ can be given by

$$\begin{aligned} \pi(h) &= \Pr(h|h-1)\pi(h-1) + \Pr(h|h)\pi(h) \\ &= (R - f_{i,h-1})\pi(h-1) + (1-R)\pi(h), \quad 1 < h \leq H, \end{aligned} \tag{3.4}$$

which is derived from (3.2). In particular, $\pi(h)$ is equal to the sum of the probabilities, 1) that *Scenario 2* happens given that the considered node is on the awaiting gap $h-1$, i.e., $\Pr(h|h-1)\pi(h-1)$; and that 2) *Scenario 1* happens given that the considered node is on the awaiting gap h, i.e., $\Pr(h|h)\pi(h)$.

By rearranging (3.4), $\pi(h)$ can be rewritten as

$$\pi(h) = \frac{R - f_{i,h-1}}{R}\pi(h-1) \tag{3.5a}$$

$$= \pi(1) \prod_{h'=1}^{h-1} \frac{R - f_{i,h'}}{R}, \quad 1 < h \leq H, \tag{3.5b}$$

where (3.5b) is obtained by recursively substituting $\pi(h-1)$ with $\pi(h-1)$, $\pi(h-2)$, $\cdots$, $\pi(2)$ into the right-hand side of (3.5a).

In the case of $h > H$:

The steady-state probability $\pi(h)$ can be given by

$$\begin{aligned}\pi(h) &= \Pr(H+1|H)\pi(h-1) + \Pr(h|h)\pi(h) \\ &= (R - f_{i,h})\pi(h-1) + (1-R)\pi(h), \quad h > H, \end{aligned} \tag{3.6}$$

which is also derived from (3.2). In particular, $\pi(h)$ is equal to the sum of the probabilities that, 1) *Scenario 1* happens given that the considered node is on the awaiting gap h, i.e., $\Pr(h|h)\pi(h)$; and that 2) *Scenario 2* happens given that the considered node is on the awaiting gap $h-1$, i.e., $\Pr(H+1|H)\pi(h-1)$.

Note that in our proposed model, there exist the states with $h > H$ in which the probability of *Scenario 2* is unchanged (i.e., $\Pr(H+1|H)$). The probability that the miner of interest has an awaiting gap H can be interpreted as an accumulation of all $\pi(h)$ with $h > H$, i.e., $\lim_{h\to\infty} \sum_{x=H+1}^{h} \pi(x)$.

By rearranging (3.6), $\pi(h)$ can be rewritten as

$$\pi(h) = \frac{R - f_{i,h}}{R}\pi(h-1) = \pi(H)\left(\frac{R - f_{i,H}}{R}\right)^{h-H} \tag{3.7a}$$

$$= \pi(1)\left(\frac{R - f_{i,H}}{R}\right)^{h-H} \prod_{h'=1}^{H-1} \frac{R - f_{i,h'}}{R}, \quad h > H, \tag{3.7b}$$

where (3.7a) can be converted $\pi(h)$ to multiple of $\pi(H)$, and (3.7b) is obtained by substituting $\pi(H)$ into (3.5b).

In the case of $h = 1$:

As $\lim_{h\to\infty}\sum_{x=1}^{h}\pi(x)=1$, we can add up $\pi(h)$ in all the three cases, i.e., $h=1$, $1<h\leq H$, and $h>H$, as given by

$$\begin{aligned}1&=\pi(1)+\sum_{x=2}^{H}\pi(x)+\lim_{h\to\infty}\sum_{x=H+1}^{h}\pi(x)\\&=\pi(1)\left[1+\sum_{h=2}^{H}\prod_{h'=1}^{h-1}\frac{R-f_{i,h'}}{R}\right)+\frac{R-f_{i,H}}{f_{i,H}}\prod_{h'=1}^{H-1}\frac{R-f_{i,h'}}{R}\Bigg],\end{aligned}\tag{3.8}$$

wherein,

$$\begin{aligned}\lim_{h\to\infty}\sum_{x=H+1}^{h}\pi(x)&=\lim_{h\to\infty}\sum_{x=H+1}^{h}\pi(H)\left(\frac{R-f_{i,H}}{R}\right)^{x-H}\\&=\pi(H)\left(\frac{R-f_{i,H}}{R}\right)\frac{1}{1-\frac{R-f_{i,H}}{R}}\\&=\pi(H)\frac{R-f_{i,H}}{f_{i,H}},\end{aligned}$$

which is derived from (3.7a). Also from (3.8), the steady probability $\pi(1)$ can be obtained as

$$\pi(1)=\frac{1}{1+\sum_{h=2}^{H}\left(\prod_{h'=1}^{h-1}\frac{R-f_{i,h'}}{R}\right)+\frac{R-f_{i,H}}{f_{i,H}}\prod_{h'=1}^{H-1}\frac{R-f_{i,h'}}{R}},\qquad h=1.\tag{3.9}$$

Consequently, $\pi(h)$ can be derived with (3.5), (3.7) and (3.9) for any awaiting gap h in the cases with $1<h\leq H$, $h>H$, and $h=1$.

3.3.3.4 Relation between the total per-slot mining probability R and the per-slot mining probability of an individual node $f_{i,h}$

Recall the mining probability of the entire network per slot, R, as given in (3.2). It can be interpreted as the steady mining rate of the whole network. R is given by

$$R=\lim_{h\to\infty}\sum_{i=1}^{N}\sum_{x=1}^{h}f_{i,x}\pi_i(x),\tag{3.10}$$

where $\pi_i(x)$ denotes the steady-state probability of Node-i with the awaiting gap x, and $\pi_i(x)$ can be obtained by substituting (3.3) into (3.5) and (3.7).

By using (3.5) and (3.7), the total per-slot mining probability R can be expanded to the following form:

$$R = \sum_{i=1}^{N} f_{i,1}\pi_i(1) \tag{3.11a}$$

$$+ \sum_{i=1}^{N}\sum_{h=2}^{H} f_{i,h}\pi_i(1) \quad \prod_{h'=1}^{h-1} \frac{R - f_{i,h'}}{R} \Bigg) \tag{3.11b}$$

$$+ \sum_{i=1}^{N} f_{i,h}\pi_i(1)\frac{R - f_{i,H}}{f_{i,H}} \quad \prod_{h'=1}^{H-1} \frac{R - f_{i,h'}}{R} \Bigg). \tag{3.11c}$$

Therefore, R can be calculated with (3.11) by summing up the steady mining rate for any awaiting gap $h \in [1, \infty]$ among all nodes $i \in [1, N]$. Note that $\pi_i(1)$ is the corresponding steady-state for Node-i with awaiting gap-1; therefore, $\pi_i(1) = \pi(1)$. $\pi(1)$ is given in (3.9).

3.3.3.5 Generalization of $\mathcal{P}_{i,\gamma}^{win}$

Recall that $\mathcal{P}_{i,\gamma}^{win}$ is the probability of Node-i being elected as the block generator. We derive $\mathcal{P}_i$ which generalizes $\mathcal{P}_{i,\gamma}^{win}$ in terms of $f_{i,h}$ for PoX schemes. By using such generalized form, γ is abstracted into the configurable functions (Fairness function and/or Degree function) of $f_{i,h}$ for any Node-i. As such, any miners can obtain the probability being elected as the block generator by calculating $f_{i,h}$. We define $\mathcal{T}_i$ as the block generation rate of Node-i. It is measured by the average number of awaiting gaps required for the miner to be elected again. For simplicity of notation, we let $f_{i,h} = \alpha\Upsilon(\omega_i)\Phi(h, H) = \alpha_i\Phi(h, H)$, where $\alpha_i = \alpha\Upsilon(\omega_i)$ and α is a network parameter normalizing the probability. This operation is reasonable as the *Degree function* $\Phi(\cdot)$ can equal to 1 and be ignored for some PoX schemes (e.g., PoW), while we can realize α_i is (non-)Fairness-oriented by observing whether $\Upsilon(\omega_i) = \omega_i$

holds. Then, $\mathcal{T}_i$ can be given by

$$\begin{aligned}\mathcal{T}_i &= \lim_{h\to\infty}\sum_{x=1}^{h} x\frac{\pi(x)}{\pi(1)}f_{i,x} \times \lim_{k\to\infty}\sum_{x=0}^{k}(1-R)^x \\ &= \left(\sum_{x=1}^{H} x\frac{\pi(x)}{\pi(1)}f_{i,x} + f_{i,H}\lim_{h\to\infty}\sum_{x=H+1}^{h} x\frac{\pi(x)}{\pi(1)}\right)\times\frac{1}{R} \\ &= \frac{\sum_{x=1}^{H} x\frac{\pi(x)}{\pi(1)}f_{i,x}}{R} + \frac{f_{i,H}\left(\prod_{h'=1}^{H-1}\frac{R-f_{i,h'}}{R}\right)}{R}\times\lim_{h\to\infty}\left[\sum_{x=H+1}^{h} x\left(\frac{R-f_{i,H}}{R}\right)^{x-H}\right].\end{aligned} \tag{3.12}$$

As a result, $\mathcal{P}_i$ can be given by

$$\mathcal{P}_i = \frac{1}{\mathcal{T}_i}. \tag{3.13}$$

Note that $\lim_{k\to\infty}\sum_{x=0}^{k}(1-R)^x = 1/R$, and it indicates that no new block has been finalized in the last k slots of the current round. $\frac{\pi(x)}{\pi(1)}f_{i,x}$ is the probability that the awaiting gap of Node-i starts at the height of x. Here, $\pi(x)$ is divided by $\pi(1)$ to eliminate the effect of the initial state (i.e., $h = 1$).

As a result, $\mathcal{T}_i$ can be calculated based on (3.5) to (3.11), and the given $\{\alpha_i, \Upsilon(\cdot), \Phi(\cdot)\}$. Since $\mathcal{T}_i$ is the generation rate of Node-i (i.e., how many awaiting gaps on average a miner needs to wait until it can be elected as the block generator since the last time it was elected), the generalized form of $\mathcal{P}_{i,\gamma}^{win}$ in (3.1), $\mathcal{P}_i = 1/\mathcal{T}_i$.

3.3.4 Proposed Evaluation Metrics

Based on the proposed analytical model and the resulting $\mathcal{P}_i$ in (3.12) and (3.13), we propose three important metrics to evaluate PoX schemes, i.e., *Resource Sensitivity*, *System Convergence*, and *Resource Fairness*. Currently popular PoX-based consensus algorithms, as summarized in Table 3.1, can all be evaluated by using the proposed metrics.

3.3.4.1 Resource sensitivity

The proposed *Resource Sensitivity* evaluates *the correlation between the system resource ratio* $\frac{\omega_i}{\sum \omega_i}$, *and the average probability of Node-i being the elected as the block generator,* $\mathcal{P}_i$.

For any PoX scheme, we have $\mathcal{P}_i = f(z)$, where $z = \omega_i / \sum \omega_i$. The gradient g of any point and the corresponding area $E(g)$ bounded by $f(z)$ can be defined as

$$g = \frac{\mathrm{d}(f(z))}{\mathrm{d}z};$$

$$E(z) = \int_0^{50\%} f(z)\mathrm{d}z, \quad \forall g \geq 0.$$

We define *Zero-Resource-Sensitivity* if $g = 1$; *Positive-Resource-Sensitivity* if $g > 1$; and *Negative-Resource-Sensitivity* if $0 \geq g < 1$. *Zero-Resource-Sensitivity* indicates the mining probability $\mathcal{P}_i$ is proportional to the resource ratio in a 1:1 ratio. The 50% is the resource ratio with which this node can launch the double-spending attack in a *Zero-Resource-sensitive* context (1:1 ratio). A positive sensitivity leads to a larger impact to $\mathcal{P}_i$ by the resource ratio, while a negative one leads to a smaller impact to $\mathcal{P}_i$. We also assert the relationship between *Resource Sensitivity* and the security of a PoX scheme, i.e., the smaller $E(z)$ is, the less Resource-sensitive it can achieve, thus a more secure PoX scheme that has better performance on *Resource sensitivity*.

By referring to the security model proposed in Bitcoin's whitepaper [135], security is specified to be the probability that an attacker could catch up with loyal miners in some consecutive rounds. It is closely dependent on the probability that an attacker potentially wins the puzzle race and generates a malicious block, as captured by our proposed $\mathcal{P}_i$. For example, $\mathcal{P}_i$ of traditional PoX schemes, e.g., PoW and PoS, leads to a *Zero-Resource-Sensitivity* with $f_{i,h} = \alpha\omega_i$, $\Upsilon(\omega_i) = \omega_i$, and $\Phi(\cdot) = 1$ (which is the identity line illustrated as the dark blue solid line in Fig. 3.6).

This consequently indicates that $\mathcal{P}_i$ increases along with the system resource in a 1:1 ratio, i.e., $E(z) = \mathbf{1}$ (i.e., an isosceles right triangle. Here "**1**" represents a normalized area.), hence more Resource-sensitive and less secure than those with $E(z) < \mathbf{1}$.

Resource Sensitivity is complementary to the Bitcoin's security model, in terms of the correlation between the system resource ratio and the average probability of any node being the block generator. Such definition of *Resource Sensitivity* based on the resource distribution is reasonable, as influential attack strategies (e.g., selfish mining and double-spending attack) depend on the resource distribution.

3.3.4.2 System convergence

The entire system takes rounds to reach the steady-state, while the steady-state is satisfied if both (3.4) and (3.6) hold. Thus, *System Convergence* is evaluated by *the number of rounds needed to reach steady-state*.

Here, the gap between the theoretical value of $\mathcal{T}_n$ (driven by (3.12)), and the simulation one (obtained by the Monte Carlo-based simulation) is upper bounded by a chosen tolerance (the steady-state is reached if the tolerance is satisfied). The tolerance is chosen as $\sim 3\%$ (see Fig. 3.4). This is because the PoX schemes considered in Table 3.1 have a margin of error of 3% (the y-axis of Fig. 3.4) while the ratio of system resource owned by an arbitrary node $f_{i,h} \simeq 50\%$ (the x-axis of Fig. 3.4). This satisfies the requirement of the fault tolerance (FT) of PoX-based consensus algorithms, $N > 2\mathrm{f}+1$ (N denotes the total number of participating miners; f denotes the number of malicious miners).

A stable $\mathcal{P}_i$ can be useful for each individual miner to estimate the profits more accurately, while an unstable consensus algorithm does not provide such benefits in the absence of a steady-state. As a result, *System Convergence* can be an important metric for rational users who tend to run more controllable PoX schemes. They will

be able to observe how much longer they need to wait until the entire system becomes stable and predictable with high precision, so that a more controllable cost-risk assessment can be conducted. Here, a more controllable cost-risk assessment implies that, the faster a system becomes stable, the earlier users obtain an accurate profit estimation, thus the users can be more thoroughly prepared for all possible financial challenges. Moreover, we prove the model realistic with *System Convergence* in a real-world system based on the simulation in Section 3.5.2.1. The simulation reveals that our model can provide reasonably accurate estimation of the number of rounds in the real world, especially in a well-connected network with low latency.

3.3.4.3 Resource fairness

Resource Fairness [123] is defined as *an indicator that indicates (in the case of* $E(z) < \mathbf{1}$*) whether there exists a threshold* η *with respect to the resource ratio, to the right-hand-side of which* $g \to 0$ *based on the corresponding Fairness function* $\Upsilon(\cdot)$.

The asymptotically zero gradient ($g \to 0$) provides *Resource Fairness* against a wealthy, resourceful node (i.e., Fairness-orientation). Here, a wealthy node has at least 50% resource ratio, with which this node can launch the double-spending attack in a *Zero-Resource-sensitive* context (1:1 ratio).

Resource Fairness is a specific requirement of a PoX-based consensus algorithm. It prevents monopolization of wealthy miners, and incentivizes all miners to participate in the mining process. The miners are expected to voluntarily apply *Fairness function* because of the similar reason how miners remain decentralized among centralized mining pools [55][2]. *Resource Fairness* has an impact on *Resource Sensitivity*

[2]The loss caused by a double-spending attack launched by a centralized mining pool will make the participated miners migrated out. Similarly, miners are expected to voluntarily prevent the monopoly by restricting the wealthy when they are rich enough.

by narrowing down the gap between the lowest and highest $\mathcal{P}_n$ with an unchanged value of H. *Resource Fairness* also affects *System Convergence* by introducing a many-to-one function with respect to h, e.g., a partition function $\zeta(\cdot)$ in Section 3.5. As such, the Fairness function $\Upsilon(\cdot) = \zeta(\cdot)$ can significantly decrease the number of rounds to reach the steady-state; see Fig. 3.7 for further details.

As of the time of writing, the proposed model is the first, among existing studies, to determine the Fairness function and Degree function, thus considering the effects on the correlation between the system resource ratio and the individual mining probability in different settings of these two functions, i.e., *Resource Sensitivity* and *Resource Fairness*. The model is also the first to be able to evaluate the long-term stability by considering the convergence of Markov steady state, i.e., *System Convergence*, which provides a reliable cost-risk assessment.

3.4 Consideration on Network Setting

3.4.1 Sparse Blockchain Networks

Our model, using the similar assumption of [68], i.e., a well-connected network with a small $\frac{t}{T}$, and $\frac{\sum \omega_i^{\text{forked}}}{\omega}$ is also negligible with a constant $\sum \omega$. A small $\frac{t}{T}$ also contributes to a negligible orphan rate, giving attackers no opportunities to exploit any attack strategies on the orphans; refer to [8, Section IX-A].

Forking is purposely prevented when every miner mines on a new block with the same block height, and the assumption of a zero propagation time t is subject to the following reasons.

For the PoW schemes, the de facto probability that a forked block is mined and inserted is also (apart from $\mathcal{P}_{i,\gamma}^{win}$) directly proportional to,

1. the ratio between t and T, i.e., $\frac{t}{T}$;

2. the ratio of computation resource $\sum \omega_i^{\text{forked}}$ working on a block that would be an orphan one, to the total resource ω; i.e., $\frac{\sum \omega_i^{\text{forked}}}{\omega}$ (the *chain quality* proposed in [68, 208]).

The upper bound of time consumption to finalize a new block, δ, becomes unpredictable as $\frac{t}{T} \to 1$. This means the synchronization becomes looser so that the performance and security of PoW deteriorates. Thus, as we consider

1. $\frac{t}{T}$ approaches zero (the throughput is not considered here);

2. rational miners are incentivized to wait until the finalized block of the current round is consistently accepted across the whole network before mining the next block, [3]

$\frac{\sum \omega_i^{\text{forked}}}{\omega}$ can be negligible in our model.

For the PoS and other PoX schemes without the computation resource, punishment mechanisms (such as the Slashers in Ethereum 2.0 [37] or the Verifiable Delay Function (VDF) [32]) are applied to miners who mine multiple blocks on the same height to prevent Nothing-at-stake and long-range attacks. In the case that $\frac{t}{T}$ approaches zero, $\frac{\sum \omega_i^{\text{forked}}}{\omega}$ is also negligible.

3.4.2 Large-scale Blockchain Networks

The proposed model is designed for a large-scale Blockchain network with N potential miners competing for the role of block generator. We define a finite scale of an upper bound of the awaiting gap h, i.e., $N > H$. This is reasonable and usually

[3] In this paper, we consider that the attackers do not behave maliciously at the time when the honor miners are waiting for the consistency while T becomes large. When $\frac{t}{T} \to 0$, $T > \delta$, the partially synchronous network can be thought to be completely synchronous, in turn, satisfies the proposed small-slotted mechanism.

implemented in PoS-Velocity [3]. Similar designs have also been implemented to prevent coins hoarding. This is a critical issue of traditional coinage-based PoS. Attackers hoard the coins on multiple accounts with infinite H to boost the success probability [180]. On the other hand, it is practical to implement an upper bound so that the physical operation can be more affordable due to the worst-case searching complexity of $O(NH)$ for traversing the awaiting gaps of all N miners in the entire network.

3.5 Simulation and Evaluation

In this section, we present the simulation and evaluation, based on *Resource Sensitivity*, *System Convergence*, *Resource Fairness* of our proposed analytical model of the considered PoX schemes listed in Table 3.1.

3.5.1 Framework

The simulation setting is presented in the following.

Hardware setting: A 2017 iMac with 10.13.3 macOS High Sierra, a processor of 2.3GHz Intel Core i5 and 16 GB 2133 MHz DDR4 memory are used.

Software setting: We carry out a Monte Carlo simulation using Python-2.7 to conduct the mining process given N, H, and $\alpha = \frac{1}{2H\sum\omega_i}$, to obtain the simulated value of $\mathcal{P}_i$ for Node-i. The calculated value of $\mathcal{P}_i$ for Node-i based on (13) is obtained by using Matlab-2017. Here, the values of N, H, and α are set based on the hardware performance.

Samples setting: The parameter $f_{i,h}$ for each of the PoX-based consensus algorithms is described in Table 3.1, given the distribution of $\{\alpha_i\}$. Here, a coinage-based PoS with a strong restriction implies that the awaiting gap of an elected generator starts from 0 (zero probability of being elected consecutively), while a weak restriction starts from non-zero.

In this simulation, we use a normal distribution and a Pareto distribution for the resource distribution among the miners. The reason is that these two distributions feature normal-distributed wealth inequality (non-monopoly) and 80/20-rule-based wealth inequality (monopoly), respectively, covering most well-known PoX schemes listed in Table 3.1. Any PoX scheme where miners need to put great efforts, e.g., computational power and token values, in winning the race results in a Pareto Distribution [181], while it is normal-distributed if the system resource becomes less costly to the system resource among the miners, e.g., PoP. As such, we assume that $\{\alpha_i\}$ of PoW, PoS, and PoA follow the Pareto distribution; PoP follows the normal distribution. To be specific, we implement Proof-of-Collaboration (PoC) [193][4], for the simulation of both strongly restricted and weakly restricted coinage-based PoS consensuses.

We implement a typical type of Fairness-oriented PoS-Velocity with a Pareto distribution in the simulation, where the Fairness function $\Upsilon(\omega_i) = \zeta(\omega_i)$ and the Degree function $\Phi(\cdot) = \mu(\min\{\cdot\})$ are used[5]. Here, we use the definition in [180] of PoS-Velocity. That is the linear $\Phi(\cdot)$ is substituted by other forms of Degree functions, e.g., a non-linear function $\mu(\cdot)$; the Fairness function is set to the form of a partition function, e.g., $\zeta(\cdot)$.

[4]This consensus defines two new parameters, $\mathcal{CC}$ and $\mathcal{P}_{PoC}$. The winner of each round of generating the new block will earn $\mathcal{CC}$, while $\mathcal{P}_{PoC}$ is defined as the time since the last $\mathcal{CC}$ changes. On the other hand, $\mathcal{P}_{PoC} \in [\mathcal{L}, \mathcal{R}]$, where $\mathcal{L}$ can be constant during a long-term period and $\mathcal{R} = 3\mathcal{L}$. Also, $\mathcal{P}_{PoC}$ of the winner is set to 0 for the next single round (so that $\mathcal{P}_{PoC}$ starts from 0). Therefore, the PoC consensus can be regarded as a variant of strongly restricted coinage-based PoS and $f_{n,h} = \alpha_n \min\{h-1, H\}$, where $1 \leq h \leq H$, $h = \mathcal{P}_{PoC}$, $h = 1 = \mathcal{L}$, $h - 1 = 0$, $H = \mathcal{R}$. In addition, the PoC consensus can be with a weak restriction if we set $\mathcal{P}_{PoC}$ to $\mathcal{L}$ instead of 0.

[5]An example is that, $\zeta(\cdot)$ can be a partition function where the lower and upper bound are pre-defined to avoid the monopoly and starvation; $\mu(\cdot)$ can be a non-linear function where the gradient g remains flat from the beginning up to a threshold, followed by a sharp increase after the threshold (so that the poor miners can be more likely to win).

3.5.2 Simulation Result

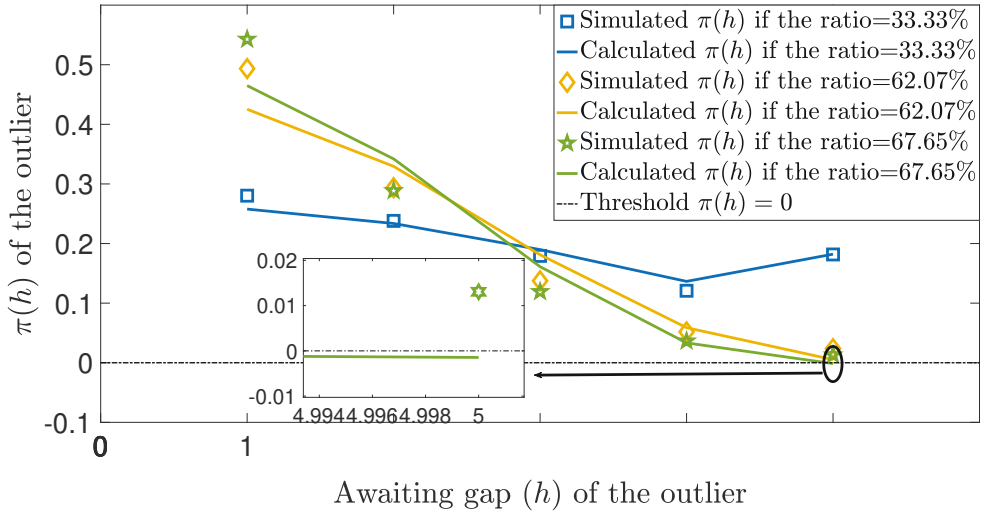
First, the scope of our proposed model in terms of the margin of error is discussed. After that, the proposed metrics, *Resource Sensitivity*, *System Convergence*, and *Resource Fairness* are simulated among the (non-)Fairness-oriented PoX schemes listed in Table 3.1. Finally, we deliver the implicit findings for miners to evaluate PoX schemes in different scenarios based on the proposed model.

3.5.2.1 Accuracy of the proposed model - margin of error

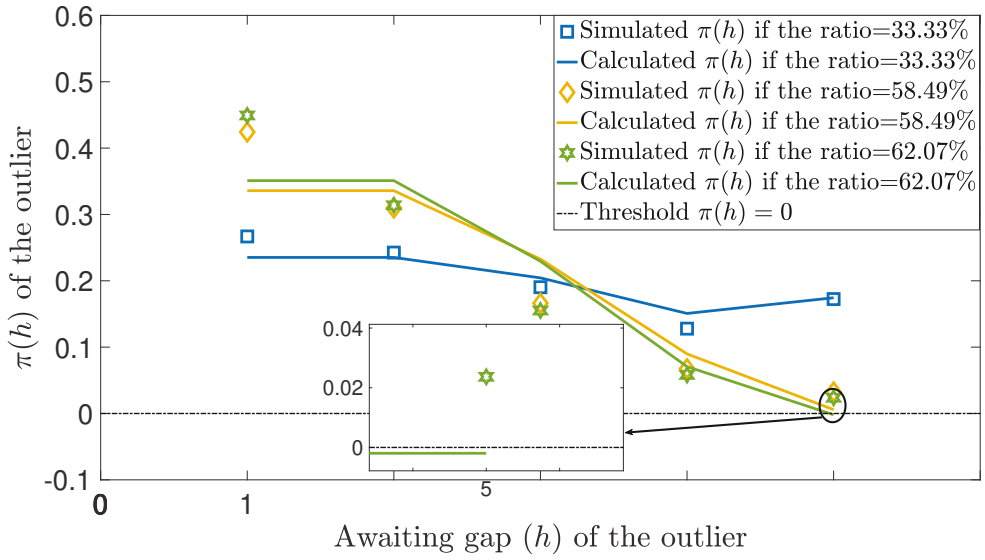
To investigate the accuracy of the estimation derived from our model and possible factors impacting on such accuracy, we consider two types of margin of errors in this section.

- **Standard Error** (S). It is also known as the standard error of the estimate, representing the average distance between the estimated values and observed values. Smaller S implies a better fitted model.
- **Adjusted R-squared** ($ARSQ$) [130]. $ARSQ$ is known as the adjusted coefficient of determination in statistics, representing the ratio of the variance in the dependent variable that is predictable from the independent variable(s) with considering the number of independent variable(s). It is often used to assess how good the estimated model fit the observed values, the closer to 1 the better. Note that, in our simulation $ARSQ$ is a complemented metrics to S as a non-linear model may imply an inaccuracy due to the unexpected over-fitting. A high-$ARSQ$ indicates a good fitting only if S is within the acceptable range. In contrast, we can still reliably approximate the trend with a high-$ARSQ$ when S is slightly higher than the range.

Our analytical model is applicable to Pareto distributions (that is the worst case), where the outlier owns up to 50% of the system resource that is equal to the



(a) $f_{i,h} = \alpha \min\{h, H\}$



(b) $f_{i,h} = \alpha \min\{h - 1, H\}$

Figure 3.3 : $\pi(h)$ of the outlier with a Pareto distributed system resource for coinage-based PoS and Non-Fairness-oriented PoS-Velocity respectively, where $N = 10$, $H = 5$, $\alpha = \frac{1}{2H\sum\omega_i}$. An invalid $\pi(H)$ that is negative appears when $h = H$.

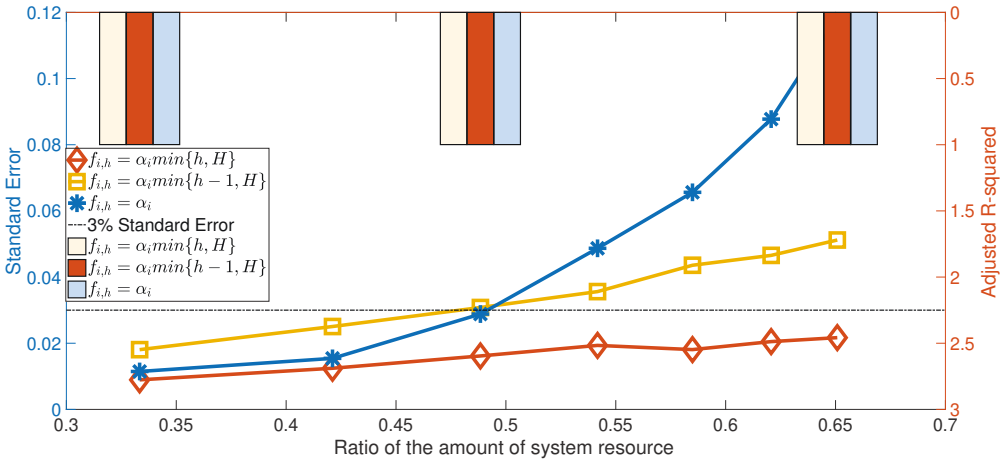


Figure 3.4 : The line plots are with respect to the blue axis on the left-hand side, while the bar plots are with respect to the upside-down red axis on the right-hand side. The margin of error with the growth of the resource ratio owned by an arbitrary miner that is the outlier, where $N = 10$, $H = 5$, $\alpha = \frac{1}{2H\sum\omega_i}$, in terms of both S and $ARSQ$. Here, the system resource is Pareto distributed. Note that the ratio is the system resource ratio that a specific outlier miner owns.

FT of all PoX schemes. An outlier denotes Node-i that owns the majority of the Pareto-distributed system resource. For example, in the following list if $N = 10$,

$$\boldsymbol{\omega} = [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.55],$$

where the n-th element in the list is the ratio of the amount of system resource of Node-i. The node with $\omega = 0.55$ is defined as the outlier.

According to Fig. 3.3, when $f_{i,h} = \alpha_i \min\{h, H\}$ (see Fig. 3.3(a)) and $\alpha_i \min\{h-1, H\}$ (see Fig. 3.3(b)), the invalid negative $\pi(i, H)$ appears at $h = H$, as the ratio of system resource owned by Node-i increases.

Fig. 3.4 shows the correlation of the resource ratio owned by the outlier, with the two types of margin of error between the estimated and simulated values. It shows that $ARSQ$ of all considered $f_{i,h}$ remains closed to 1, which results in a good fitting if S is within the acceptable range. S remains low when the ratio of the amount of system resource is less than 50% for all considered $f_{i,h}$. Also, S increases

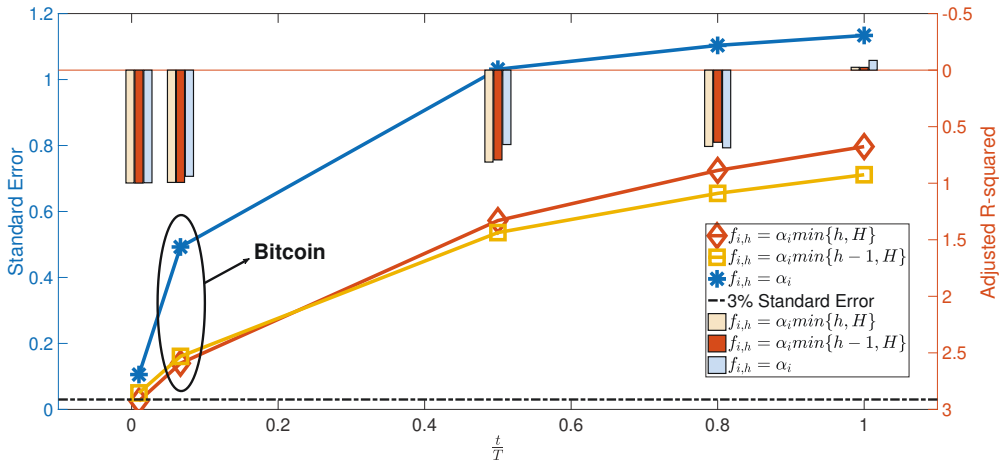


Figure 3.5 : The line plots are with respect to the blue axis on the left-hand side, while the bar plots are with respect to the upside-down red axis on the right-hand side. The margin of error with the growth of $\frac{t}{T}$, where $N = 10$,$H = 5$,$\alpha = \frac{1}{2H\sum\omega_i}$, in terms of both S and $ARSQ$. Here, the system resource is Pareto distributed with the ratio of the amount of system resource owned by a specific outlier miner is 33%.

exponentially as the ratio increases for $f_{i,h} = \alpha_i$ (the blue line). Thus, it can be concluded that, the proposed model suits in the Pareto distribution with an outlier owning up to $\sim 50\%$ resource, but does not suit an accurate prediction for a Pareto-distributed system with an outlier that is too fart apart (greater than $\sim 50\%$), except for algorithms satisfying *Resource Fairness* (referring to the example of PoS-Velocity shown in Table 3.1). In spite of this, the proposed model can still be reliable on approximating the trend. Note that the smallest outlier has satisfied the required FT ($N \geq 2f + 1$), where f is the number of faulty miners. This consequently leads to an acceptable range of S for the accuracy of the proposed model, i.e., 3%.

Fig. 3.5 shows the correlation of $\frac{t}{T}$ with the two types of margin of error between the estimated and simulated values. By investigating what range of $\frac{t}{T}$ the margin of error can be acceptable, we can subsequently determine the upper bound of $\frac{t}{T}$ which can tolerate the possible deviation in (3.2). It shows that the values of $ARSQ$

of all considered $f_{i,h}$ remain closed to 1 when $\frac{t}{T} \leq \frac{40}{600}$ (Bitcoin point, 95% confident interval) [135, 57], decrease smoothly when $\frac{t}{T} \leq 0.8$, and incur a sharp decrease onwards. S of $f_{i,h} = \alpha_i \min\{h, H\}$ and $f_{i,h} = \alpha_i \min\{h-1, H\}$ remain closed to the 3% range when $\frac{t}{T}$ falls around the Bitcoin point, which still results in a reliable trend-approximation. However, S of $f_{i,h} = \alpha_i$ (the blue line) supports the reliable trend-approximation only if $\frac{t}{T}$ stands around the Bitcoin point, and incurs a sharp increase onwards. The same circumstance happens for $f_{i,h} = \alpha_i \min\{h, H\}$ and $f_{i,h} = \alpha_i \min\{h-1, H\}$ if $\frac{t}{T}$ is greater than the Bitcoin point. Thus, it can be concluded that, the proposed model supports a reliable trend-approximation for $\frac{t}{T}$ that is smaller than the Bitcoin point.

Validated by Figs. 3.3 to 3.5, it can be further concluded that

- the model is accurate if the FT of PoX schemes is satisfied with either Pareto-distributed or normal-distributed resource;
- the model can provide a reliable trend-approximation when $\frac{t}{T}$ is sufficiently small (the network latency is comparatively negligible to the block period), which corresponds to the circumstance of a well-connected network with low latency in the real world.

3.5.2.2 Resource sensitivity

We simulate the process ranging from PoW to PoP, and both of the coinage-based PoS with strong and weak restrictions, as shown in Table 3.1. Also, the corresponding calculated values are obtained by calculating (13) under different settings of $f_{i,h}$.

Finding 1: The coinage-based PoS (Strong restriction) [96, 193] has the best performance on Resource Sensitivity among our considered non-Fairness-oriented PoX schemes.

This finding is revealed in Fig. 3.6, where $\mathcal{P}$ is subject to the ratio of the amount of system resource. An identity line regardless of the distribution type is obtained for $f_{i,h} = \alpha_i$ (the dark blue line), i.e., zero-Resource-sensitive. Note that $\mathcal{P}_i$ is collectively generalized as $\mathcal{P}$ among the miners. The light blue curve of $f_{i,h} = \alpha_i \min\{h-1, H\}$ appears to have a better performance on *Resource Sensitivity* than that of $f_{i,h} = \alpha_i \min\{h, H\}$ (the purple line) due to the setting of strong restriction rather than weak restriction. Referring to Section 3.2.1, it can be concluded that the cost for attackers to catch up with the honest miners can be higher with $f_{i,h} = \alpha_i \min\{h-1, H\}$ or $f_{i,h} = \alpha_i \min\{h, H\}$ than only $f_{i,h} = \alpha_i$.

Finding 2: The poor (i.e., the less resourceful miners) can gain more profit with a positive Degree function (that increases the mining probability by multiplying the resource ratio and the awaiting gap) [3, 96, 193]. In contrast, the obtained profit becomes lower for the rich with the increased ratio of resource owned.

Based on Fig. 3.6, it is conceivable that poor miners can obtain a greater gradient g than wealthy miners (positive-Resource-sensitivity), in the case of $f_{i,h} = \alpha_i \min\{h-1, H\}$ (the light blue line) and $\alpha_i \min\{h, H\}$ (the purple line) with a Pareto-distributed resource. There exists a threshold intercepting the identity line, to the left of which the gradient m is greater so that poor miners can obtain a larger $\mathcal{P}$ than they used to deserve with only $f_{i,h} = \alpha_i$ (the dark blue line). Likewise, wealthy miners, i.e., the outliers, can only obtain a smaller g than that of $f_{i,h} = \alpha_i$. This implies a mechanism that taking from the wealthy to help the poor to balance the profits among the whole participated miners.

3.5.2.3 System convergence

We proceed to evaluate *System Convergence* of the considered PoX schemes, where each of them runs for 1,000 tries. In each of the schemes, the system starts from the

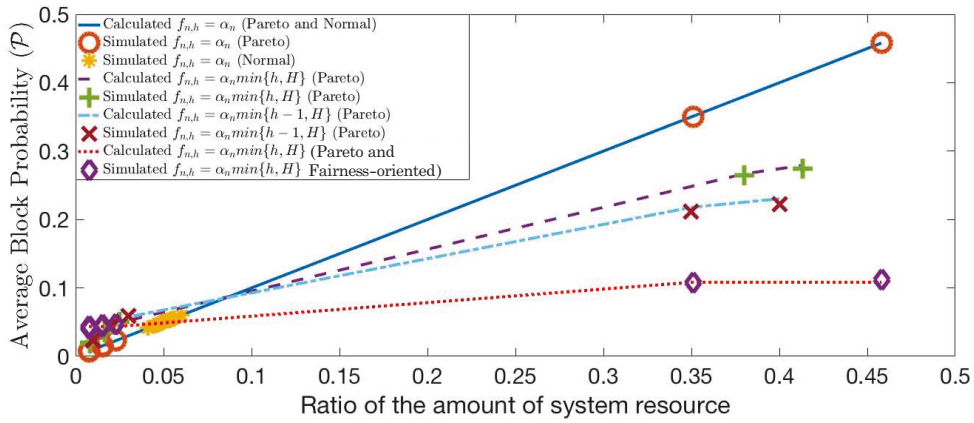


Figure 3.6 : The correlation between the resource ratio and the average block probability $\mathcal{P}$, where $N = 20$, $H = 10$, $\alpha = \frac{1}{2H\sum\omega_i}$.

same initial state. We name the number of rounds required to reach the steady-state *system convergence period*. During this simulation, we set that the steady-state is reached once the gap between the calculated and the simulation value of $\mathcal{T}_i$ decreases down to 3% (the explanation of 3% refers to the definition of *System Convergence* in Section 3.3.4).

Finding 3: For PoX schemes disabling the Fairness function, 80/20-rule-based wealth inequality deteriorates System Convergence, compared to normal-distributed wealth inequality.

This finding is shown in Fig. 3.7, where a Pareto distribution applying to $f_{i,h} = \alpha_i$ (the brown box) takes the longest time to reach the steady-state, while it converges the most quickly with a normal-distributed resource (the purple box). Thus, PoP resulting in a normal-distributed resource has the lowest number of rounds to reach the steady-state, compared with those with a Pareto-distributed resource. This is because of the outlier of Pareto-distributed resource overwhelmingly dominates the mining process.

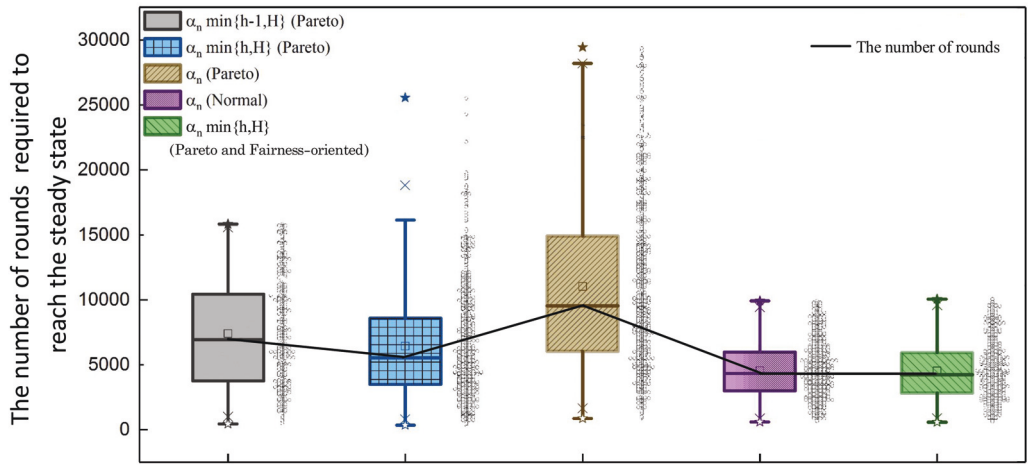


Figure 3.7 : The comparison among the four different PoX schemes in terms of *System Convergence*, where $N = 10$, $H = 5$, $\alpha = \frac{1}{2H\sum\omega_i}$.

Finding 4: The system convergence period can be reduced by enabling Resource Fairness and applying a positive Degree function (that increases the mining probability by multiplying the resource ratio and the awaiting gap).

According to Fig. 3.7, it can be found that $f_{i,h} = \alpha_i \min\{h, H\}$ (the green box) with an active Fairness function needs fewer rounds to reach the stead-state than that of $f_{i,h} = \alpha_i \min\{h, H\}$ (the blue box) with an inactive Fairness function. On the other hand, $f_{i,h} = \alpha_i$ (the brown box) with Degree function $\Phi(\cdot) = 1$ has longer *system convergence period* than the blue box with a positive Degree function. This is because the active Fairness function and positive Degree function apply a stronger restriction to the Monte Carlo variables, compared to those without an inactive Fairness function.

3.5.2.4 Resource fairness

Fig. 3.6 shows that none of the considered PoX schemes (except the red dot line) is upper bounded and Fairness-oriented, as the $\mathcal{P}_i$ of wealth miners remain a linear increasing based on the resource ratio owned by each miner. In other words, *Resource Fairness* is inactive for these PoX schemes, while *Resource Fairness* holds in some circumstances (i.e., $\Upsilon(\omega_i) \neq \omega_i$, to meet different requirements) for the Fairness-oriented PoS-Velocity listed in Table 3.1, referring to the red dot line in Fig. 3.6 and green box in Fig. 3.7. In this section, we show how *Resource Fairness* "encourages" such kind of PoX schemes to achieve better performance of *Resource Sensitivity* and *System Convergence*.

Recall that we implement a typical type of PoS-Velocity (see Section 3.5.1) as an example of a Fairness-oriented PoX scheme. It is revealed in Fig. 3.6, where the considered PoS-Velocity has the best performance on *Resource Sensitive* (i.e., the smallest $E(g)$) among all of the considered PoX schemes. In addition to the better performance of *Resource Sensitivity*, $\mathcal{P}$ remains constant when the upper bound is met with a partition function $\zeta(\omega)$. In other words, *Resource Fairness* can be satisfied with a simple linear $\Upsilon(\omega_i)$ being substituted by the design of a partitioned $\Upsilon(\omega_i) = \zeta(\omega)$. Thus, the considered PoS-Velocity prevents wealthy miners from monopolizing the entire network and incentivizes all miners to participate in the mining process and getting rewards.

Furthermore, the 80/20-rule-based wealth inequality can be addressed by the considered PoS-Velocity. Fig. 3.7 shows that *System Convergence* of the considered PoS-Velocity with a Pareto distribution (the green box) performs as good as that of PoP, i.e., $f_{i,h} = \alpha_i$ with a normal-distributed resource (the purple box).

It turns out that by enabling *Resource Fairness* with the designed $\Upsilon(\cdot)$ and $\Phi(\cdot)$, the considered PoS-Velocity achieves,

- **best** *Resource Sensitivity*: the best performance on *Resource Sensitivity* among any other non-Fairness-oriented PoX schemes listed in Table 3.1, based on the red dot line in Fig. 3.6;
- **improved** *System Convergence*: a performance on *System Convergence* that is as good as that of PoP with a normal-distributed resource, based on the comparison between the purple and green boxes shown in Fig. 3.7.

3.5.2.5 Summary

To sum up, apart from the considered PoS-Velocity scheme (defined in Section 3.5.1), other Fairness-oriented PoS-Velocity schemes can also reveal their optimized *Resource Sensitivity* and *System Convergence* by using our model. This can be achieved as long as the proper $\Upsilon(\cdot)$ and $\Phi(\cdot)$ are set (e.g., partition $\Upsilon(\cdot)$ and non-linear $\Phi(\cdot)$). By using the proposed model, we reveal that carefully designed *Resource Fairness* is particularly important to balance *Resource Sensitivity*, and *System Convergence* of PoX-based consensus algorithms in the long-term steady-state. Such steady-state analysis and findings have not been possible without our model.

3.6 Conclusions

Remark that participators hardly succeed to evaluate the profits by observing the generic property of any PoX schemes due to the lack of a unified analytical model. This question was resolved in this chapter by developing a new infinite-dimensional Markov model to unify the steady-state analysis for weighted resource distribution of different PoX-based Blockchains in large-scale networks. The probability of an arbitrary node being elected as the block generator was derived. Based on the analytical model, we evaluated PoW, balance-based and coinage-based PoS, PoA and PoP, in terms of *Resource Sensitivity*, *System Convergence*, and *Resource Fairness*. We also assessed a typical PoS-Velocity scheme with a weight consisting of the proper set

Fairness function and Degree function, and showed the balanced performance of the scheme in regards to all the three metrics. Extensive simulation results also prove that the applicability and generality of the model. This can significantly encourage the adoption of Blockchains in large-scale networks that provide public services to the communities.

Chapter 4

Scaling-out Blockchains with Sharding

The decentralized security relies on the consensus regarding the duplicated communication, storage, and computation among each full node in a large-scale network. However, the duplicated overhead is inevitable, leading to the scalability issue, i.e., an upper limit of Blockchain throughput, due to the limited communication, storage, and computation resource in a live context. Having an ability to increase the throughput along with the increasing network size becomes crucial. One of the most practical methods to overcome this issue is sharding, by partitioning network into multiple shards so that the overhead of duplicated resources among full nodes can be downgraded while maintaining the decentralized security. There has been existing studies presenting various sharding mechanisms, but none of them provides a comprehensive comparison. Industry and academia are desperate for a generic and systematic framework to learn various sharding mechanisms. Under this background, this chapter refers to [205], presents a detailed comparison and evaluation of major sharding mechanisms, along with our insights analyzing the features and restrictions of the existing solutions. The remaining challenges and future research directions are also reviewed.

4.1 Introduction

Sharding is first proposed by [49] and commonly used in distributed databases and cloud infrastructure. Based on the pioneering proposals [53, 120] integrating sharding with permissioned and permissionless Blockchain, respectively, the sharding technology is thought to be able to partition the network into different groups

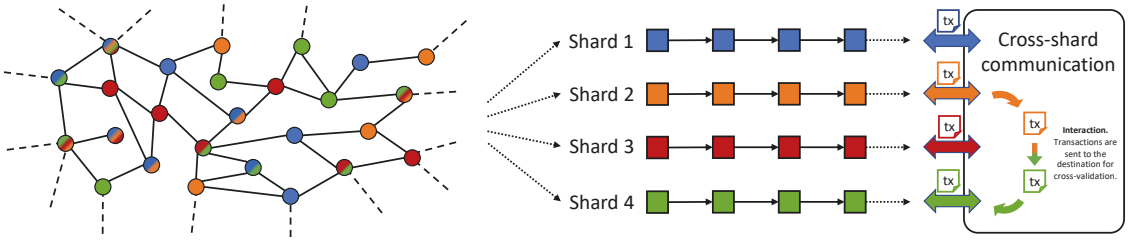


Figure 4.1 : The sharding technology partitions the network into different groups, while each of the groups maintains its own ledger and processes and stores a disjoint set of transactions. By implementing a secure cross-shard communication protocol, such disjoint transaction sets that could not have been interacted become securely verifiable and interactively executable in parallel. Note that, nodes in some sharding mechanisms (e.g., Monoxide) can choose to participate in the processing of multiple shards and maintain their ledgers, as illustrated by the multicolored circles, while the unicolored circles denote the nodes only participating in a single shard to which they are assigned in terms of the color.

(shards), so that the compulsory duplication of three resources (i.e., the communication, data storage, and computation overhead) can be avoided for each participating node, while these overheads must be incurred by all full nodes in traditional non-sharded-Blockchains. This partition is essential because the restriction incurred by the three resources owned by a single node may make the system unable to take full advantage of a scalable consensus algorithm. Sharding is so far one of the most practical solutions to achieve a *scale-out* system where the processing, storage, and computing can be conducted in parallel, as illustrated in Fig. 4.1. As such, the capacity and throughput being linearly proportional to the number of participating nodes or the number of shards become possible, while preserving decentralization and security. However, sharding poses new challenges to Blockchains, i.e., the *intra-consensus-safety*, *cross-shard-atomicity*, and the *general improvements*:

- *intra-consensus-safety*: how to secure the consensus algorithm inside a shard away from both the Nakamoto-based and BFT-based 1% attack [37] in a scalable way, while the latter can also be corresponding to a secure randomness

generation process, as discussed in Section 4.2; note that 1% attack is an attack strategy in sharded networks where attackers can dominate a single shard more easily than dominating the whole network;

- *cross-shard-atomicity*: how to support the cross-verification, and guarantee the *Atomicity* [74, 77] of cross-shard transactions for both unconditional transactions (simple payment) and conditional contract-oriented transactions in an efficient way (inefficient if the latency and overhead for achieving atomic-safe cross-shard transactions are higher than $O(n)$; n denotes the number of shards being partitioned or the number of participating nodes), as discussed in Section 4.3;

- *general improvements*: based on the *intra-consensus-safety* and *cross-shard-atomicity*, we focus on the improving factor $\mathcal{N}$ regarding the multiple of optimized global throughput for each considered sharding mechanism, while $\mathcal{N}$ is subject to the linear order $O(n)$. On the other hand, the additional latency and overhead originated from the proposed solutions also reveal the new problems that sharding brings to us. In regard to this, some *general improvements* are discussed in Section 4.4.

There have been a few studies working on these challenges regarding the sharding in permissionless Blockchains [120, 99, 206, 5, 37, 184], prior to which [53] proposes a sharded permissioned Blockchain that will not be discussed in this chapter due to its forfeit of permissionless decentralization. Rather, the sharding in permissionless Blockchains is focused (as permissioned Blockchains do not take full advantage of the sharding technology due to the smaller network size and its forfeit of permissionless decentralization). Also this chapter is based on the published research papers and other research references of Monoxide [184], Elastico [120], OmniLedger [99], Rapidchain [206], Chainspace [5], and Ethereum 2.0 [37]. Our contribution can be

characterized as follows.

1. Our work, for the first time, provides an introduction of state-of-the-art sharding mechanisms ranged from BFT-based to Nakamoto-based sharding mechanisms, while the latter has never been systematized in any of the existing surveys at the time of writing.

2. We gain our own insights analyzing the features and restrictions into the existing solutions to the intra-consensus-safety, atomicity of cross-shard transactions, and general challenges and improvements proposed by the considered sharding mechanisms. Based on the insights of the features and restrictions of each existing sharding solution, a comprehensive comparison is proposed.

3. Finally, we point out the current remaining challenges of sharding mechanisms, followed by suggestions for the future trend of designing reliable sharding mechanisms.

4.2 Intra-Consensus Protocol

Sharding significantly increases the throughput in $O(n)$, but sacrificing security in intra-consensus protocols, i.e., the per-zone security or 1% attack [184, 37]. Concretely, it is categorized into the Nakamoto-based 1% attack and BFT-based 1% attack.

The total amount of mining power among the network, i.e., $\mathbb{P}$, guarantees the low probability for a single entity to dominate over 50% mining power. By purposely dividing the network into n partitions (shards), we can greatly increase the throughput in $O(n)$, where rational miners tend to ideally distribute their mining power in multiple shards (at most n shards) in order for the maximum rewards. However, this also decreases the security of PoW in each shard in $O(1/n)$. Such a system can be more prone to double-spend attack by a malicious miner that only needs to

own the mining power $\mathcal{P} > \mathbb{P}/n \times 50\%$ due to the smaller shard size compared to the entire network size. This issue deteriorates as n increases in order for a larger throughput, which becomes the most serious barrier to PoW being implemented for the intra-consensus protocol of a sharding mechanism.

On the other hand, BFT-based consensus algorithms are considered instead of PoW in order to solve the security challenge, as discussed above. However, such designs introduce another kind of vulnerabilities other than that of the PoW-based one, as discussed in the following.

- It is of importance to carefully design a scheme to generate an unpredictable and unbiasable randomness without any third-parties in permissionless Blockchains. The randomness can be used to 1) allocate validators (an alias for nodes participating in the intra-consensus process in the context of BFT-based systems) into different shards at the beginning phase and every reconfiguration phase; 2) select the leader of each shard; and 3) decide which shards a cross-shard transaction should broadcast to, etc. Without such a strictly-chosen randomness, malicious validators may be able to bias the allocation and control the elections at will, such as collusion within a shard (with a small number of validators due to the weak scalability of traditional BFT-based consensus algorithms [59], e.g., PBFT [40]).

- Then it ends up encountering the dilemma of BFT-based 1% attack that the weak scalability of BFT-based consensus algorithm restricts the shard size, i.e., the number of members in a shard, while too small a size can potentially decrease the security of the intra-consensus with a strict fault-tolerance (FT), as described by the following cumulative binomial distribution,

$$s(c, m, p) = P[X \leq c] = \sum_{k=0}^{c} \binom{m}{k} p^k (1-p)^{m-k},$$
$$f(c, m, p) = 1 - s(c, m, p), \tag{4.1}$$

where X is the random variable that represents the number of times a malicious miner is picked [98, 120, 99, 62]; m denotes the shard size; c denotes the number of malicious members within a shard; and p denotes the total FT among the entire network. It is strongly suggested that $s(c, m, p)$ should be greater than 99% [98], while only $m \gtrsim 144$ can satisfy, of which the traditional BFT-based consensus algorithm cannot be capable[1]. In order to resolve this, highly scalable BFT-based consensus algorithms with large shard size require more attractions.

In this section, we compare and discuss the *intra-consensus* protocols of the considered sharding mechanisms, i.e., Monoxide, Elastico, Chainspace, OmniLedger, RapidChain, and Ethereum 2.0. Note that the Shasper used in Ethereum 2.0 features its novel and engineering-oriented design that combines the two major issues (*intra-consensus-safety* and *cross-shard-atomicity*) and kills two birds with one store. Elastico and Chainspace directly use PBFT for *intra-consensus* that are not discussed in detail in this section. This is because, as the first practical BFT-based consensus algorithm, PBFT has been the baseline of any other more advanced BFT-based consensus algorithms. The randomness generator of Chainspace is also not discussed as the detail is not provided in [5].

Also note that, a threat model where the attackers can refuse to participate or collude others (behave arbitrarily) takes effect in all discussed sharding mechanisms in this chapter. Also, Elastico [120], OmniLedger [99], and RapidChain [206] assume

[1] A few sharding mechanisms are incurring a total 25% FT based on the 33% FT in each shard, e.g., Elastico, OmniLedger, and Chainspace. This can be a BFT-based 1% attack, by dispersing validators into as many shards as possible to maximize the possibility to control some shards. Elastico and Chainspace suffer from this security issue, while OmniLedger implements a scalable BFT-based consensus algorithm to address this issue.

the slowly adaptive attackers (who can only succeed to attack in a long time), while Monoxide [184], Ethereum 2.0 [37], and Chainspace [5] assume a model of uncoordinated majority where all participators are game-theoretically rational, i.e., egoism (with an upper-bounded fraction that can coordinate the majority). Therein Chainspace [5] also introduces an audit scheme to prevent attacks from dishonest shards.

4.2.1 Nakamoto-based - Monoxide - Chu-ko-nu mining

Monoxide is the first sharding mechanism that eliminates the need for generating randomness, and implements Nakamoto consensus algorithm for its intra-consensus. It introduces a one-off bootstrapping in the beginning, to allocate each node (including miners and non-miners) into different shards based on their identity addresses. By using the proposed Chu-ko-nu mining, Monoxide can achieve a large-scale network with a huge number of shards and a flexible shard size. It involves a Merkle Patricia Tree (MPT) [191] root consisting of all proposed blocks among multiple shards, thus the $\mathbb{P}/n$ can be multiplied by a factor k (k denotes the number of shards a particular miner manage to mine on). Consequently, dispersing mining power can be re-aggregated to solve the 1% attack, i.e., $\mathbb{P}k/n \simeq \mathbb{P}$ as $k \to n$.

In order to meet the requirement of $\mathbb{P}k/n \simeq \mathbb{P}$, Monoxide needs most of miners to conduct Chu-ko-nu mining across as many shards as possible, i.e., $k = n$ in the best case. However, this implies the fact that if miners only mine on k out of n shards, i.e., $\mathbb{P}k/n$, where $k \ll n$, the factor expected to amplify the effective mining power will be too small to secure the mining process, hence reducing the attack cost. On the other hand, rational miners tend to mine on all n shards to reap the maximum profit, which may also result in the power centralization due to the huge cost of bandwidth, disk storage, and computing processors that only the professional mining facilities can afford.

Insight 1. *The amplification to the effective mining power relies on an incentive scheme that should encourage miners to mine across $k \rightarrow n$ shards in Chu-ko-nu mining. This also poses the issue of power centralization and additional overhead to Monoxide.*

4.2.2 BFT-based - Elastico

Using BFT-based algorithms for the intra-consensus is an alternative to bypass the vulnerability of Nakamoto-based algorithm (**Insight 1**). Thus, including but not limited to Elastico, OmniLedger, RapidChain, Chainspace, and Ethereum 2.0 choose to implement BFT-based algorithm. Therein, Elastico uniformly (re)allocates potential validators in terms of the different least-significant bits of the unpredictable PoW solutions at the beginning of each epoch, followed by running PBFT for the intra-consensus. The randomness used during the mining is generated by a proposed distributed commit-and-xor scheme.

Consensus Algorithm - PBFT's restrictions in sharding

Due to the weak scalaibilty of PBFT, Elastico incurs an unacceptable failure probability of 8% with $f(c, m, p) = f(6, 16, 0.25)$ based on the result of [59], while it still incurs 2.76% with $f(c, m, p) = f(34, 100, 0.25)$ even extending to a larger-scale network of $m = 100$ (which can be the bottleneck [99]) by running powerful servers in cloud. This security issue has been hindering Elastico to be practically used, which are greatly resolved and improved by OmniLedger and RapidChain.

Insight 2. *The traditional non-scalable PBFT incurs unacceptably high failure probability with total FT of only 25%, unless increasing the size of the consensus group, which leads to a chicken-and-egg problem due to huge communication over-*

head.

Generating Randomness - Distributed commit-and-xor scheme

The distributed commit-and-xor scheme is implemented for the randomness generation in Elastico. It can be categorized into the commit-and-then-reveal scheme [136], with an exception that the final result (randomness) varies depending on the different combinations of seeds λ_i every validator chooses. Also, the randomness generation is conducted by a global subset, i.e., the final committee.

This design, however, is not perfectly unbiased. It is exponential biased and bounded by the size of λ_i, i.e., $|\lambda_i|$, and m (m denotes the size of the final committee). In order to prevent the attacks from biasing the randomness by deliberately choosing a specific set of $m/2 + 1$ values of λ_i in his favor, $|\lambda_i|$ should be large enough as m also increases. This incurs large communication overhead, in addition to the overhead of the extra verification during PoW process. In the case of only $2m/3$ values of $(\lambda_i, Hash(\lambda_i))$ being received, the lack of Verifiable Secret Sharing (VSS) [65, 147, 175] forces all senders of these $2m/3$ values to be online all the time with no network outage or delay.

Insight 3. *The distributed commit-and-xor scheme of Elastico has weak availability and robustness, and it is not a perfectly unbiased randomness generator unless paying more for the communication overhead.*

4.2.3 BFT-based - Chainspace

Chainspace uses an optimal implementation of PBFT, Mod-SMaRt [174], which accounts for the intra-part of the $\mathcal{S}$-**BAC** protocol proposed by Chainspace. However, Mod-SMaRt does not scale PBFT to address the issue of 1% attack. It decouples the communication and consensus primitives, while it only reduces the

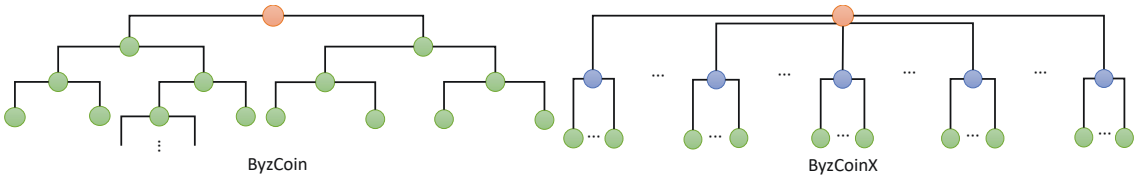


Figure 4.2 : (Left) ByzCoin implements a tree with a fixed branching factor and an increasing depth. (Right) ByzCoinX implements a shadow tree with a fixed depth and an increasing branching factor.

overhead of the latter with an unchanged overhead of $O(n^2)$ by replacing the process with the Validated and Provable Consensus (VP-Consensus). In addition, the high failure probability of the intra-consensus in Elastico also takes effects in Chainspace, which restricts the use of Chainspace in a large-scale network. Note that, the stages of *Propose* and *View change* take as input the elected leader, while the detail of randomness generator is not provided in [5].

4.2.4 BFT-based - OmniLedger

OmniLedger combines RandHound [177] and Algorand-based Verifiable Random Function (VRF) [71] to produce an unpredictable and unbiasable randomness under a 25% FT for re-allocation and leader-election of each shard and sub-group. Also, a new scalable BFT-based consensus algorithm, ByzCoinX, is proposed by optimizing ByzCoin [98], which resolves the dilemma of BFT-based 1% attack in sharding, by increasing the shard size to hundreds and up to a thousand.

Consensus Algorithm - ByzCoinX

Initially, ByzCoin [98] was the first scalable consensus protocol that combines PoW and BFT algorithms in a tree-based structure, by means of scalable collective signing (CoSi) [33, 178]. ByzCoinX optimizes ByzCoin in terms of the better latency and more robust FT for a shard with hundreds of validators. Concretely, ByzCoinX implements a shallow tree with a fixed depth-3 and an increasing branching factor.

Based on the shard size, each group leader is responsible for a group forming a sub-tree with a fixed number of group members.

By using such a new tree-based structure, ByzCoinX can outperform ByzCoin by a better latency for a shard with hundreds of validators due to the shorter path from leaves to the root with a fixed depth, and a robust fault-tolerance due to the increasing branching factor. When the number of validators goes above a threshold, the latency of ByzCoin outperforms that of ByzCoinX due to the increasing branching factor. On the other hand, ByzCoinX can achieve a failure probability around 1.5% with $f(c, m, p) = f(48, 144, 0.25)$, and even 1% with $f(342, 1024, 0.3)$ at the cost of latency, as shown in Fig. 10 of [99].

Insight 4. *ByzCoinX improves the scalability with a lower failure probability for the intra-consensus of OmniLedger, by sacrificing the transaction latency in large-scale networks.*

Generating Randomness - Combination of RandHound and VRF

In order to address the issue of **Insight 3**, OmniLedger implements a scalable bias-resistant distributed randomness generator, RandHound [177], combined with a VRF-based leader election algorithm proposed by Algorand [71].

Insight 5. *The combination of RandHound and VRF suffers from the reliance on a third-party initial randomness pre-defined in the genesis block. A falling-back to an inefficient scheme occurs in the context of asynchronous networks, which limits the salability that RandHound could have guaranteed.*

4.2.5 BFT-based - RapidChain

RapidChain [206] implements a VSS-based [65] distributed random generation (DRG) protocol to agree on an unbiased randomness. On top of the DRG protocol, RapidChain addresses **Insight 5** by introducing a deterministic random graph where a certain fraction (50% with high probability [206]) of the number of malicious validators can be guaranteed in the initial set (the reference committee, similar to the final committee in Elastico). Inspired by [159], in addition, RapidChain resolves the dilemma of BFT-based consensus algorithm in sharding, by increasing the FT of the intra-consensus protocol up to 50%.

Consensus Algorithm - 50% BFT

RapidChain aims for higher FT (50% BFT) of the intra-consensus protocol to address the dilemma of BFT-based 1% attack for sharding mechanisms with a small shard size. To be specific, RapidChain runs an autonomous pre-scheduled scheme within a shard to agree on a timeout Δ, based on which the consensus speed can be adjusted by the system to prevent the asynchronization. This ensures a synchronous network in the long-term, in which a non-responsive synchronous (with constant rounds) BFT-based consensus protocol with FT of 50% can be used. Referring to (4.1), the design of 50% BFT achieves a failure probability around 1.5% with $f(c, m, p) = f(17, 32, 0.33)$, and even 1% with $f(51, 100, 0.39)$ at a cost of communication overhead.

Insight 6. *Differing from ByzCoinX in OmniLedger, the* 50% *BFT of RapidChain solves the BFT-based* 1% *attack by increasing the FT of intra-consensus protocol, nevertheless, this can only suit small-sized shards (not scalable with communication overhead of* $O(n^2)$*). In addition, the pre-scheduled scheme defining the timeout is not conceivably proved synchronous enough to run the pipelining* 50% *BFT.*

Generating Randomness - VSS-based DRG protocol

The proposed DRG protocol by RapidChain, in fact, only implements a basic VSS-shares scheme, where all participating validators can reconstruct the final randomness r by the share of r (the share equals to $\sum_{l=1}^{m} \rho_{lj}$ calculated by other validators except validator-j) received from other validators. Note that, $\rho \in \mathbb{F}_p$ denoting a finite field of prime order p, and m denotes the size of the reference committee. As a result, the DRG protocol encounters a similar issue to that of any other typical VSS scheme, i.e., non-scalable (even though it suits with the 50% BFT in small-sized shards).

4.2.6 BFT-based PoS - Ethereum 2.0

Ethereum has been running publicly as the first decentralized Blockchain platform (Blockchain 2.0 that implements a Turing-complete programming language to develop smart contracts for the first time since 2014 [191]. With the gradually rising demands of high throughput, Casper-FFG with sharding (Shasper) is proposed [37] to allow the current Ethereum mainnet (a PoW-based single chain, also referred to Ethereum 1.0) to migrate to the new architecture stably and securely.

Consensus Algorithm - Solving the intra-consensus in a global way

Shasper chooses the BFT-based consensus algorithm to solve the 1% attack issue of *intra-consensus*. Concretely, the Casper-FFG of Shasper can be regarded as a variation of BFT-based PoS consensus algorithms [71, 95] with careful designs for generating randomness, as opposed to the virtual-mining PoS consensus algorithms [97, 194]. Note that, we assume a scalable BFT algorithm similar to ByzCoin [98] and ByzCoinX of OmniLedger is used in Shasper.

Shasper decouples the member allocation and consensus process, which leads to

the fact that the intra-consensus within a shard also involves those validators from other shards being the attesters. The members of attesters group associated with a specific shard can be updated every slot. This implies that an eligible validator in Shasper should at least store all block headers (headers is called collations in Shasper) of all shards regardless of which shard this validator is allocated at the beginning of every epoch. The group of attesters can be re-allocated for each proposed collation in a times slot, which provides the strongest security but incurs huge overhead when, 1) each shard conducts the consensus among continuously updated validators; 2) validators need to store data of more shards; and 3) the 1-slot-period re-allocation has to be executed.

Insight 7. *The security level of Ethereum 2.0 - Shasper provides more flexible allocation for intra-consensus than that of any other considered sharding mechanisms, nevertheless, by incurring larger overhead.*

Generating Randomness - Combination of RANDAO and VDF

RANDAO [2] is implemented based on the commit-and-then-reveal scheme [136] written in a pre-defined smart contract running on the beacon chain. A Verfiable Delay Function (VDF) as a "hash onion" [32] is implemented on each seed λ to ensure the unbiased randomness, as only the serial computing can be run regardless of the computation power owned by a validator. The design is thought as a feasible solution to prevent malicious manipulation such as deciding not to reveal the commitment. However, the design still suffers from three flaws:

- A VDF consisting of n times $Hash(\cdot)$ incurs a computation overhead of $O(n)$, which is inefficient. There have been a few advanced VDF schemes proposed by the recent researches [189, 148, 66].

- This design is prone to the censorship attack [35]. Malicious validators can send irrelevant transactions with a high gas fee to fill up a block. Thus, the commitment may have to be interrupted as the gas limit of the block is run out.

- This design is also prone to the grinding attack [47] if the seed λ is based on the hash of the parent block, because validators can send arbitrary transactions, and try to find out the most biased seed by collecting different sets of transactions.

Insight 8. *Current design of randomness generator in Ethereum 2.0 incurs high computation overhead, and is overwhelmingly dependent on the incentive scheme (punishment). It is prone to censorship attack and grinding attack, if the attack cost is acceptable.*

4.3 Atomicity of Cross-Shard

It is of importance that a sharding mechanism can support the cross-shard-verification and cross-shard transactions for validators allocated in different shards, according to the result shown in [99, 206] (showing that the probability of cross-shard transactions approaches to 100% as the total number of shards increases). Maintaining an individual global root chain may be one of the solutions to verification, but it does not natively support cross-shard transactions without any additional mechanism, e.g., lock/unlock operation in synchronous networks or lock-free operation in asynchronous networks. The demand for a secure protocol of cross-shard transactions gradually outweighs a naive mechanism lacking the support of cross-shard transactions (even it can achieve a high improving factor $\mathcal{N}$).

Differing from the traditional database system, the support of cross-shard trans-

actions proposes a challenge to guarantee the *Atomicity* of the data that was first defined in [74, 77] across multiple shards. Not only a simple payment transaction involving withdraw and deposit operations needs to be atomically protected, but also the demand for the complicated conditional statements attracts more attention to the contract-oriented *Atomicity*.

In this section, we compare and discuss the protocols to achieve *cross-shard-atomicity* in the considered sharding mechanisms. We focus on the design of cross-shard transaction, including Monoxide that supports asynchronous lock-free simple payment transactions; OmniLedger, RapidChain, and Ethereum 2.0 that supports simple payment transactions with lock/unlock scheme; and Chainspace that supports cross-shard operations for smart contracts (Elastico is vaguely discussed as it does not support atomic-safe cross-shard transactions.

4.3.1 Monoxide - Relay Transactions

In order to bypass the overhead of lock/unlock operation that greatly constrains the throughput and performance in regards to cross-shard transactions, Monoxide proposes *Eventual Atomicity* where a single cross-shard transaction is decoupled into an originated transaction (t_l) in the local shard, and a relay transaction (t_r) being put into the outbound transactions set (and hence becoming an inbound transaction when it is received by the destination shard). Rather than the immediate atomicity, *Eventual Atomicity* features its lock-free design and takes advantage of Chu-ko-nu mining across parallel shards in an asynchronous network, in order to maximize the global throughput via simple message exchange. As a result, a cross-shard transaction in Monoxide achieves an improving factor of $\mathcal{N} = \frac{n}{2}$ as it is split into the locally-executed transactions and relay transactions expected to be outbound.

However, differing from the cross-shard transactions that can be proactively rejected by an acknowledgement from an entity (this is in charge by clients in Om-

niLedger, as discussed later), the chain forking in Monoxide can cause a reversion of the history and orphanize the block containing the t_l that has been executed within a shard. Without any existing of acknowledgement reminding the originated shard the status of t_r in the destination shard, the forking not only invalidates t_r in the destination shard (if t_r has been sent out before the forking occurs), but also invalidates all the subsequent cross-shard transactions relayed to any other shards. This implies the following drawbacks.

Incompatibility to Smart Contracts. There does not exist an upper-bound of timeout indicating if *Eventual Atomicity* of a cross-shard transaction has been finalized, leading to the incompatibility of conditional transactions, e.g., complicated operations in smart contracts.

Additional Latency. There must be λ confirmation blocks delaying the execution of the inbound transaction, i.e., t_r, in order to ensure the corresponding t_l in the originated shard is finalized and unlikely reverted. Also, the absence of acknowledgement and strict upper-bound of timeout deteriorates the latency and throughput due to the inevitable message loss, which incurs additional latency.

Unexpected Replay. To invalidate the inbound transactions t_r and all the subsequent t_rs due to the failure and reversion of t_l in the originated shard, and prevent the history of all destination shards from being reverted, the history needs to be rebuilt from the genesis block of each shard. This incurs unexpected overhead even if a checkpoint scheme is introduced, e.g., the shard pruning in OmniLedger [99].

Insight 9. *In order to maximize the global throughput, Eventual Atomicity achieves the lock-free asynchronous cross-shard transactions at the cost of incurring Incompatibility to Smart Contracts, Additional Latency, and Unexpected Replay.*

4.3.2 Elastico - No cross-shard Transactions

The elected leader of the traditional PBFT consensus algorithm in each shard finalizes and sends an agreement in regards to local transactions to a global subset, i.e., the final committee, as discussed in Section 4.2.2. A final global block is stored in the global ledger and broadcast to all validators among the network, so that validators can verify the transactions from other shards. However, Elastico does not provide a secure protocol to ensure the atomicity across shards via this global ledger. There will be a fund loss as an unexpected dead-lock occurs if the cross-shard transaction sent to the destination shard gets rejected.

4.3.3 OmniLedger - Atomix Protocol

To simplify the *cross-shard-atomicity*, OmniLedger proposes a client-driven Atomix protocol that is UTXO-based, where the communication overhead is shifted outside the shards. This indicates that the clients act as a gateway exchanging messages across multiple shards, by paying an extra cost of overhead themselves.

Consequently, a cross-shard transaction containing inputs from one single input shards (IS) and output shards (OS) can achieve an improving factor of $\mathcal{N} = \frac{n}{2}$, as this transaction is only stored in two shards, i.e., this IS and OS. On the other hand, inputs and outputs of multiple ISs and OSs result in the transaction being stored among the involved shards, i.e., an improving factor of $\mathcal{N} = 1$ in the worst case that the entire network is involved.

Insight 10. *Atomix Protocol is, in fact, a band-aid at best. It sacrifices the support of light-weighted clients, but requires powerful performance for a client-driven exchange of messages.*

Insight 11. *Atomix Protocol has poorer support for UTXO-based cross-shard trans-*

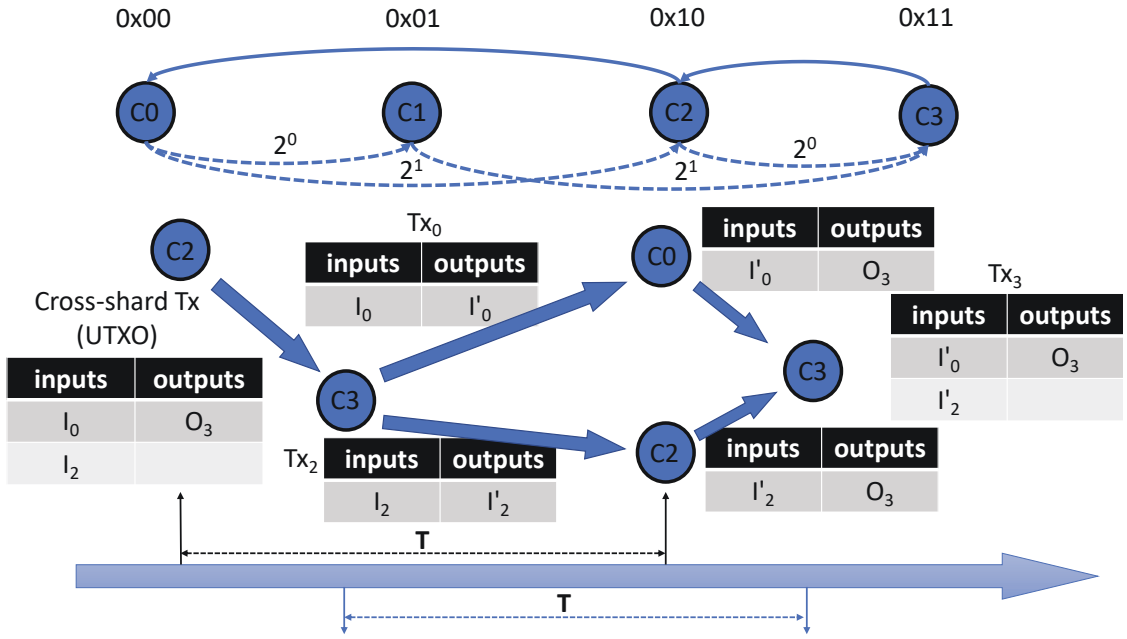


Figure 4.3 : (Top) Each committee (shard) maintains a routing table containing $\log_2 n$ other committees. The routing table improves the efficient communication among multiple shards, as described in Section 4.4.2. Committee C_0 can locate C_3 (via C_2) responsible for transactions with prefix $0x11$. (Bottom) To cross-validate a UTXO-based cross-shard transaction requires this transaction to be spilt in three-way confirmation.

actions as the number of participating shards increases, which is unable to take full advantage of the UTXO format.

4.3.4 RapidChain - Three-way Confirmation

To verify a UTXO-based cross-shard transaction, there proposes a three-way confirmation in RapidChain to optimize the Atomix Protocol in OmniLedger, as shown in the bottom part of Fig. 4.3. Concretely, $k-1$ sub-transactions (Tx_0 and Tx_2) destined for each committee that stores its own I_i of the cross-shard transaction, with I_i as the inputs and I'_i as the outputs, respectively, and k is the number of inputs of this cross-shard transaction, are created by the output committee, i.e., C_3 as the C_{out}. After passing the verification on each input committees, i.e., C_2 and C_0 as the two C_{in}(s) of the original cross-shard transaction, Tx_0 and Tx_2 are stored

in their own local ledger, respectively. Finally, all C_{in}(s) send the corresponding transactions back to C_3, and end up aggregating Tx_3 to be finally stored in the local ledger of C_3.

In order to determine the improving factor $\mathcal{N}$, we assume that a single committee can only be either a sender committee or a receiver committee (practically a shard can be both a sender or a receiver) at the same time for simplicity. In the worst case where a full-sized cross-shard transaction contains only the input from a single committee, C_{in} has to send this full-sized transaction twice (each corresponds to invoking the inter-communication once), i.e, 1-st and 3-rd handshaking. On the other hand, the period from C_{in} sending C_{out} the cross-shard transaction to it finishing verifying the sub-transactions received, equals to the period from C_{out} finishing verifying the original cross-shard transaction to it finishing verifying the confirmations sent by C_{in}, i.e., one block period. It is because the original cross-shard transaction is spilt into,

- the sub-transactions that are supposed to be stored in the local ledger of each C_{in} (a full-sized of the original cross-shard transaction with inputs from a single committee or inputs involving all committees);
- the final transaction that is supposed to be stored in the local ledger of C_{out} (another full-sized of the original cross-shard transaction) at the end of the protocol.

Consequently, either of these two kinds of transactions accounts for the intra-throughput of a committee, hence one block period, as shown by the $\mathbf{T}$ at the bottom of Fig. 4.3. Therefore, an improving factor of $\mathcal{N} = \frac{n}{2}$ can be achieved.

Insight 12. *The routing table and three-way confirmation resolve the issue of OmniLedger, by significantly reducing the overhead of communication, even with*

a large number of participating shards in a single UTXO-based cross-shard transaction. However, by polluting specific routing tables, the eclipse attack [80] becomes a concern.

4.3.5 Ethereum 2.0 - Using Receipts

Having known the beacon chain, validators can not only address the issue of *intra-consensus*, but also address the issue of *cross-shard-atomicity*, i.e., cross-verifying the normal transactions in each shard the validators care about, and enabling the cross-shard transactions. Note that, Shasper so far can only support a simple account-based (as opposed to the UTXO-based) payment transaction, while the design contract-oriented cross-shard transaction has not been finalized and presented.

The cross-shard transactions in Shasper rely on the receipts. Receipts correspond to accepted cross-shard transactions that are used to verify and log the validity of the transactions' operations. Also, the result of these operations can be obtained by the involved validators conducting cross-validation in the destination shards. By means of receipts whose identities are contained in *Txgroup root* field (Receipt root), the cross-shard transactions are split into multiple sub-transactions being executed in the originated and destination shards, respectively (the original transaction, a *proof-of-receipt*, and a *proof-of-response*). This can be regarded as a variation of the synchronous lock/unlock scheme implemented in OmniLedger and RapidChain, while the receipts take the actual role of the lock. Thusm, a cross-shard transaction that is account-based in Ethereum 2.0 - Shasper can achieve an improving factor of $\mathcal{N} = \frac{n}{3}$ due to the preliminary transaction, *proof-of-receipt*, and *proof-of-response*.

Insight 13. *Ethereum 2.0 - Shasper introduces account-based cross-shard transactions by implementing the global (stored by all validators) beacon chain to exchange the essential message, i.e., the receipts and proofs. However, Shasper cannot be more*

than a transitional version due to the disadvantage of possible overhead.

4.3.6 Chainspace - The inter-part of $\mathcal{S}$-BAC

$\mathcal{S}$-BAC refers to Sharded Byzantine Atomic Commit, whose intra-part makes use of an optimal PBFT, Mod-SMaRt, to handle the intra-consensus process; see Section 4.2.3. Upon the intra-consensus being finalized within a shard (Chainspace allocates nodes in different shards based on the objects management, as described in Section 4.4.6), the elected leader of the shard, the BFT-Initiator, takes responsibility for the atomicity of cross-shard transactions. It is worth noting that Chainspace makes use of the concept of BFT to ensure such atomicity, which constitutes the inter-part of **$\mathcal{S}$-BAC**. It resembles the Atomix Protocol in OmniLedger, with a crucial optimization where BFT consensus process must be conducted instead of a naive client-driven model. However, similar to the problem the Atomix Protocol of OmniLedger has encountered, i.e., **Insight 11**, the improving factor upon a cross-shard transaction can be ranged from $\mathcal{N} = n$ to $\mathcal{N} = 1$ with T containing only one input object and no object being output, and T involving all objects around the entire network, respectively.

4.4 General Improvements

In this section, some general key challenges and improvements particularly proposed by the considered sharding mechanisms are listed. Such improvements can be generally implemented to address the new issues the considered sharding solutions pose to the entire system, as shown in Figure 4.4. They include transaction latency, inter-communication protocol, shards ledger pruning, decentralized bootstrapping, securing the epoch reconfiguration, sharded smart contract, and replay attacks and defenses against cross-shard transactions.

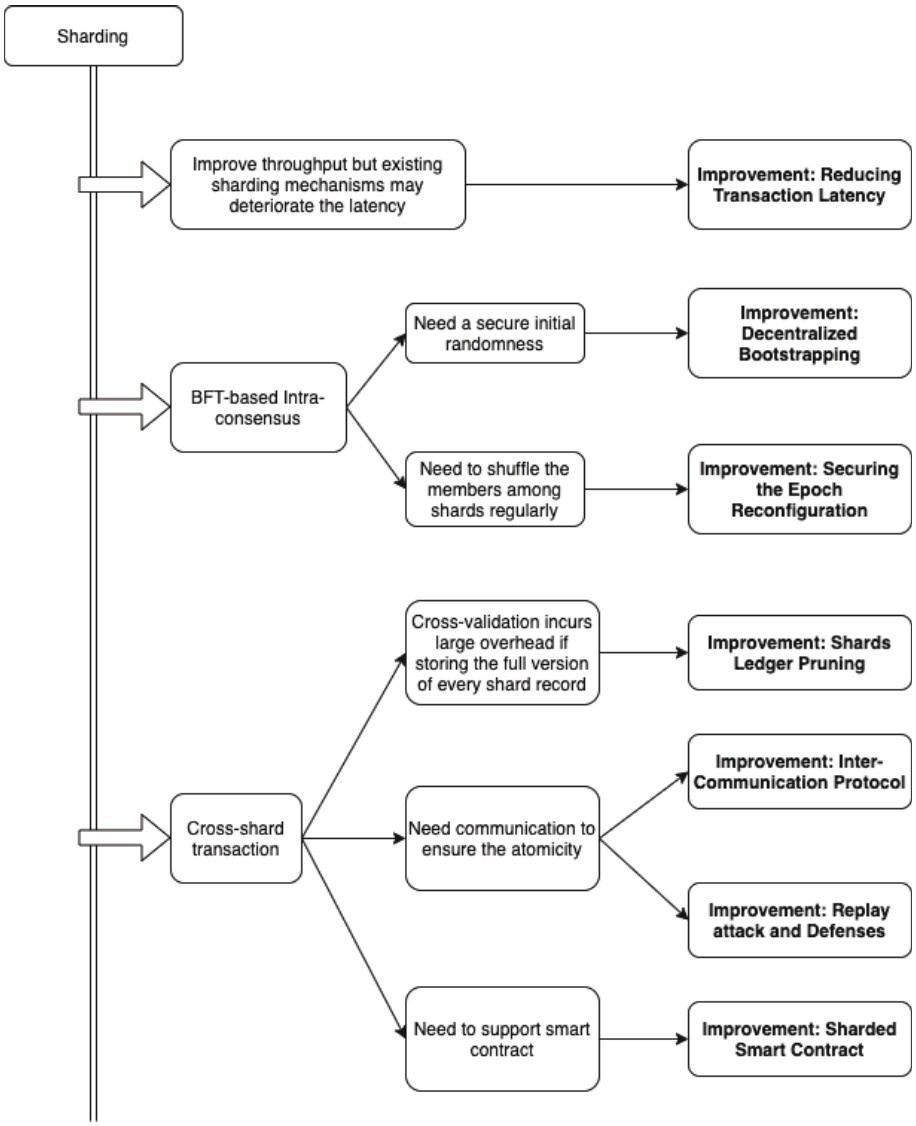


Figure 4.4 : The pain points that each of the proposed improvements are expected to solve

4.4.1 Reducing Transaction Latency

Apart from the throughput, the transaction latency, referring to how long a transaction is deterministically confirmed and finalized, is most likely more sensitive to individual users. It has been shown that the BFT-based 1% attack (refers to Section 4.2) can be either resolved by implementing a scalable BFT consensus, e.g., OmniLedger and Ethereum 2.0, or increasing the FT within a single shard, e.g., RapidChain. However, it remains the issue of transaction latency, as described below.

- *The transaction latency deteriorates as a scalable BFT consensus features a large scale shard size to address the 1% attack,* according to the evaluation shown in [98, 99]. Thus, Omniledger introduces the *trust-but-verify transaction validation* scheme running within each shard to provide the real-time transaction confirmation time, which can also be implemented in any compatible sharding scheme, such as Ethereum 2.0. Concretely, an optimistic validation and a core validation are conducted. Those users who are care more about the latency than security can choose to accept transactions after the optimistic validation is passed.

Insight 14. *The real-time transaction latency is achieved by sacrificing the security, as the further 1% attack can still happen in optimistic groups. Similar to IoTA [152], this real-time transaction latency can only be used in specific scenarios with lower security requirements.*

- *The transaction latency deteriorates as a non-scalable 50% BFT consensus incurs larger communication overhead.* Thus, upon the 50% consensus only agreeing on a digest of the block. RapidChain implements the *information dispersal algorithm* (IDA)-based gossip protocol [9, 10] to transmit large payload

more efficiently. Concretely, the sender divides the original message into some n-equal-sized chunks, followed by applying an (m, n) erasure code scheme to encode the n chunks to m chunks. As a result, each node can reconstruct the original message by receiving valid n chunks from its neighbors with the help of some proofs, e.g, the MPT proofs, hence significantly reduces the latency.

4.4.2 Inter-Communication Protocol

Differing from the protocol to achieve the *atomicity-cross-shard*, the inter-communication protocol focuses on the overhead of data transmission among shards. The related schemes discussed in this chaoter include the following two major types.

- A global root chain acting as a message distributor is implemented, while each validator (or miner in the context of Monoxide) needs to store this chain. Sharding mechanisms using this kind include Ethereum 2.0, Monoxide with identical PoW targets, and Elastico.

Insight 15. *The bottleneck is shifted to the global root chain due to its single-chained structure, as opposed to sharded structure. This can only be a transitional version but not a real solution.*

- The most straightforward way is used by OmniLedger and Chainspace, i.e., full-mesh connection. This requirement tends to hold in those latency-sensitive systems, which incurs an considerable overhead.

In order to bypass the full-mesh connection, RapidChain proposes a novel inter-communication protocol based on a routing table stored by each validator; see the top side of Fig. 4.3. It is inspired by Kademlia-based [126] routing protocol, where each validator in a shard maintains a routing table containing all members of its shard as well as $\log_2 \log_2 n$ validators of other $\log_2 n$ shards which are distance 2^i

for $0 \leq i \leq \log_2 n - 1$ away. The inter-communication is conducted by having all validators in the sender shard send messages to all validators on the receiver side. By taking advantage of P2P network, the communication overhead can be significantly reduced.

4.4.3 Shards Ledger Pruning

The reason most of the existing Blockchain system with a single-chained structure tends to store the full version of its chain is that they intend to improve the communication and computation overhead of censorship and audition. Storing a full version of ledger of every shard incurs an unacceptable overhead of disk storage to validators, as validators need to track the history of each shard in order to support the cross-shard transactions, as well as the re-allocation (bootstrapping) during each epoch. To solve this, OmniLedger proposes the design of state blocks (SB).

SBs of a shard summarizes the state as well as all transactions of its shard associated with each epoch. The design of SBs is similar to stable checkpoints in PBFT [40], fast-sync mode in Ethereum [38], and stable checkpoints of Node Hash-Chains in Chainspace [5]. According to the evaluation in [5], such kind of pruning incurs an overhead of $O(m + \log T)$ for a *partial audit* and $O(T)$ for a *full audit*, where m denotes the shard size, and T denotes the number of transactions. The *partial audit* allows any users to obtain a proof to verify the existence of any transactions in any shards; the *full audit* allows a full verification by replaying the entire history of a shard. However, the design of SB raises two issues, 1) the overhead of transaction proofs might become the bottleneck, but it can still be relieved by introducing the Simple Payment Verification (SPV) [135, 38], several multi-hop backpointers [138, 21, 158], or Proofs of Proof of Work (PoPoW) [93, 92]; and 2) **Insight 16**,

Insight 16. *The design of State blocks faces the same problem as that of the Atomix Protocol in OmniLedger and light-client protocol in Ethereum 1.0 (if used in Ethereum 2.0), i.e., shirking the most important duty to the client side.*

4.4.4 Decentralized Bootstrapping

For sharding mechanisms involving a randomness generator that is responsible for a PoW-based entry ticket in the BFT-based intra-consensus protocol, it is important to select the initial set with an honest majority, e.g., the final committee in Elastico, and the reference committee in RapidChain[2].

Thus, RapidChain proposes a decentralized bootstrapping in the form of *sampler-graph election network* [206], with only a hardcoded seed and some network settings. In such an election network, participating validators are uniformly distributed into a few groups, within each of which a PoW-based result is computed by each member based on the randomness generated by the VSS-based DRG protocol (Section 4.2.5) and its identification ID. Based on the result, a subgroup can be obtained for each group. Finally, a unique *root group* (it randomly selects the members of the reference committee) can be obtained with 50% honest majority (high probability), when this process is iterated. Consequently, the communication overhead can be improved from $\Omega(n^2)$ to $O(n\sqrt{n})$ with n denoting the total number of participating validators.

4.4.5 Securing the Epoch Reconfiguration

For sharding mechanisms running a BFT-based intra-consensus protocol, (new) validators have to be swapped-out and re-allocated in other shards every epoch in

[2]OmniLedger eliminates the necessity of an initial global set that responsible for verifying the PoW result, by using RandHound and VRF. However, an initial global randomness is still needed to derive VRF. Ethereum 2.0 builds the design on top of PoW-based mainnet, where the PoS-based Casper is used instead of PoW.

order to prevent attacks from slowly adaptive adversaries, i.e., attacker can corrupt or Distributed Denial of Service (DDoS)-attack validators, but it takes a bounded time for such attacks to take effect. This indicates that the epoch length should be carefully designed to be lower than the bounded time.

Recall that Elastico and Chainspace do not provide such a solution, while Ethereum 2.0 solves the intra-consensus with a global validator pool by frequently updating the member participating in the intra-consensus protocol for each shard. Both of them require validators to track the status of each shard to speed up the reconfiguration phase. OmniLedger implements a random permutation scheme to swap-out the validators, ensuring the number of validators being swapped is bounded by $k = \log n/m$ at a given time, where n denotes the total number of participating validators; m denotes the number of shards. Here, new validators that require to register their ID on a global identity Blockchain are also assigned to random shards. As such, the number of remaining honest validators can be sufficient to reach consensus while some are swapped-out, thus the idle phase can last shorter to improve the throughput. However, this scheme incurs a significant delay and scales moderately, which cause 1-day-long epoch that does not suit highly adaptive adversaries (when the bounded time becomes smaller).

In contrast, RapidChain proposes a light-weighted reconfiguration protocol based on the Cuckoo rule [19, 166], where only a constant number of validators are allowed to move between committees in each epoch. To be specific, the reference committee (C_r) announces a PoW puzzle based on the randomness generated in epoch $i-1$ ($\mathcal{R}_i$) by the DRG protocol, thus validators that wish to participate in epoch $i+1$ (including those that have participated in epoch $i-1$ and i) can solve the puzzle and inform C_r by the end of epoch i. During epoch $i+1$, C_r defines the active and inactive lists of validators of epoch $i+1$, and swap-out a constant number of validators from one to another committee based on $\mathcal{R}_{i+1}$ generated in epoch

i. Finally, C_r agrees on a reference block stored in the local ledger of C_r, and broadcasts it to the entire network. This design, compared to that of OmniLedger, incurs less overhead and allows a more frequent epoch reconfiguration to suit more highly adaptive adversaries.

4.4.6 Sharded Smart Contract

None of the considered sharding mechanism has achieved the smart-contract-oriented sharded so far except Chainspace introducing such functionality for the first time, by using a new transaction structure based on new atoms *Objects* denoted as o. *Objects* defines a series of new parameters including input objects $\mathbf{x}$, output objects $\mathbf{y}$, reference objects $\mathbf{r}$, contracts c as special objects, procedures *proc*, and checkers v. By following a serious of operations, the atomicity of smart contracts can be ensured in Chainspace.

Insight 17. *By modifying the transaction structure and involving the concept of the new atoms and objects, it can safely shard a smart contract with strong atomicity, but at the cost of considerable overhead and hence low throughput.*

4.4.7 Replay Attacks and Defenses against Cross-shard Protocols

As raised by [173] for the first time, the replay attacks and defenses against BFT-based cross-shard protocols have attracted increasing attention (i.e., Monoxide is Nakamoto-based and has a lock-free cross-shard protocol, thus immune to this kind of replay attacks). By utilizing the property of unanimous voting, the replay attacks strategy has the ability to compromise the *cross-shard atomicity*, and launch the double-spending attack with a low cost. Specifically, each shard participated in a cross-shard transaction needs to transmit its own decision (i.e., accepting/aborting the transaction) to the other participants, in order to lock/unlock the internal objects and thus guaranteeing the *cross-shard atomicity*. However, an effective replay

attack can be easily launched by conducting the following strategy. Here, we consider an attacker and a honest client who is about to sending a cross-shard transaction $T(x_1, x_2) \longrightarrow (y_1, y_2, y_3)$ where x_i represents the input objects managed by shard-i, and y_i represents the output objects managed by shard-i.

1. *Eliciting and invalidating the decision-message sent from shard-1*: The attacker races the client by sending a $T'(x_2, \cdots)$ to shard-2 so that the involved objects will be locked in shard-2[3]. The attacker quickly follows up by submitting T to shard-1 and shard-2. As soon as T reached shard-1, an $accept(T)$ is sent out and can be pre-recorded by the attacker. In contrast, T will be invalidated in shard-2. An $abort(T)$ will be sent out and pre-recorded by the attacker due to the locked objects of shard-2.

2. *Compromising the consistency*: At any time shard-1 is about to sending the decisions, the attacker can race shard-1 by broadcasting and replaying the pre-recorded message which always opposes to shard-1. As a result, the input objects of shard-1 is still active, while new output objects have been created in shard-2 and shard-3, i.e., the consistency of the system is compromised.

Authors of [173] also pointed out the reasons making the replay attacks possible. First (①), there lacks a way for the input shards to know the correspondence between a protocol message received (i.e., $accept(T)$ or $abort(T)$) and a specific transaction T. Second (②), there also lacks a way for the output shards to know the context of a specific transaction as they are, in fact, excluded from the intermediate processing.

To address the limitations, a modified version of Chainspace, Byzcuit [173], is proposed along with two new features. In regards to ①, a sequence number scheme

[3]The destination of T or T' is replaced by a self-driven client in OmniLedger as such a client is considered to be the handler to achieve the *cross-shard atomicity*.

is applied to each transaction to ensure the correspondence, while a dummy object of each output shard is added to the input field of a transaction (i.e., forcing the output shards to participate in the intermediate processing) in order to address ②.

Insight 18. *The proposed replay attacks and defenses against BFT-based cross-shard protocols is significant and worth more attraction. However, the sequence number scheme still has a synchronization issue, and the dummy object remains poor at the scalability, both of which strive for optimization.*

4.5 Challenges and Future Trend

We have elaborated on the designs and protocols of each considered sharding mechanisms, i.e., Monoxide, Elastico, OmniLedger, Rapidchain, Ethereum 2.0, and Chainspace, in terms of the *intra-consensus*, *cross-shard atomicity*, and *general improvements*, based on which a comprehensive comparison is presented in Tables 4.1 and 4.2.

We conclude that RapidChain and Ethereum 2.0 implement optimizations that reduce restrictions of Elastico and OmniLedger, which leads to RapidChain and Ethereum 2.0 being the most advanced BFT-based sharding mechanisms in terms of throughput and cost. On the other hand, Monoxide pushes the upper-bound of throughput to Mega level, and opens up a new direction of the Nakamoto-based sharding mechanisms. Chainspace has plenty of room for performance improvement for sharded-smart contract.

Furthermore, we point out the challenges remaining unsolved practically, as well as the future trend being discussed.

Table 4.1 : A comparison regarding the protocols (ranged from the settings of intra-consensus to the design of cross-shard atomicity, as well as the corresponding overhead) among the discussed sharding mechanisms in this chapter is elaborated.

			Monoxide	**Elastico**	**OmniLedger**	**RapidChain**	**Ethereum 2.0**	**Chainspace**
Network model			Partial-sync	Partial-sync	Partial-sync	**Intra** Sync; **Total** Partial-sync	Partial-sync	Partial-sync
Security model	**Threat model**		Attackers behave arbitrarily, Uncoordinated majority	Attackers behave arbitrarily, slowly adaptive	Attackers behave arbitrarily, slowly adaptive	Attackers behave arbitrarily, slowly adaptive	Attackers behave arbitrarily, Uncoordinated majority	Attackers behave arbitrarily, Uncoordinated majority
	FT	**Intra**	50%	33%	33%	50%	33%	33%
		Total	50%	25%	25%	33%	33%	25%
Intra-Consensus Protocol			PoW-based Chu-ko-nu mining	PBFT	ByzCoinX	50% BFT	BFT-based PoS	Mod-SMaRt implementation of PBFT
Randomness ($\mathcal{R}$)	**Existence**		No	Yes. $\mathcal{R}_{i+1}$ is generated by the final committee at the end of epoch i	Yes. $\mathcal{R}_{i+1}$ is generated by using RandHound + VRF in the beginning of epoch i+1	Yes. $\mathcal{R}_{i+1}$ is generated by the reference committee at the end of epoch i	Yes. Each $\mathcal{R}$ is generated by using RANDAO + VDF on the beacon chain	Unknown
	Use		N/A	1. The seed of PoW puzzle for the next epoch; 2. Select the intra-leader	1. Select the intra-leader and the sub-group; 2. Epoch reconfiguration; 3. trust-but-verify validation	1. The seed of PoW puzzle for the next epoch; 2. Select the intra-leader; 3. Bootstrapping; 4. Epoch reconfiguration	1. Select the proposer/attesters; 2. Select the validators for checkpointing from the global pool.	Unknown
Members	**Allocation**		One-off allocation based on the identity (address) of nodes	Allocation based on the least-significant bits of the result of PoW puzzles	Allocation based on $\mathcal{R}$	Allocation based on the result of PoW puzzles	Allocation based on $\mathcal{R}$	One-off allocation based on objects
	Safe Epoch reconfiguration		N/A	Unsafe	Yes, swapping-out bounded by 2/3 at a given time	Yes, swapping-out a constant number of node	Yes	N/A
Additional global Blockchain			Mixed targets: No; Identical targets: Yes	Yes, a global ledger	Yes, identity Blockchain	Yes, reference Blockchain	Yes, the mainnet and beacon chain	No
Transaction structure			Account	UTXO	UTXO	UTXO	Account	Object-driven, contract-sharded
Cross-shard Tx	**Support**		Yes	No	Yes	Yes	Yes	Yes
	Method		Async, Lock-free	N/A	Sync, Lock/Unlock	Sync, Lock/Unlock	Sync, Lock/Unlock	Sync, Lock/Unlock
Complexity	**Communication**		Mixed PoW targets: $O(m + n \log_2 n)$ Identical PoW targets: $O(m + n)$	$O(m^2 + n)$	$O(\log_2 m + n)$	$O(m^2 + m \log_2 n)$	$O(m^2 + n)$	$O(m^2 + n)$
	Storage		$\Omega(\lvert C\rvert) \sim O(\lvert C\rvert + n\lvert C_h\rvert)$	$O(n\lvert C\rvert)$	$O(\lvert C\rvert)$	$O(\lvert C\rvert + \lvert C_r\rvert)$	$\Omega(\lvert C\rvert + n\lvert H\rvert + \lvert C_g\rvert) \sim O(n\lvert C\rvert + \lvert C_g\rvert)$	$O(\lvert C\rvert + \lvert C_{nh}\rvert)$
Features and Restrictions			Insights 1, 9, 15	Insights 2, 3, 15	Insights 4, 5, 10, 11, 14, 16, 18	Insights 6, 12, 18	Insights 7, 8, 13, 15, 18	Insights 2, 11, 17, 18

Table 4.2 : A comparison among the discussed sharding mechanisms in this chapter is elaborated. The results of throughput and cost are shown in [205]. The latency is also obtained and shown (N/A: Not Available).

		Monoxide	**Elastico**	**OmniLedger**	**RapidChain**	**Ethereum 2.0**	**Chainspace**
Shards' settings	**Number of shards** (n)	$2^{10}\sim2^{18}$	$<10^2$	$<2^6$	$<2^8$	$<2^9$	$<10^2$
	Shard size (m)	$10^2\sim10^4$	$<10^2$	$2^2\sim2^{10}$	$(2^2 - 1)\sim2^8$	$<10^2$	$<10^2$
Epoch length		N/A	$\sim$10min	$\geq$one day	$\leq$one day	one week	Exists, details not provided
Latency	**Transaction confirmation**	23s	$<$900s	$\sim$100s	70s	6s$\sim$8s [153]	2s
	Epoch reconfiguration	N/A	N/A	1000s	200$\sim$350s	Unknown	Unknown
Upper-bound	**Improving factor** ($\mathcal{N}$)	n/2	n	1$\sim$n/2	n/2	n/3	1$\sim$n/2
	Throughput	1.23$\sim$2.56Mtps	48ktps	28.8ktps	128ktps	134ktps	$<$400tps
	Cost	30$\sim$80 USD/hour	30$\sim$35 USD/hour	0.2$\sim$0.3 USD/hour	0.2$\sim$0.3 USD/hour	0.4$\sim$0.45 USD/hour	N/A

4.5.0.1 Future Trend for Reducing the Overhead

Three common pitfalls in existing sharding mechanisms prevent the system from being horizontally scaled to the theoretical upper bound due to the communication and storage overhead.

- *An existing global chain that is needed to be stored by all participating miners/validators.* Such a global chain tends to be responsible for all global operations, such as generating randomness, cross-validating transactions in different shards, reshuffling operation. However, this simply poses the bottleneck threat (not only the performance bottleneck but also the security bottleneck) back to a single global chain, which is the root issue sharding technologies would have tried to solve. **Insight 15** and other (most recently proposed) sharding mechanisms hit this pitfall, e.g., SSChain [43] and Thinkey [45]. SSChain simply utilizes a two-layer architecture where a global chain is set to deal with all data migration and reshuffling operations. Thinkey also implements a root chain to achieve the cross-shard transactions and reshuffling operations. **Trend 1: Restricting the use of a global chain in any operations, and the bottleneck requiring to be solved if used.**

- *Requiring miners/validators to store ledgers from other shards.* This is necessary in some of the existing sharding mechanisms in order to cross-validating transactions and reshuffling operation. However, it leads to miners/validators incurring high communication and storage overhead in $O(n)$ (n is the number of shards). **Insights 1**, **7**, **9**, **10**, **11**, **13** hit this pitfall. **Trend 2: Balancing the storage and communication overhead for miners/validators in sending cross-shard transactions and reshuffling, so that the order can be lower than $O(n)$.** One of the potential solutions might be the fraud proof that enables light nodes to be as secure as full nodes without needing to

store the whole ledger [6], yet it has not been mature at the time of writing.

- *Allocating participating nodes to shards based on their business requirements in order to bypass the overhead of cross-shard communication.* Business-driven members allocation for shards has been proposed and discussed in some designs, e.g., Ethereum 2.0 [117][4] and VAPOR [160][5], in order to reduce, 1) the frequency that a participating node gets swapped out; and 2) the ratio of non-cross-shard transactions, for the ease of management and lower overhead. However, this results in a very long epoch reconfiguration for participating nodes and unevenly shard size, which ultimately poses a risk of crowed transactions to a single shard as time passes and the size and throughput increases, thus hitting the bottleneck of intra-consensus. **Trend 3: Avoiding simple business-driven members allocation that risks shards suffering from crowed transactions.**

4.5.0.2 Future Trend for Strengthening the Security and Atomicity

This trend corresponds to the intra-consensus and atomicity of cross-shard transactions, respectively. We point out the potential direction on more secure intra-consensus and more efficient cross-shard transactions, as shown in the following.

Intra-consensus:

- **Trend 4: Scaling the unbiased and unpredictable randomness generator in large-scale networks with as few third-party hardcoded**

[4] A possible design proposed by Ethereum 2.0 is to merge shards that interact more frequently than others.

[5] Another design proposed by VAPOR is to define a shard as a subset of nodes who care about some transactions. Transactions in VAPOR feature the ownership and record the nodes who have ever held the ownership.

settings as possible. The unbiased and unpredictable randomness plays an important role in BFT-based intra-consensus design. Improving this kind of algorithms can significantly prevent the validators from being under DDoS attacks. **Insights 3**, **5**, and **8** belong to this aspect.

- **Trend 5: Improving the PoW-based intra consensus, and generalizing it into other types of Nakamoto-based consensus algorithms.** Chu-ko-nu mining of Monoxide takes advantage of PoW to bypass the vortex of randomness, nevertheless, the security of which is dependent on the storage. As such, the future direction can be potentially decoupling the security and storage, and generalize the concept to other Nakamoto-based consensus algorithms, e.g., PoS.

- **Trend 6: Balancing the uses of stochastic and biased members allocation for shards.** All discussed sharding mechanisms (except Chainspace) use a stochastic allocation. A stochastic allocation is helpful to protect the shards from malicious biased allocations. On the other hand, new notions have been proposed to improve the scalability by taking advantage of a biased allocation. For example, [196] proposes a biased allocation to force the number of members an attacker can own within a single shard to be upper-bounded, in order to achieve a total FT of 50%. However, a vulnerability of this mechanism has been revealed that attackers can simply save the redundant resources from a specific shard, and has more sufficient resources to control more shards. Thus, a balance of this use still strives for a solution.

Efficient atomicity:

- **Trend 7: Enabling efficient conditional cross-shard transactions that enable contract-orient operations.** Only Chainspace and the future phase

of Ethereum 2.0 claim to support such conditional cross-shard transactions so far, but at the cost of unacceptable overhead and latency, which requires more focus in the future trend.

- **Trend 8: Combining sharding technologies with DAG being the data structure of any shards.** DAG provides more flexibility specifically in some IoT contexts, such as the micro-payment. However, the implications of the cross-shard transactions with DAG still require more focuses.

4.6 Conclusions

Remark that the lack of a generic and systematic framework to learn various sharding mechanisms occurs along with the sharding technology has taken centre stage of scaling Blockchains. This was resolved in this chapter by highlighting the importance of sharding for the design of *scale-out* Blockchains and systematizing the state-of-the-art sharding mechanisms in regards to the *intra-consensus security*, *atomicity of cross-shard transactions*, and *general challenges and improvements*. We also proposed our insights analyzing the features and restrictions, based on which a comprehensive comparison among the considered sharding mechanisms was obtained.

A list of the key observations and conclusions are as follows:

- For the first time Monoxide proposes a Nakamoto-based sharding mechanism, but at the cost of storing headers of all shards to guarantee the maximum intra-consensus-safety.

- The traditional PBFT used in Elastico and Chainspace does not guarantee the *intra-consensus-safety* due to its weak scalability, while the BFT-based sharding mechanisms, i.e., OmniLedger, Rapidchain, and Ethereum 2.0, improve

the *intra-consensus-safety* by scaling the traditional PBFT or increasing the fault tolerance of the traditional PBFT.

- The randomness generators of all considered sharding mechanisms in this chapter need strict network settings, otherwise the unpredictiability and unbiasability in scaled networks will be compromised.

- Monoxide, OmniLedger, Rapidchain, and Ethereum 2.0 all propose their own solution to the issue of cross-shard transactions, none of which can support cross-shard smart contracts. Only Chainspace proposes a smart-contract-oriented sharding mechanism, but at the cost of low throughput.

- All considered sharding mechanisms introduce the optimizations to address the new challenges their proposed sharding mechanisms pose to the system, e.g., latenc, storage, fair randomness generator, and replay-attack defense, but further improvements are necessary.

Chapter 5

Enabling Attribute Revocation for Fine-grained Access Control in Blockchain-IoT Systems

Regardless of the network size, Blockchain-based IoT networks become increasingly reliant on fine-grained access control due to the growing size of IoT data stream and the growing requirement of flexible data management. ABE has been considered appropriate to provide fine-grained access control in Blockchains that inherently lack such features. However, the adoption of ABE in Blockchains has been severely hindered by the incompatibility between the immutability of typical Blockchains and the attribute updates/revocations of ABE, which requires more attentions. Under this background, this chapter proposes a new Blockchain-based IoT system which is compatible with ABE technique, and fine-grained access control is implemented with the attribute update enabled by integrating CH algorithms into the Blockchains.

5.1 Introduction

The Blockchain technology, originating from cryptocurrency, has been recently employed in the IoT as the root of trust for authorization management [144], policy management [141], and data security [111]. In a Blockchain, all participants can verify and certify data by following a common consensus protocol to provide decentralized, reliable, and tamper-resistant services [188]. There have been a number of attempts to implement data access control on the traditional Blockchains where data are publicly stored, e.g., Bitcoin and Ethereum [135, 191]. Cryptographic technologies, e.g., homomorphic encryption and zero-knowledge-proof [116, 129], have been adopted. However, these technologies can suffer from prohibitive overhead

for IoT devices [197]. This prevents fine-grained access control in Blockchain-based IoT systems. Meanwhile, Hyperledger Fabric [85] implements a traditional access control by using transaction encryption and key management. However, the size of encrypted conversation keys grows linearly with the number of users. The overhead would still be very high in Blockchain for fine-grained access control.

The Attribute-based Encryption (ABE) technologies have the potential to achieve fine-grained access control where a ciphertext can only be decrypted when the attributes of a user match a predefined set of rules [185, 156]. In [156], the conversation keys of encrypted transaction data are protected with ABE by the transaction senders (i.e., data owners). Any outdated ciphertext needs to be updated and overwritten with an updated version when the attributes are revoked. However, the immutability of Blockchain prevents the update of attributes and the dynamic group membership of ABE. The existing solution tends to store the updated ciphertext of ABE by smart contracts [154]. A risk of a direct data leakage to revoked members arises. Adequate designs of ABE-based Blockchain access control have yet to be properly addressed.

In this chapter, we propose a novel multi-layer Blockchain-IoT data service system which enables secure attribute updates in an ABE-based fine-grained access control mechanism for Blockchains. We develop a redactable key chain along with a standard data chain to secure and control the access to the data chain. Empowered by redactable hash functions, the redactable key chain allows the access policies of ABE to be updated by key chain miners. The data chain can be any existing Blockchain with any scalable structure and preserves the immutability of the IoT data. Collectively, these crytographic primitives address the inherent incompatibility between the immutability of Blockchains and the indispensable need of updating attributes to manage the access to Blockchains. To the best of our knowledge, the proposed Blockchain-IoT data service system is the first of its kind to enable the

attribute updates and provide effective access control in the presence of the updates. The system is compatible with the major types of cryptographic primitives and consensus algorithms in Blockchains.

The key contributions of this chapter are shown as follows.

1. We identify the incompatibility between the immutability of Blockchain and the update of attributes in ABE, and the resultant potential risk of direct data leakage to revoked members. The incompatibility hinders the adoptions of ABE in Blockchain-based IoT systems.

2. We design a new multi-layer Blockchain structure consisting of a data chain and key chains, in order to decouple data storage and access control management. ABE is used for fine-grained access control in the proposed key chains.

3. We propose to integrate the CH algorithm in the Blockchain structure along with a new verification scheme to support the tamper-resistant attribute update of ABE in the proposed key chains. As a result, the system allows trusted policy update in ABE, while still preserving the tamper-resistance of Blockchains.

The analysis and simulation results show that our proposed mechanism is able to outperform existing solutions in terms of overhead, searching complexity, and security. Our mechanism also provides excellent compatibility with major types of consensus protocols, cross-chain protocols, and crypto algorithms (ABE and CH).

The rest of this chapter is organized as follows. Section 5.2 presents the proposed Blockchain system. Section 5.3 presents the multi-layer Blockchain design, as well as the design of redactable key chain with a new verification scheme. In Section 5.4, we conduct comprehensive system analysis and simulation. Section 5.5 concludes the chapter.

Table 5.1 : Notation Definition

Notation	**Definition**
DP	Data Publisher
DU	Data User
PT	Plaintext of a message
CT	Ciphertext of a message
$\mathbb{PT}$	Plaintext of a conversation key
$\mathbb{CT}$	Ciphertext of a conversation key
$\mathbb{E}$	Encrypting function to an encrypted message
$\mathbb{D}$	Decrypting function to a decrypted message
$\{\mathbf{A}\}$	Access policy that defines the access to conversation keys
$SK_{\mathbf{A},P}/PK_{\mathbf{A},P}$	Conversation keys that used to an encrypt/decrypt message P, with access policy assigned as $\{\mathbf{A}\}$
$\mathbb{ABE}_{\mathbf{A}}$	Attribute-based encryption algorithm to encrypt/decrypt conversation keys, with access policy assigned as $\{\mathbf{A}\}$
$ABE_{\alpha,SK,i}/ABE_{\alpha,PK}$	Private/Public key of the given $\mathbb{ABE}$, where the private key is identity-based for each member i in attribute α
$ABE_{\alpha,SK}$	An abbreviation of $ABE_{\alpha,SK,i}$ for simplicity
$\alpha \vDash \mathbf{A}$	An attribute α satisfies an access policy $\{\mathbf{A}\}$
$CH_{SK,P}/CH_{PK,P}$	Private/Public key of the CH algorithm with respect to a message P
$t_{i,j}$	j-th transaction on Block-i, i.e., the block with height i
$\bigsqcup_m^n$	A LOOP process starting from index m to index n

5.2 Preliminary

5.2.1 Attribute-based Encryption

ABE provides access control to data based on a set of rules associated with data and attributes of a data user. In general, an ABE scheme consists of the following five algorithms. Tab. 5.1 summarizes the notations used in the rest of the paper.

ABE_Setup $(1^k) \rightarrow (MK)$: *This algorithm sets up ABE. It takes as input a security parameter* 1^k*, and outputs a master key* MK.

ABE_KeyGen $(MK, r) \rightarrow (ABE_{SK,i}, ABE_{PK})$: *This algorithm generates attribute-based keys. It takes as input the master key* MK *and a random number* r*, and*

outputs a pair of ABE keys. Each pair of ABE key includes a private parameter set, $ABE_{SK,i}$, and a public parameter set, ABE_{PK}. Note that $ABE_{SK,i}$ is identity-based, which indicates that each member i assigned with the same attribute has its own unique $ABE_{SK,i}$.

ABE_Encrypt $(\mathbb{PT}, \mathbf{A}, ABE_{\alpha,PK}) \rightarrow (\mathbb{CT})$: *This algorithm encrypts data according to the assigned access policy. It takes as input a message to encrypt* $\mathbb{PT}$*, an access policy* $\mathbf{A}$ *and a set of public parameters* $ABE_{\alpha,PK}$ *of attribute* α*, and outputs* $\mathbb{CT}$ *which is the ciphertext of* $\mathbb{PT}$.

ABE_Decrypt $(\mathbb{CT}, \mathbf{A}, ABE_{\alpha,SK}) \rightarrow (\mathbb{PT})$: *This algorithm decrypts data if the attribute-related parameters satisfy the access policy. It takes as input a message to decrypt* $\mathbb{CT}$*, an access policy* $\mathbf{A}$ *and a set of private parameters* $ABE_{\alpha,SK}$ *by each member in attribute* α*, and outputs* $\mathbb{PT}$ *which is the plaintext of* $\mathbb{CT}$.

ABE_Update $(MK, \alpha, \mathbb{CT}, ABE_{\alpha,SK}, r') \rightarrow (\mathbb{CT}', ABE'_{\alpha,SK})$: *This algorithm updates the* $\mathbb{CT}$ *and the corresponding* $ABE_{\alpha,SK}$ *to a new version, i.e.,* $\mathbb{CT}'$ *and* $ABE'_{\alpha,SK}$*, respectively, to meet the update of attribute members. It takes as input the master key* MK*, an attribute* α*, the outdated* $\mathbb{CT}$*, the outdated* $ABE_{\alpha,SK}$*, and a new random number* r'*, and outputs the updated* $\mathbb{CT}'$ *and* $ABE'_{\alpha,SK}$.

5.2.2 Chameleon Hash (CH) Algorithm

A generic CH algorithm involves a trapdoor (private key) to allow one to quickly identify an arbitrary hash collision within the domain of this algorithm. It consists of the following three algorithms.

CH_KeyGen $(\lambda, Para_{CH}) \rightarrow (SK_{CH}, PK_{CH})$: *This algorithm generates CH keys, given security parameters. It takes as input a security parameter* λ *and a system parameter* $Para_{CH}$*; and outputs a private (trapdoor) key* SK_{CH} *and a public (hash) key* PK_{CH}.

CH_Hash $(M, PK_{CH}, r) \rightarrow (CH)$: *This algorithm generates CH value for a message. It takes as input a message to hash M, a private key PK_{CH} and a random number r; and outputs a CH value CH.*

CH_Update $(M, M', r, SK_{CH}) \rightarrow (r')$: *This algorithm updates CH-related parameters to guarantee that the original CH value remains valid after the message update. It takes as input a message before the update M, the message after the update M', the CH random number r before the update and the corresponding private key SK_{CH}; and outputs an updated CH random number r'.* The CH associated with the updated CH random number and the updated message is identical to the original CH value before an update, i.e., **CH_Hash** (M', PK_{CH}, r') = **CH_Hash** (M, PK_{CH}, r).

5.3 Multi-layer Blockchain-IoT System with Redacatble Key Chain

This paper proposes a novel multi-layer Blockchain-IoT data service system where secure attribute updates are enabled in an ABE-based fine-grained access control mechanism for Blockchains. We develop a redactable key chain to provide secure access to another standard data chain. The redactable key chain enables the tamper-resistant attribute updates of ABE via a new verification scheme conducted by the key chain miners. As a result, a secure fine-grained access control mechanism for Blockchain-IoT systems can be achieved.

5.3.1 System Overview

The proposed system uses Blockchain to provide IoT data storage and data sharing services where a CP-ABE scheme is integrated to enable fine-grained access control. The proposed system consists of the following four roles:

Trusted Authority (TA): The TA assigns and manages attributes for the entities

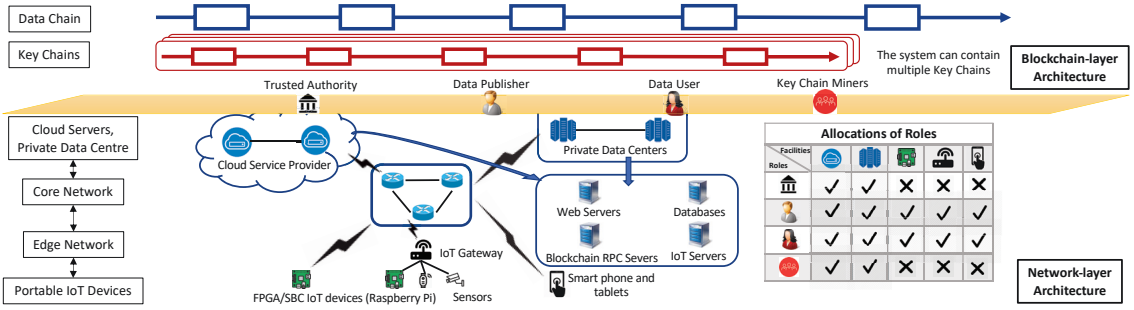


Figure 5.1 : This figure illustrates an overview of the proposed system. The bottom-half-side shows the architecture of the network-layer where the hierarchical topology of facilities is specifically described, upon which Blockchains being run among these facilities charged with different roles are described in the top-half-side. The table regarding the allocation of roles describes the relationship between the considered facilities and roles.

in the system. The accessibility of an entity to a particular piece of data depends on the attributes of the entity, as well as the access control policy of the data. The TA can be a standalone entity or a multi-authority group of which each member manages a set of attributes for robustness and scalability. In this paper, we consider the TA as a single entity.

Data Publisher (DP): The DPs, such as IoT devices, produce, packetize, and upload data to Blockchain in the form of transactions. The DPs also specify the access policy for each new published message, so that only the data users with attributes satisfying the access policy can access the message. *Data User (DU)*: The DUs access the Blockchain to retrieve data. Their attributes may change over time, and only the TA can update DUs' attributes and attribute-related private information.

Miners: A *Miner* is responsible for mining data into the Blockchain and providing tamper-resistance, as typically done in any existing Blockchain systems. A new responsibility of the *Miner*s is to conduct a new verification scheme to avoid abusing the CH updating function during the consensus process, as will be described

in Section 4.3.1. In the rest of this paper, *Miner*s refer to those running, managing, and storing the key chains. They are powerful servers typically located in either the cloud or private data centers.

Fig. 5.1 shows the network construction of a hierarchical IoT system, where these roles can be decoupled from the physical devices and one device may take multiple roles. For example, servers providing different services (e.g., Web servers, IoT servers, and Blockchain RPC servers) can take the role of DP or DU. The table at the right-bottom corner of Fig. 5.1 provides the mapping between network elements in the network architecture and the roles in the (key) Blockchains. As a result, every network element assigned with one or multiple roles, is connected via the network. Each of the roles can participate in the data exchange of Blockchains, i.e., uploading or fetching data from the *Data Chain* or *Key Chain*s.

An IoT device serves as a DP/DU (primarily DP) in the proposed protocol. It only uses its own conversation key SK_P/PK_P associated with message P, ABE public key ABE_{PK} and private key ABE_{SK} associated with each of its attributes, to encrypt its data before uploading the data to the Blockchains (or to decrypt its data after downloading the data from the Blockchains). Based on FPGA and System-on-Chip (SoC) boards such as AVNET's Zedboards or single-board computers (SBCs) such as Raspberry Pi, many current IoT devices are reasonably powerful to carry out encryption operations. These are the IoT devices considered in this paper. On the other hand, ABE_{PK} and ABE_{SK} are generated by the TA and maintained/stored in one of the key chains. The IoT devices only retrieve the keys from the key chain, and are not involved in running, managing, or storing any of the Blockchains (including the key chains and the data chain) due to limited memory and communication resources of the devices.

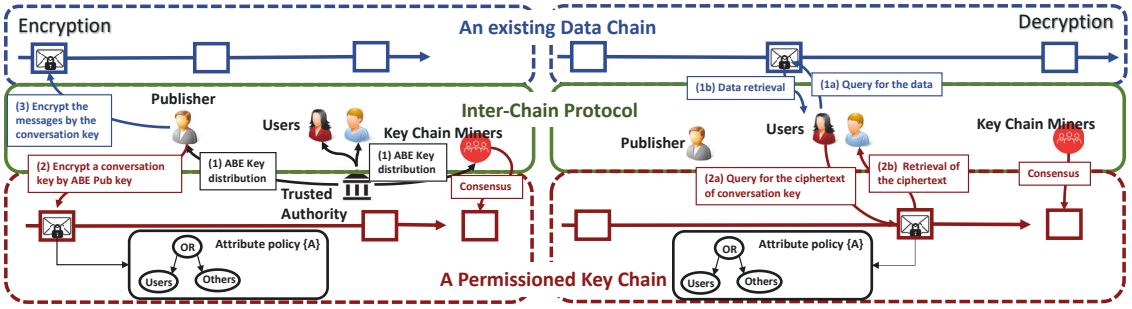


Figure 5.2 : The process of encrypting and publishing a new message is shown on the left-hand side. The right-hand side shows the process of decrypting and retrieving a new published message. A Data Publisher (DP) and Data Users (DUs) implement the Inter-chain Protocol to deal with the data encryption/decryption between the *Data Chain* and *Key Chain*.

5.3.2 The Proposed Multi-layer Blockchain System

We propose a multi-layer Blockchain-based data ledger service system consisting of two types of Blockchains, *Data Chain* for immutable confidential data service, and *Key Chain* for key management, as shown in Fig. 5.2.

Data Chain - A *Data Chain* can be any existing permissionless or permissioned Blockchains used for recording data, e.g., Ethereum [191], with the following scalability-enhancing techniques:

- existing scalable consensus algorithms or data structures, e.g., ByzCoin [98] and Directed-acyclic-Graph [152], that can help reduce the communication overhead of the proposed system to at most $O(n)$, where n is the network size;
- existing *scale-out* techniques that can partition a network (including the proposed network) into shards from the perspective of data storage, communication bandwidth, and computation, to achieve horizontal scalability (sharding) to accommodate huge amounts of data, e.g., OmniLedger [99].

By using these techniques, the total capacity and throughput of a scalable *Data Chain* can be expected to support the huge data of IoT systems. A DP encrypts

its data with conversation keys before publishing encrypted data in the form of transactions in the *Data Chain*.

Key Chain - A *Key Chain* stores the ciphertext of the conversation keys, i.e., $\mathbb{CT}$, for key management. The *Key Chain* is permissioned and has a unique TA to manage the key generation and distribution of the ABE scheme. Here, the conversation keys, used to encrypt data in the *Data Chain*, are encrypted by the ABE algorithm by the DP, before the DP publishes the encrypted data in the *Data Chain*. Every *Key Chain* is redactable along with a new verification scheme conducted by the *Miners* on the *Key Chain* to guarantee the tamper-resistance. Thus, the attribute updates of ABE can be enabled.

Inter-Chain Protocol - This protocol is implemented for data exchange among a *Data Chain* and *Key Chains*[1]. Any participants acting as clients on both *Data Chain* and *Key Chains* operate the new the **Inter-Chain Protocol**. The protocol includes the following subprotocols.

Encryption and publishing a new message P: A DP encrypts $PK_{\mathbf{A},P}$ to $\mathbb{CT} = \mathbb{ABE}_{\mathbf{A}}(PK_{\mathbf{A},P})$, by using $ABE_{\alpha,PK}$ where an attribute α satisfies an attribute policy $\{\mathbf{A}\}$. This $\mathbb{CT}$ is uploaded to a *Key Chain* as a transaction, $t_{i,j}$ (j-th transaction of the block with height i). After that, the DP encrypts the message P into CT by using the corresponding $SK_{\mathbf{A},P}$. As such, the indexes to $t_{i,j}$, i.e., i and j, are uploaded along with CT on a *Data Chain* for the cross-reference in the **Decryption** subprotocol. The details are provided in Section 5.3.4.1.

Decryption and retrieving a new message P: A DU assigned with the attribute α satisfying $\{\mathbf{A}\}$ intends to retrieve the message P, and the DU identifies the block

[1]A major type of cross-chain protocol [36] can be rather implemented to bridge the communication among *Data Chain* and *Key Chain* instead of making DPs send the transactions to each Blockchain by themselves. Remark that our system is a generic one and independent of the cross-chain protocols.

containing the related CT and the indexes, i and j on the *Data Chain*. The DU then searches for the corresponding $t_{i,j}$ on the *Key Chain*. Next, the DU can retrieve the $PK_{\mathbf{A},P}$ by decrypting the $\mathbb{CT}$ with its own $ABE_{\alpha,SK}$. As such, the DU can decrypt CT on the *Data Chain* with $PK_{\mathbf{A},P}$. The details are provided in Section 5.3.4.2.

Attribute updates by updating blocks on a Key Chain*:* The non-revoked (this means the DUs that are reserved after the corresponding attribute changes) and new DUs assigned with the attribute α satisfying $\{\mathbf{A}\}$ update the outdated $ABE_{\alpha,SK}$. Meanwhile, the miners of the *Key Chain* update the outdated $\mathbb{CT}$. The revoked DUs cannot directly retrieve $PK_{\mathbf{A},P}$ from the *Key Chain*, whilst the non-revoked and new DUs can access any $\mathbb{CT}$ assigned with the attribute policy $\{\mathbf{A}\}$. The details are provided in Section 5.3.4.3.

The proposed system can have a *Data Chain* and multiple *Key Chains* to support different and independent access policies to the *Data Chain*, as shown in Fig. 5.1. The *Data Chain* ensures the data integrity and transparency among multiple communities, and all communities can contribute to maintaining the *Data Chain*. The multiple *Key Chains* allow each community to manage their own private keys independently, using their own *Key Chain* to meet different requirements. Attributes are managed independently from other *Key Chains*. This is useful because two *Key Chains*, for example, may be managed by two communities that have similar business requirements. This implies that these two communities may use identical attributes expected to be mutually privacy-reserved on their own *Key Chain*, thus leading to the design of a cluster of *Key Chains* for flexible management of IoT.

The *Data Chain* and *Key Chains* collaborate to provide the following properties.

Fine-grained access control: An ABE scheme integrated with the proposed multi-layer Blockchain system guarantees the secured, fine-grained access control. Take the access control of a message M as an example. M is encrypted by a conver-

sation key before stored in the *Data Chain*. The conversation key is a prerequisite to access M. Only the DUs satisfying the access policy of the conversation key can retrieve the key which is stored in the *Key Chains* and encrypted by the ABE algorithm.

Revocable attribute of ABE: The editability of a *Key Chain* enables the support of attribute revocability of ABE in the *Key Chain*. All related conversation keys remain unchanged, while the corresponding ciphertext stored in the *Key Chain* is updated if an attribute of a member is revoked. Only DUs with attributes satisfying the attribute policy are able to decrypt the conversation keys and, in turn, the data. As a result, the *Data Chain* is unaffected by the attributes update, while the *Key Chain*s adopt CH to run as redactable Blockchains to embrace the addition and revocation of attributes. The design of the redactable *Key Chains* and the details of attribute updates will be given in Sections 5.3.3 and 5.3.4.3, respectively.

Anti-tampering: The tamper-resistance of both the *Data Chain* and *Key Chains* is inherited from the Blockchain. In the case with the member update supported, the *Data Chain* is unaffected, as discussed earlier. The *Key Chain* can also guarantee the tamper-resistance by conducting a new verification scheme with the editability of the CH algorithm, as will be given in Section 5.3.3.1. Thus, the secure update of attributes of ABE can be ensured on the *Key Chain*.

5.3.3 New Design of Redactable Key Chain

As mentioned in Section 5.3.2, the key to achieving the revocable attributes of ABE and anti-tampering is the key management in the design of *Key Chain*s. In this section, the design of a single redactable *Key Chain* is presented for illustration conveniences, such as the block structure, and the new verification scheme to ensure the tamper-resistance of the *Key Chain*.

5.3.3.1 Block Structure

Different from the existing Blockchain such as Bitcoin, our *Key Chain* is designed to be a redactable Blockchain to address the conflict between the immunity of Blockchains and attribute updates of an ABE scheme. We propose to use a CH algorithm, instead of traditional collision-free hash algorithms, to hash the data field of the blocks. The CH allows one to easily find an arbitrary hash collision regarding a specific hash value with a trapdoor, i.e., CH_{SK} [39], and has been used for removing sensitive information from Blockchains [16]. The CH can preserve the editability of the *Key Chain*, while allowing the *Key Chain* to be updated in response to additions and revocations of members and attributes. Here, (CH_{SK}, CH_{PK}) is short for $(CH_{SK,P}, CH_{PK,P})$.

Each pre-defined *Miner* is assigned with a CH_{SK}, as given by,

$$PK_{\mathbf{Miner},P} = \mathbf{ABE\_Decrypt}(\mathbb{CT}, \mathbf{Miner}, ABE_{Miner,SK}), \tag{5.1a}$$

$$CH_{SK} = \mathbb{D}_{PK_{\mathbf{Miner},P}}(\mathbb{E}_{SK_{\mathbf{Miner},P}}(CH_{SK})), \tag{5.1b}$$

where $\mathbb{CT} = \mathbb{ABE}_{\mathbf{Miner}}(PK_{\mathbf{Miner},P})$ on the *Key Chain*. Each block has a data field containing the $\mathbb{CT}$ and a random number in the block body. Any *Miner* which holds CH_{SK} is able to edit the data field of the blocks, while the CH value of the transaction remains unchanged. In this way, the chained structure of the *Key Chain* is maintained. Only a node of the *Key Chain* possessing CH_{SK} can edit blocks.

As shown on Fig. 5.3, each block of a *Key Chain* has a block header and a block body. Take Block-h, i.e., the block at height-h, for an example. The block header contains the following four key fields. The first two fields, $c(h)$, i.e, the Consensus Info of the consensus algorithm, and $H_p(h)$, i.e., Parent Hash accounting for the linked structure, are inherited from the traditional Blockchain structure. $n(h)$ and $MPT(h)$ are new in the *Key Chain* for security consideration to monitor and prevent any unauthorized Blockchain updates.

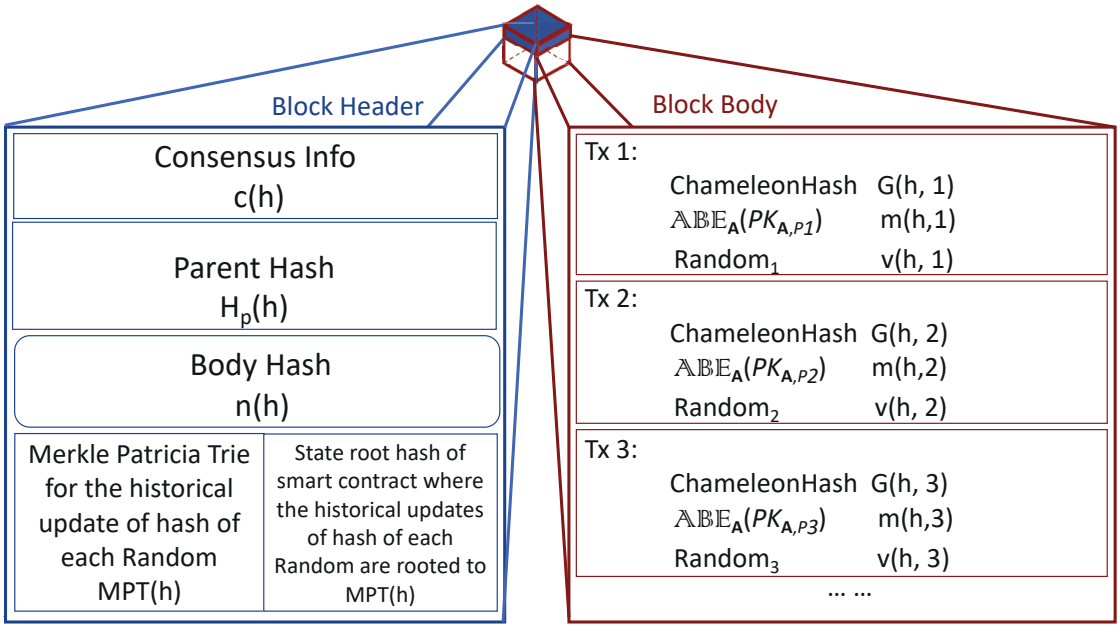


Figure 5.3 : The block structure of a block in a *Key Chain* that stores redactable messages

Body Hash $n(h)$: This field can be regarded as the hash value of the block body, i.e., the right-hand side of Fig. 5.3, except for the message itself and the corresponding random number. That is,

$$n(h) = Hash(G(h,1), G(h,2), G(h,3), ...), \tag{5.2}$$

where $n(h)$ remains unchanged given that $G(\cdot)$ is unchanged, even in the presence of an update of the data field and the related random number. $G(\cdot)$ is the CH value of the transaction. An optional field containing all other information related to the transactions, such as event logs or receipts defined in Ethereum-like Blockchain, can also be included in the Body Hash [191].

Updating Log $MPT(h)$: It denotes the state of the hash of every single random number in each transaction of the block body, i.e., $v(i,j)$, all through the LOOP process defined in Table. 5.1, i.e., $\bigsqcup_i^h \bigsqcup_j (t_{i,j})$ on Block-h (the latest state). This describes a process that,

1. any j-th transaction starting from the first to the last of a Block-i is traversed;

2. on top of 1), any Block-i starting from the first to Block-h is traversed.

The immutability of $MPT(h)$ is a clear indicator of the effectiveness of the proposed verification scheme in terms of overhead and search complexity, as shown in Section 5.3.3.2 and the simulation result in Section 5.4.1. $MPT(h)$ can prevent a malicious *Miner* from conducting a false update, which can be an outdated content intentionally preserved without breaking the linked structure because of the property of the CH algorithm (the users cannot distinguish any change of ciphertext by just looking at the unchanged $G(\cdot)$). Otherwise, the revoked DUs from an attribute α can directly access the outdated $\mathbb{CT}$ with their outdated $ABE_{\alpha,SK}$ and retrieve the conversation key PK.

There is no restriction on the data structure of $MPT(h)$, nevertheless, it contains the following rule formulated in (5.3):

$$mapping[i][j] \Longrightarrow Hash(v(i,j)), \tag{5.3}$$

which indicates that a hash value of a specific $v(i,j)$ should be traceable given the unique block height i and the transaction index j of Block-i, when the verification scheme of $MPT(h)$ is conducted, as shown in Fig. 5.4. Here, storing the state of $Hash(v(i,j))$ prevents the outdated $v(i,j)$ from being directly exposed to revoked members on the *Key Chain*.

The following structures are provided as possible options for $MPT(h)$:

- A variable stored in a smart contract through which the state root of the world state, in the format of a Merkle Patricia Tree (MPT) used in Ethereum, can notify the world state of the hash of every single random number [191];
- A new dedicated MPT in addition to the world state MPT;
- An engraved transaction in the block body that is immutable and engraved inside Block-h and all blocks beyond.

In the case where the world state MPT or the new dedicated MPT is used, the values stored in the leaves of the tree constitute the values of their parents and ancestors, i.e., up to the value of the state root. The proposed verification scheme can be achieved due to the fact that any updates at an individual leaf, i.e., the value of $Hash(v(i,j))$, would change the value of the state root stored in the block header; see more details of the MPT in [191].

As shown on the right-hand side of Fig. 5.3, the block body contains several redactable transactions which store encrypted conversation keys[2]. A redactable transaction has the following fields. Here a transaction $t_{i,j}$, i.e., the j-th transaction in the body of Block-i, is taken for an example.

- ABE Encrypted Key $m(i,j)$: This field stores the latest encrypted conversation key $\mathbb{ABE}_{\mathbf{A}}(PK)$. Each transaction only records one encrypted conversation key. This field is not required when calculating $n(i)$ of the block header.
- Random Number $v(i,j)$: This field stores the latest random number used for calculating the CH value based on $m(i,j)$. This field is also not used when calculating $n(i)$ in the block header.
- The CH Value $G(i,j)$: It is the CH value of the transaction. This field remains unchanged as it is involved in the calculation of $n(i)$ in the block header.

Here, $m(i,j)$ and $v(i,j)$ can be jointly edited without changing their CH value $G(i,j)$, by following the algorithm **CH_Update** defined in Section 5.2.2. For example, we assume $m(i,j)$ in the transaction $\big(m(i,j), v(i,j), G(i,j)\big)$ changes to $m'(i,j)$,

[2]In the case that the data structure of the engraved transaction is chosen for $MPT(h)$, the immutable engraved transactions used for recording the update history are contained in the block body.

CH_Update outputs $v'(i,j)$ which satisfies:

$$\begin{aligned} & \textbf{CH\_Hash} \quad \big(m'(i,j), v'(i,j)\big) \\ = \ & \textbf{CH\_Hash} \quad \big(m(i,j), v(i,j)\big) \\ = \ & G(i,j). \end{aligned} \tag{5.4}$$

The design in (5.4) contributes to the efficient and secure editability of the *Key Chain*. On the one hand, the design saves computation overhead by keeping only a single edited block valid in the presence of a change of $m(i,j)$ and $v(i,j)$. This is because $G(i,j)$ is the only field in the block body that is taken to validate the *Key Chain*. This allows $G(i,j)$ not to be changed even with the update of $m(i,j)$ and $v(i,j)$, leading to an unchanged $n(h)$ in the block header. In other words, a valid block with an immutable block header can still be validated, as long as $m(i,j)$ and $v(i,j)$ are edited based on the algorithm **CH_Update** to satisfy the property of $G(i,j)$, as shown in (5.4). On the other hand, such editability does not breach the tamper-resistance of the *Key Chain*. One reason is that the editability of the block body is secured by CH_{SK} (trapdoors) which are an indispensable input to **CH_Update**. Another reason is that all fields in the block header of the *Key Chain* are immutable to provide high tamper-resistance. Specifically, $c(h)$ and $H_p(h)$ jointly guarantee that the Blockchain is robust to block tampering. $MPT(h)$ protects the historical path of redactable transaction updating, and prevents malicious *Miner*s from exposing the outdated version of $\mathbb{ABE}_{\mathbf{A}}(PK)$ by an intentional false update.

5.3.3.2 Block Generation and Verification

With the transaction and block form defined above, the DPs of the permissioned *Key Chain* generate and broadcast transactions in the *Key Chain*. In contrast, the *Miner*s reach consensus on generating blocks based on a collection of pending transactions.

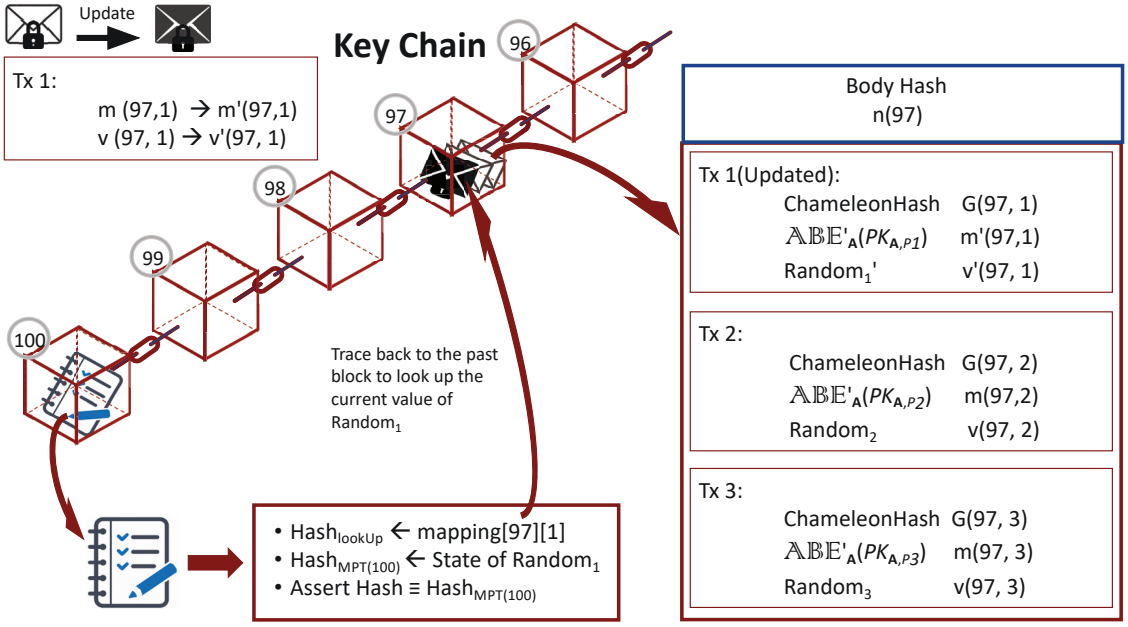


Figure 5.4 : The scheme of $MPT(h)$ delivers reliable verification for every editing conducted at block height-h, by maintaining an additional MPT dedicated for the logs.

Block verification is required when mining blocks and retrieving blocks from the *Miner*s. Block-h in a *Key Chain* is valid if and only if the following three conditions are all satisfied:

Verify $\left(MPT(h),\ t_{i,j}\right)$: This function implements a verification mechanism where an additional pointer links to the past block headers in addition to $H_p(h)$. Here, $H_p(h)$ denotes the hash of the parent block header, which provides a pointer linking to the past generation. In contrast, $MPT(h)$ provides a pointer linking to an arbitrary ancestor block headers. Based on $MPT(h)$, the latest state of $Hash(v(i,j))$, provided in (5.3), is matched with the hash value of the random number contained in $t_{i,j}$, as stated at the bottom in Fig. 5.4. As long as $\mathbb{CT} = \mathbb{ABE}_{\mathbf{A}}(PK)$, the *Key Chain* is updated and edited (the first redactable transaction turning to black color and getting updated from the outdated version, $m(97,1)$ to the latest version, $m'(97,1)$, as shown in the top left corner of Fig. 5.4), its related random number $v(97,1)$ is also updated to the latest version, $v'(97,1)$.

While $G(97,1)$, denoting the CH value of the first redactable transaction in Block-97, remains unchanged as described in (5.4). This also contributes to an unchanged $n(97,1)$, as shown in the block header on the right-hand side of Fig. 5.4.

Verify $\big(h_p(h)\big)$ and **Verify** $\big(c(h)\big)$: These function verify the linked structure via $h_p(h)$ and the consensus requirement via $c(h)$, respectively. They are standard to existing Blockchain systems [135, 191].

In general, the *Miner*s owning the CH_{SK} on a *Key Chain* can update the transaction $t_{i,j}$ containing $m(i,j)$ and $v(i,j)$, and append a new Block-h where $Hash(v(i,j))$ can be traceable via $MPT(h)$. This can ensure the update of $\mathbb{CT} = \mathbb{ABE}_{\mathbf{A}}(PK)$ without corrupting the linked structure. Note that $m(i,j)$, i.e., the data field of $t_{i,j}$, denotes $\mathbb{ABE}_{\mathbf{A}_m}\{PK_{\mathbf{A}_m,P_n}\}$ where $\mathbf{A}_m$ denotes the m-th access policy and P_n denotes the n-th message. $MPT(h)$ records the state of the hash of $v(i,j)$ on Block-h associated with each individual update of $(\mathbb{ABE}_{\mathbf{A}_M}\{PK_{\mathbf{A}_M,P_N}\})_{\mathbf{A}_0,P_0}^{\mathbf{A}_m,P_n}$, i.e., all $\mathbb{ABE}_{\mathbf{A}_M}\{PK_{\mathbf{A}_M,P_N}\}$ from $\mathbf{A}_0$ to $\mathbf{A}_m$ with messages from P_0 to P_n.

5.3.4 New Design of Inter-Chain Protocol

This section elaborates on the design of our proposed Inter-Chain Protocol. Three subprotocols are discussed separately in Sections 5.3.4.1 to 5.3.4.3. Algo. 1 presents the whole process of data sharing in our system regarding the encryption and publishing a new message conducted by a DP, and the decryption and retrieving the new message conducted by entitled DUs. Therein, Algo. 2 describes the process of the encryption and decryption used in Algos. 1 and 3. Algo. 3 presents the complete process of updating an attribute (both adding and revoking) by updating blocks on a *Key Chain*.

5.3.4.1 Protocol of Message Encryption and Publication

We start with the **Encryption** subprotocol of the **Inter-Chain Protocol**, with details in the following (referring to Lines 4-5 of Algo. 1 and Steps (1)-(3) of Fig. 5.2 on the left-hand side, denoted as Fig. 5.2-L),

Step 1: Initialization by TA. The unique TA of a *Key Chain* takes responsibility for initializing the system, as shown in Line 1 of Algo. 1 and Step (1) of Fig. 5.2-L.

1. *User Register and Attribute Initialization:* Entities register at the TA, after which the TA allocates attributes for each entity as a DU.

2. *ABE Initialization:* The TA generates ABE parameters for each attribute α with the Algorithm **ABE_KeyGen**, as defined in Section 5.2.1.

3. *CH Function Initialization:* The TA generates the parameters of a chosen CH algorithm according to **CH_KeyGen**, as defined in Section 5.2.2.

4. *Key Distribution:* The TA distributes the private ABE parameters $ABE_{\alpha,SK}$ in a secure channel to the corresponding DUs. Recall that $ABE_{\alpha,SK}$ is identity-based, i.e., each member of α has its own unique $ABE_{\alpha,SK}$. In the meantime, the TA broadcasts the public ABE parameters $ABE_{\alpha,PK}, \forall\alpha$ to the network, as shown in Line 2 of Algo. 1.

Step 2: Publish Data by DP. $\mathbb{E}_{SK_{\mathbf{Miner},P}}(CH_{SK})$ assigned to the DUs owning an attribute $Miner \vDash \{\mathbf{Miner}\}$ (attribute $Miner$ satisfies the access policy $\{\mathbf{Miner}\}$) is published on the *Data Chain* by the DP, prior to publishing a new message. Thus, only the *Miner*s of the *Key Chain* are entitled to the update process. Every new message is assigned with such a pair of CH key pair (CH_{SK}, CH_{PK}) regarding a chosen CH algorithm, where $\mathbb{E}_{SK_{\mathbf{Miner},P}}(CH_{SK})$ is stored in the *Data Chain* and CH_{PK} is publicly broadcast, and $\mathbb{ABE}_{\mathbf{Miner}}(PK_{\mathbf{Miner},P})$ is stored in the *Key Chain*.

Then a DP publishing a new message PT is subject to the following procedures to generate two transactions, one in the *Data Chain* and the other in the *Key Chain*; see Steps (2) and (3) of Fig. 5.2-L.

1. *Define Access Policy:* The DP defines the access policy, $\{\mathbf{A}\}$, to its data PT, i.e., PT is only accessible to the DUs whose attributes satisfy the access policy $\{\mathbf{A}\}$.

2. *Generate Conversation Key:* The DP generates[3] a pair of conversation keys $(SK_{\mathbf{A},P},PK_{\mathbf{A},P})$[4] for the access policy $\{\mathbf{A}\}$ DP also generates a pair of conversation key $(SK_{\mathbf{Miner},P},PK_{\mathbf{Miner},P})$ to publish the CH private key CH_{SK} based on the $\{\mathbf{Miner}\}$; see Line 3 of Algo. 1. Here, (SK, PK) is short for $(SK_{\mathbf{A},P}, PK_{\mathbf{A},P})$ in the rest of the paper for simplicity.

3. *Encrypt Conversation Key:* The DP uses the ABE algorithm **ABE_Encrypt** to encrypt the conversation (public) key PK to $\mathbb{CT} = \mathbb{ABE}_{\mathbf{A}}(PK)$. Here, $\mathbb{ABE}_{\mathbf{A}}(PK)$ denotes the ABE ciphertext of PK with its access policy $\{\mathbf{A}\}$. The corresponding attribute scheme is shown on the bottom-left of Fig. 5.2-L, where an attribute $User \subset \alpha$, $User \vDash \mathbf{A}$ exists.

4. *Generate Transaction for Conversation Key:* The DP generates and uploads a transaction to the *Key Chain*; see Line 1 of Algo. 2. Here, the transaction contains $\mathbb{CT} = \mathbb{ABE}_{\mathbf{A}}(PK)$, and the corresponding CH value and random

[3]The key generation depends on the data encryption algorithm, which is chosen according to specific applications and beyond the scope of this paper.

[4]In fact, the conversation keys, $(SK_{\mathbf{A},P}, PK_{\mathbf{A},P})$, can be a symmetric or an asymmetric key pair, depending on the chosen encryption algorithm. To simplify the illustration, we only discuss the case of asymmetric encryption algorithm and conversation keys in the rest of this paper. In the case of symmetric keys, the encryption key $SK_{\mathbf{A},P}$ and the decryption key $PK_{\mathbf{A},P}$ are identical, i.e., $SK_{\mathbf{A},P} = PK_{\mathbf{A},P}$.

number. This transaction is denoted as $t_{i,j}$ after the successful mining of the transaction on the *Key Chain*.

5. *Encrypt Data:* The DP encrypts PT to the ciphertext CT by using the conversation (private) key SK, and then $CT = \mathbb{E}_{SK}(PT)$. Here, $\mathbb{E}$ denotes the encryption operation of the chosen encryption algorithm.

6. *Generate Transaction for Data:* The DP generates a transaction based on the encrypted message CT and uploads to the *Data Chain*. This transaction, t_P, contains the indexes i and j of $t_{i,j}$ to its encrypted conversation key $\mathbb{ABE}_{\mathbf{A}}\{PK\}$, i.e., the j-th transaction of Block-i on the *Key Chain* where $\mathbb{ABE}_{\mathbf{A}}\{PK\}$ is stored, in order to identify the location of $\mathbb{CT}$ in the *Key Chain*. As such, a private message can be successfully published after mining in the *Data Chain*.

5.3.4.2 Protocol of Message Decryption and Retrieval

We proceed with the **Decryption** subprotocol of the **Inter-Chain Protocol**.

This subprotocol only contains one step, *Access to Data by DUs*. In the case that a DU/*Miner* is to obtain the message PT in the transaction t_P, two procedures take place to retrieve the decryption key PK by decrypting the $\mathbb{CT}$ in a *Key Chain*; also see Lines 6-7 of Algo. 1 and Steps (1)-(3) of the right-hand side of Fig. 5.2, denoted as Fig. 5.2-R.

1. *Retrieve Conversation Key from Key Chain:* As mentioned in Steps 2-(2) to 2-(6) in Section 5.3.4.1, t_P of the *Data Chain* stores the index to its encrypted conversation key in the *Key Chain*. The DU first retrieves the indexes i and j from transaction t_P, and then obtains the latest $\mathbb{ABE}_{\mathbf{A}}\{PK\}$ in $t_{i,j}$ from the *Key Chain*. If the attributes of the DUs satisfy the access policy $\{\mathbf{A}\}$, e.g., *Users* in Fig. 5.2-R, the DUs can retrieve the conversation key PK by

decrypting $\mathbb{ABE}_{\mathbf{A}}\{PK\}$ with the corresponding private key $ABE_{User,SK}$. Here the decryption algorithm refers to **ABE_Decrypt**; also see Line 9 of Algo. 2.

2. *Retrieve Data from Data Chain:* By using the conversation key PK and the encrypted message CT in transaction t_P, the DU can decrypt CT to $PT = \mathbb{D}_{PK}(CT)$, referring Line 10 of Algo. 2. Herein, $\mathbb{D}$ is the decrypt algorithm, corresponding to the encrypting algorithm $\mathbb{E}$.

Algorithm 1: The whole process of data sharing

▷ **Define**

Receiver ⇐ Sender.***Send***(Message); `// This transmission is secured by any private and encrypted channels, e.g. TLS.`

▷ **Initialization Phase**

1 $(ABE^{n,SK}_{m,SK}, ABE^{n,PK}_{m,PK})$ TA.***Initialize***(); `//` TA `initializes the system and generates ABE key pairs from attribute` m `to` n.

2 *Public Network* ⇐ TA.***Send***$(ABE^{n,PK}_{m,PK})$ **and** $Nodes^n_m$ ⇐ TA.***Send***$(ABE^{n,SK}_{m,SK})$; `//` $Nodes^n_m$ `indicates nodes that have attribute` m `to` n`, respectively.`

3 $(SK_{\mathbf{Miner},P}, PK_{\mathbf{Miner},P})$, $(CH_{SK,P}, CH_{PK,P})$, $(SK_{\mathbf{User},P}, PK_{\mathbf{User},P})$ ←DP.***Initialize***(); `//` A DP planning to publish a message P generates three key pairs, i.e., $(SK_{\mathtt{Miner},P}, PK_{\mathtt{Miner},P})$ $(CH_{SK,P}, CH_{PK,P})$ and $(SK_{\mathtt{User},P}, PK_{\mathtt{User},P})$, where {Miner} and {User} are the corresponding attribute policies. Meanwhile, $CH_{PK,P}$ is broadcast to the network.

▷ **Encryption Phase**

4 DP.***Encryption***$(SK_{\mathbf{Miner},P}, CH_{SK,P}, ABE_{Miner,PK}, PK_{\mathbf{Miner},P})$ `//` $Miner$ `is an attribute belonging to attributes set ranging from` m `to` n.

5 DP.***Encryption***$(SK_{\mathbf{User},P}, PT, ABE_{User,PK}, PK_{\mathbf{User},P})$; `//` $User$ `is an attribute belonging to attributes set ranging from` m `to` n.

▷ **Decryption Phase**

6 PT ***Decryption***$(ABE_{User,SK}, t_{i,j})$; `//` A DU satisfying {User} retrieves the indexes i and j stored in a block in the *Data Chain*. After that, the DU retrieves PT with $t_{i,j}$, where $t_{i,j}$ denotes the j-th transaction of Block-i in the *Key Chain*.

7 $CH_{SK,P}$ ***Decryption***$(ABE_{Miner,SK}, t_{i,j})$; `//` A *Miner* of the *Key Chain* satisfying {Miner} retrieves the indexes i and j stored in a block in the *Data Chain*. After that, the *Miner* retrieves $CH_{SK,P}$ with $t_{i,j}$.

▷ **Notice**

8 ***Encryption()*** and ***Decryption()*** are shown in Algo. 2.

9 $ABE^{n,SK}_{m,SK}$ during this process is identity-based, which indicates that each member has a particular attribute m has its own $ABE_{m,SK}$.

Algorithm 2: The encryption process ***Encryption()*** and the decryption process ***Decryption()***

▷ ***Encryption()***

Input: SK, PT of the message, ABE_{PK}, PK

Output: A new block appended to the *Data Chain* and the *Key Chain*, respectively

1 Block$_{Key,Pending}$ ***SendChameleonTx***(ABE_{PK}.***Encrypt***(PK), CH_{PK}); `// The DP encrypts` PK `by using the` ABE_{PK}`, i.e.,` $\mathbb{CT}$`, and Chameleon-hashes the ciphertext by using the` CH_{PK} `prior to inserting it into a new pending block of the` *Key Chain*.

2 **while** Consensus **do**

3 *Miner*.***Verify***(Block$_{Key,Pending}$);

4 **if** *Block*$_{Key,Pending}$ *is invalid* **then**

5 **alarm and exit**

6 *Key Chain*, i, j Block$_{Key,Pending}$; `// Append the new block to the` *Key Chain*`, obtain the index` j `of the transaction in Block-`i.

7 *Data Chain* ← Block$_{Data,Pending}$ ← ***SendNormalTx***(SK.***Encrypt***(PT), i, j); `// The DP encrypts` PT `by using the` SK`, and hashes the ciphertext prior to inserting into a new pending block of` *Data Chain*`, along with the corresponding` i `and` j.

▷ ***Decryption()***

Input: CT, ABE_{SK}, $t_{i,j}$

Output: PT of message

8 $\mathbb{CT}$ ***Find***($t_{i,j}$); `// The DU retrieves` $\mathbb{CT}$ `with given` $t_{i,j}$ `in the` *Key Chain*.

9 PK ABE_{SK}.***Decrypt***($\mathbb{CT}$);

10 PT PK.***Decrypt***(CT);

▷ **Notice**

11 *ABE_PK*.**Encrypt**(PK) refers to **ABE_Encrypt** shown in Section 5.2.1, where $\mathbb{PT} = PK$, the attribute policy assigned to $PK = \{\mathbf{A}\}$;

13 *ABE_SK*.**Decrypt**($\mathbb{CT}$) refers to **ABE_Decrypt** shown in Section 5.2.1, where the attribute policy assigned to $\mathbb{CT} = \{\mathbf{A}\}$.

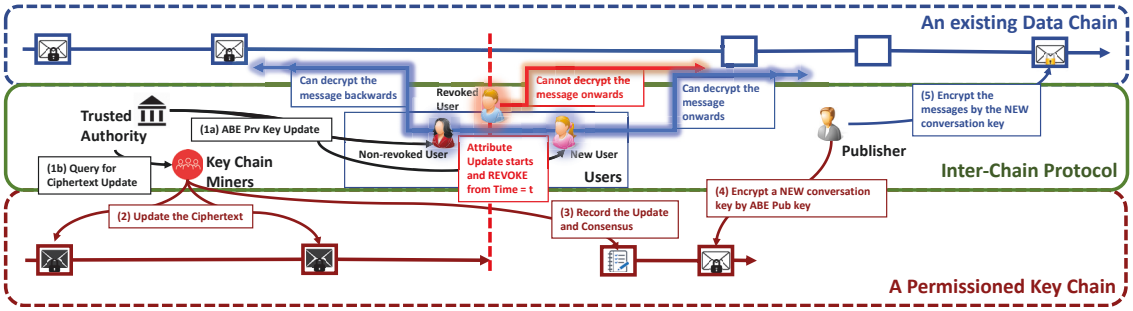


Figure 5.5 : The process of an attribute update is shown. Only the non-revoked DU and new DUs obtain the updated ABE private key, so that they can access to all related data across the *Data Chain*. In contrast, the revoked DU can only access to the history with the outdated ABE private key and updated ciphertext edited by the miners.

5.3.4.3 Protocol of Attribute Updates on a Key Chain

This section describes the **Attribute Updates** in the **Inter-Chain Protocol**. In the case that new members are to be registered in attribute α, or current members are to be revoked from α, the *Data Chain* remains stable while the *Key Chain* updates those blocks which store encrypted conversation keys assigned with access policy $\{\mathbf{A}\}$ satisfied by α, i.e., $\mathbb{ABE}_{\mathbf{A}}(PK), \alpha \vDash \mathbf{A}$. The detail of updating in a *Key Chain* is shown as follows.

Step 1: Update the Ciphertext in the Key Chain. The TA runs **ABE_Update** and sends entities the corresponding updated attribute parameters in any secure and encrypted channel; see Define of Algo. 1, Lines 1-2 of Algo. 3 and Steps (1a) and (1b) of Fig. 5.5. After that, each *Miner* calculates the updated $\mathbb{CT}$ for any of its conversation keys PK whose access policy $\{\mathbf{A}\}$ can be satisfied by α. Here the $\mathbb{CT}$s before and after the update are denoted as $\mathbb{ABE}_{\mathbf{A}}(PK)$ and $\mathbb{ABE}'_{\mathbf{A}}(PK)$, respectively.

Step 2: Update Transactions in the Key Chain. In Lines 5-6 of Algo. 3, with the aid of the CH_{SK}, each *Miner* overwrites $\mathbb{CT}$ via the given indexes i and j with the updated $\mathbb{CT}_{Update}$; also see Step (2) of Fig. 5.5. Then by including the updated

Algorithm 3: The update process of members with specific attributes

▷ **Define**

All functions inherited from Algo. 1

Input: An attribute α

▷ TA **Computation phase**

1 $(\mathrm{UF}_{SK}, \mathrm{UF}_{\mathbb{CT}})$ $TA.$***ComputeUpdate***(α); // TA computes the updating factor (UF) associated with an attribute α.

2 $Nodes_{\alpha} \Leftarrow TA.$***Send***$(\mathrm{UF}_{SK})$ **and** $Miner \Leftarrow TA.$***Send***$(\mathrm{UF}_{\mathbb{CT}})$; // TA sends the updating factor of SK to the non-revoked nodes and the updating factor of $\mathbb{CT}$ to nodes assigned with {Miner}.

▷ **Ciphertext updating phase**

3 $\mathbb{CT} \leftarrow$ ***Find***$(t_{i,j})$; // Miners retrieve $\mathbb{CT}$ with given $t_{i,j}$ in the *Key Chain*.

4 $\mathbb{CT}_{Updated} \leftarrow Miner.$***UpdateCT***$(\mathbb{CT}, \mathrm{UF}_{\mathbb{CT}})$

5 $Miner.$***Overwrite***$(\mathbb{CT}_{Updated}, t_{i,j}, CH_{SK})$; // Miners overwrite the outdated $\mathbb{CT}$ with $\mathbb{CT}_{Updated}$ in $t_{i,j}$ by using CH_{SK} obtained from Line 7 of Algo. 1.

6 $\mathrm{Block}_{Key,Pending}$ $Miner.$***UpdateMPT***$(Hash(\mathbb{CT}_{Updated}.Random))$; // The *Miners* update the latest state of $Hash(\mathbb{CT}_{Updated}.Random)$ in a new pending block of the *Key Chain*.

7 **while** Consensus **do**

8 $\quad Miner.$***Verify***$(\mathrm{Block}_{Key,Pending})$;

9 $\quad$ **if** $Block_{Key,Pending}$ *is invalid* **then**

10 $\quad\quad$ **alarm and exit**

11 *Key Chain*, i, j $\mathrm{Block}_{Key,Pending}$; // Append the new block to the *Key Chain*, obtain the index j of the transaction in Block-i.

▷ **Private key updating phase**

12 $ABE'_{\alpha,SK}$ $Nodes_{\alpha}.$***UpdateABE_SK***$(ABE_{\alpha,SK}, \mathrm{UF}_{SK})$; // $ABE'_{\alpha,SK}$ is unique for each member of α, i.e., identity-based.

13 PT $Nodes_{\alpha}.$***Decryption***$(ABE'_{\alpha,SK}, t_{i,j})$; // The non-revoked nodes can then decrypt $\mathbb{CT}_{Updated}$ in $t_{i,j}$ by $ABE'_{\alpha,SK}$.

14 $Node_{new,\alpha} \Leftarrow \mathrm{TA.}$***Send***$(ABE'_{\alpha,SK})$ **and** $PT \leftarrow Node_{new,\alpha}.$***Decryption***$(ABE'_{\alpha,SK}, t_{i,j})$; // A new member of attribute α can also conduct the decryption.

▷ **Notice**

15 ***ComputeUpdate***, ***UpdateCT*** and ***UpdateABE_SK*** constitute **ABE_Update** shown in Section 5.2.1.

$MPT(h)$ in each of the new pending Block-h, with the latest state of the hash of the updated random number related to $\mathbb{CT}_{Update}$, a consensus process runs to complete appending Block-h to the *Key Chain*; see Lines 7-11 of Algo. 3 and Step (3) of Fig. 5.5. Note that a Practical Byzantine Fault Tolerance (PBFT) (or a PBFT-like) consensus algorithm [40] with a lower-bounded number of faulty nodes $N \geq 3f + 1$ is usually considered in a permissioned Blockchain, where N denotes the number of *Miner*s and f denotes the number of faulty *Miner*s. The proposed verification scheme is applied during this process to ensure the consistency among the *Miner*s to prevent a malicious *Miner* from conducting a false update to expose an outdated version of the message.

When a DP publishes a new message with a new PK assigned to attribute α after the revocation; refer to Steps (4)-(5) of Fig. 5.5, it can be found in Lines 11-14 of Algo. 3 that only the non-revoked DUs with attribute α and the new DUs can conduct the decryption process to any messages on the *Data Chain* assigned to attribute α by their own updated $ABE'_{\alpha,SK}$. Recall that $ABE'_{\alpha,SK}$ is identity-based. On the other hand, the revoked members fail to conduct the following after this update event (update-oriented):

- retrieve the new PK, e.g., $PK_{\mathbf{A},P_2}$ in the absence of its corresponding new $ABE_{\alpha,SK}$
- directly retrieve the past PK before the revocation from an updated $\mathbb{ABE}'_{\mathbf{A}}\{PK_{\mathbf{A},P}\}$ via its own outdated $ABE_{\alpha,SK}$.

This is because the revoked members of attribute α do not have the updated $ABE_{\alpha,SK}$ regarding the $\mathbb{CT}_{Update}$ after an update happens. Meanwhile, the revoked members cannot directly access the outdated $\mathbb{CT}$ from the past blocks of the *Key Chain* as $\mathbb{CT}$ has been overwritten without breaking the linked structure by using the CH algorithm. In contrast, the non-revoked DUs and new DUs of α have the

latest updated $ABE'_{\alpha,SK}$. This indicates that they can decrypt any $\mathbb{CT}_{Update}$s in the past and any future new $\mathbb{CT}$s, where their attribute $\alpha \vDash \{\mathbf{A}\}$.

5.4 Analysis and Evaluation

In this section, we present the system analysis and simulation in terms of energy, scalability, compatibility, and security.

5.4.1 System Analysis

5.4.1.1 Energy Overhead

We consider the IoT devices are typically FPGA (e.g., AVNET's Zedboards) or SBCs (e.g., Raspberry Pi) which can be the DU or DP in the analysis regarding the energy overhead. Such IoT devices are considered to be able to handle the following behaviours, as illustrated in Fig. 1, including

- conducting the cryptographic operations for conversation keys and ABE cryptographic operations (encryption/decryption);
- signing transactions by themselves when publishing new messages; and
- exchanging network packets with the higher layer devices (e.g., the edge network or the core network).

Any other devices that are incapable of delivering and handling the above behaviours are considered to be the data collectors (e.g., sending sensor data to SBCs). It is pointed out that the proposed system is suitable for typical IoT systems with such IoT devices for the following two reasons.

Firstly, it is not the responsibility for IoT devices to maintain the chains, as illustrated in the table on the right-hand side of Fig. 5.1. Rather, it is the sufficiently

powerful machines (cloud servers or local data centres) that are in charge. The CH-oriented overhead of the update process is conducted by the miners of *Key Chain*s, while the miners also tend to be located in cloud or local data centres. As such, IoT devices only deal with the overhead of running the cryptographic operations and transmitting the ciphertext.

Secondly, a cluster of *Key Chain*s is designed to provide differentiated access control of the data chain for different organizations and communities, so that the overlapped attributes in different contexts (different *Key Chain*s) can be properly managed. This enables IoT devices to be usually involved in only a single *Key Chain* and the *Data Chain*, so that the total communication and computation overhead on IoT devices can be regarded as the overhead between the *Data Chain* and a single *Key Chain*.

Upon the considered communication and computation overhead that excludes the Blockchain maintenance for IoT devices running across the *Data Chain* and the related *Key Chain*, we present the following energy analysis.

Computation overhead and performance:

Our proposed system can be suited to IoT systems in terms of the computation overhead and performance. We take Raspberry Pi for an example, because of its popularity as an SBC. To be specific, the computation overhead mainly includes 1) the en/decryption of the (a)symmetric encryption algorithm; and 2) the en/decryption of the ABE algorithm. A DP publishing a message needs to conduct the encryption process, while the decryption process accounts for the computation overhead that a DU needs to incur. We utilize the metrics proposed in [11], Execution Time (denoted as t, and measured in second). We also define **the number of the complete process flows under the given condition** ($\mathcal{T}$) to evaluate the

energy efficiency/consumption, as given by

$$\mathcal{T} = \frac{3.6 \times CV}{\eta},$$

where C denotes the battery capacity measured in `mAh`; V denotes the working voltage of the tested Raspberry Pi measured in `Volt`; and η refers to the proposed metrics in [11], i.e., the Energy Consumption for a single execution measured in `J`.

It can be concluded from Table 5.2 that, even the most energy-consuming type of Raspberry Pi, i.e., 1B, can execute the encryption operation around 18,800 times and the decryption operation 30,000 times with a 10,000`mAh` battery, in the context of an 80-bit security level. The most energy-efficient type, Zero, and the most powerful 4B can both execute the encryption operation around 60,000∼70,000 times and the decryption operation around 90,000∼100,000 times with the same battery life. In contrast, the optimized CPU performance of 4B can potentially improve $t_{ABE,en}$ to around 10`s` and $t_{ABE,de}$ to around 9`s` with a security level of 128 bits under 30 attributes, respectively. In such context with a stronger security level, Raspberry Pi 4B can still execute the encryption operation around 6,000 times and the decryption operation around 6,700 times. As a result, 4B can even be used in scenarios with a fixed power supply, e.g., CCTV cameras or solar-based devices, so that higher computation performance (compared to 1B and Zero) can guarantee a stronger security level, at the cost of a slightly higher energy consumption.

Communication overhead and performance:

Our system is also suited for IoT systems in terms of communication overhead and performance (with Wi-Fi, or even more energy-saving protocols where messages can be chunked into pieces, e.g., LoRa, NBIoT, and Zigbee). As a DP, an IoT device uploads the ciphertext of messages and ciphertext of the conversation key (i.e., CT and $\mathbb{CT}$, respectively) to the *Data Chain* and *Key Chain*. The DP (playing the role

Table 5.2 : It shows the computation overhead and performance comparison between different types of Raspberry Pi with a security level of 80 bits under 30 attributes (the last column shows the performance of 4B with a 128-bit security level under 30 attributes); $C = 10{,}000\texttt{mAh}$, $V = 3.3\texttt{V}$ and $5\texttt{V}$. We focus on the cryptographic operations. The typical AES scheme proposed in [51] with pipelining is used to encrypt (and protect the confidentiality of) conversation keys. We consider the widely accepted CP-ABE scheme [27] which is part of AndrABEn, an open source ABE library particularly optimized for Android/smartphone/IoT operating systems. The ABE library has been adopted in many existing studies, e.g., [11, 12].

Raspberry Pi		1B	Zero	2B	3B+	4B	4B (120-bit)
Encryption (AES*+ABE**)							
t_{en} $(t_{AES,en} + t_{ABE,en})^{\dagger}$		5.30s	5.30s	3.40s	1.87s	1.06s	10.30s
η	3.3V	6.31J	1.75J	2.58J	2.47J	2.01J	19.57J
	5V	9.54J	2.65J	3.91J	3.74J	3.05J	29.67J
$\mathcal{T}$	3.3V	18,836	67,925	45,975	48,128	58,987	6,071
	5V	18,868	67,925	46,036	48,128	58,962	6,068
Decryption (AES*+ABE**)							
t_{de} $(t_{AES,de} + t_{ABE,de})^{\dagger}$		3.30s	3.30s	2.11s	1.19s	0.66s	9.30s
η	3.3V	3.93J	1.09J	1.60J	1.57J	1.25J	17.67J
	5V	5.94J	1.65J	2.43J	2.38J	1.90J	26.78J
$\mathcal{T}$	3.3V	30,252	109,091	74,083	75,630	94,737	6,723
	5V	30,303	109,091	74,181	75,630	94,697	6,720

* The data is sourced from [167] based on the pipelining scheme [176].

** The data is sourced from [211, 7].

† t_{en} and t_{de} denote the execution time for encyption and decryption, respectively. $t_{AES,en}$ and $t_{AES,de}$ are considered to be equal. $t_{AES,keygen}$ can be negligible because of the pipelining [176]. $t_{AES,keygen}$ can also be negligible as it is the TA's responsibility.

of DU) also downloads the CT and $\mathbb{CT}$ from the *Data Chain* and related *Key Chain*, respectively.

Transmitting CT: As block ciphers, the size of CT is nearly equal to that of PT. A 1MB message is considered in the computation analysis, which has been proved acceptable based on the result [44, Tables III - VIII].

Transmitting $\mathbb{CT}$: According to [187], the communication overhead of transmitting $\mathbb{CT}$ is 50kB, 96kB, and 140kB, for 80-bit, 112-bits, and 128-bit security level, respectively. The overhead consists of, 1) transmitting $\mathbb{CT}$; and 2) receiving the updating factor of SK (or $ABE'_{\alpha,SK}$ in some ABE algorithms) from the TA whenever an attribute is updated. The latter happens infrequently in practice so that we can focus on the former, which are 25kB, 48kB, and 70kB, as the two aspects of the overhead share the overhead on a fifty-fifty basis [47, Fig. 5]. Thus, Wi-Fi satisfies such overhead based on the normal LAN throughput of Raspberry Pi [211]. It is noteworthy that the energy consumption of Wi-Fi is about 10% more than the case where all communication modules are turned off [125].

5.4.1.2 Scalability and Compatibility

Because of the editability of our proposed system, this analysis focuses on the scalability of a *Key Chain* with the redactable data structure, as well as the compatibility of our system integrated with a cloud-based system. We develop a Python-based testbed which is based on a popular (4.7k stars) Python-based open source platform[5] and the Charm Crypto native libraries[6]. The testbed allows us to run multiple entities on a single machine. We can simulate multiple nodes with different roles (i.e., the TA and miners typically located in clouds/data centers, and the DPs/DUs typically running on IoT devices) on a single 2017 iMac with 10.13.3 ma-

[5]https://github.com/dvf/blockchain

[6]https://github.com/JHUISI/charm

cOS High Sierra, a processor of 2.3 GHz Intel Core i5 and 16 GB 2133 MHz DDR4 memory.

Scalability: We conduct a comparison among our three proposed data structures of $MPT(h)$ (see description below (5.3)), and the state-of-the-art design developed in [154] regarding the overhead and searching complexity of H, K and R, where the notations are described as follows.

- H, (blocks): the Block Height, starting from a block where a given $\mathbb{CT}$ being inserted or updated to the latest block.
- K, (inserts/blocks): the Insert Rate, the average rate for DUs to insert $\mathbb{CT}$s in a *Key Chain*.
- R, (updates/blocks): the Update Rate, the average rate of attribute updates being applied in a *Key Chain*.

Recall that the data structures discussed in this paper correspond to the world state and smart contract used in Ethereum [191]. In the segment figures, the blue curves denote the structure of engraving transaction; the red curves denote that of storing $Hash(v(i,j))$ in a smart contract as a pure variable in the existing world state MPT; the green curves denote that of storing $Hash(v(i,j))$ as a new dedicated MPT in addition to the MPT world state; the gray curves denote that of storing $\mathbb{CT}$ in a smart contract as a pure variable in the existing world state MPT, vaguely discussed in [154][7]. For the red and gray curves, there exist the unrelated addresses of normal accounts and smart contracts in the same *Key Chain*, the number of which is denoted as C. C has an apparent impact on the searching complexity.

[7]We can assume that $\mathbb{CT}$ is directly stored as a variable in smart contracts rather than storing $Hash(v(i,j))$, as the overwriting or purge of the on-chain-stored outdated $\mathbb{CT}$ by the latest updated version is not discussed in [154]. This implies that [154] has no new verification scheme for the update process.

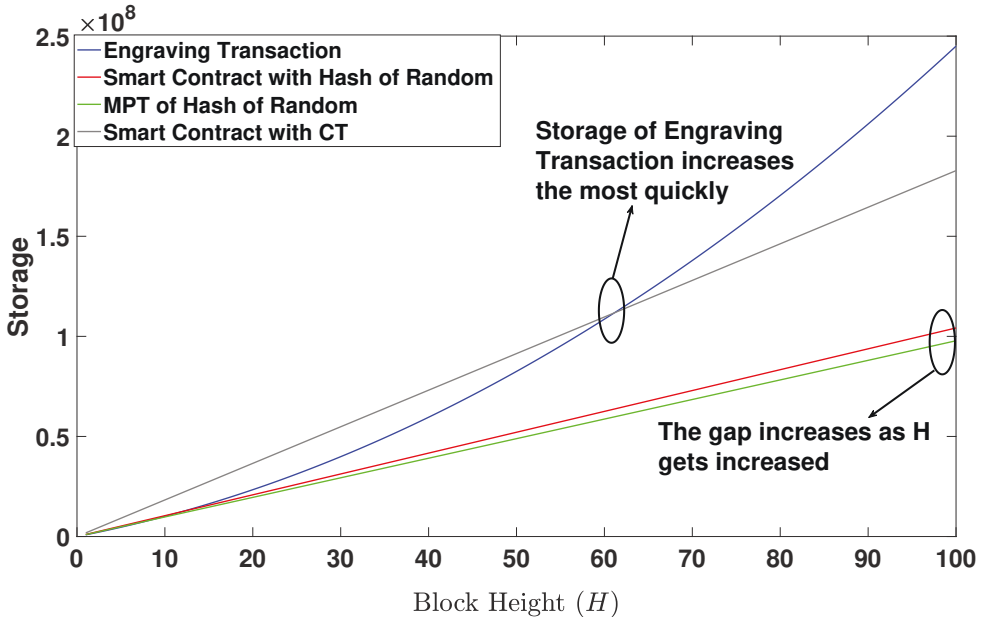


Figure 5.6 : The comparison among our three proposed structures and the structure proposed in [154] regarding the storage overhead with respect to H, where $R = K = 1000$. Note that the y-axis denotes the storage complexity.

Overhead: It is shown in Fig. 5.6 that in the case of $K \simeq R$ with the growth of H, the green, red, and gray curves have a linear growth rate. The green grows more slowly than the red and gray, with an increasing gap between the green and red curves. The blue curve grows the most quickly with an intersection point to the gray curve at around $H = 60$, which is at quite an early stage. In the case where a large value of H is expected to appear in a *Key Chain*, the blue curve has a superior growth rate, while the growth rate of the green curve is the lowest.

In Figs. 5.7 and 5.8, we conduct a controlled-experiment that the values of K and R are set to be the independent variables respectively with a certain $H = 10000$, in order to investigate the correlations between K and the growth rate, and between R and the growth rate. We also choose two different values of R in Fig. 5.7 and K in Fig. 5.8 to evaluate the side effect of one to another.

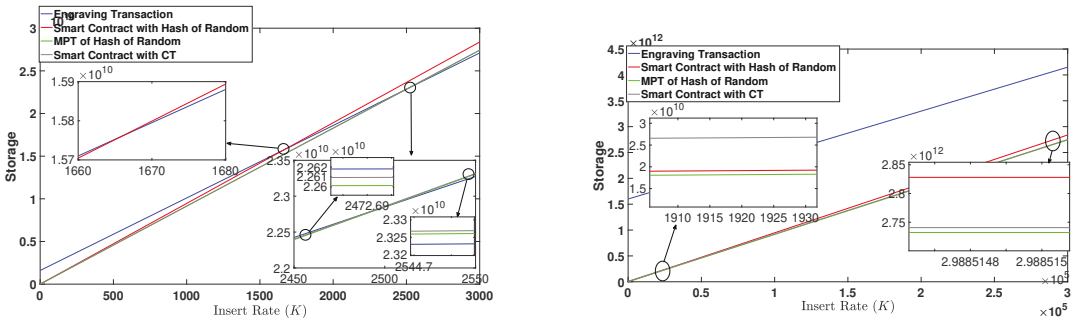
It is shown in Fig. 5.7(a) that the blue curve, which starts with the largest storage

space, has the smallest growth rate as K increases from 0 to 3000 inserts/block with $R = 1$ updates/block. This can be found at the intersections where $K \simeq 2500$, as shown in the sub-figure in the bottom right corner of Fig. 5.7(a). The red curve has the largest growth rate as K increases. For a large value of $R = 1000$ within the range of K from 0 to 300000 in Fig. 5.7(b), the blue curve has a superior growth rate. The gray curve intersects the red curve right after $K \simeq 2000$ and increasingly approaches the green curve as K increases. The growth rate of the green curve remains the lowest throughout the range. It can be concluded that K has a strong impact on the red curve, and a negligible impact on the blue curve among all these structures. Nevertheless, the blue curve is strongly impacted by a large R, while the green curve performs the best among all these structures throughout the range of K regardless of the value of R.

It is shown in Fig. 5.8(a) that the blue curve has a superior growth rate among these structures, the gray curve performs the second to the worst, and the green curve has the lowest growth rate. As shown in Fig. 5.7, K has a much stronger effect on the gray, red and, green curves than the blue curve. As such, for a small value of $K = 1$ where $R \in [0, 200]$, it can be concluded that the blue curve is incapable of the storage for a *Key Chain*. If a large K is set, e.g., $K = 100000$, shown in Fig. 5.8(b), the blue curve performs the best when $R \leq 40$. As R increases, the blue curve grows sharply, while the green curve increases slowly with the lowest growth rate. It can be concluded that R has the strongest effect on the blue curve, and the weakest effect on the green curve.

The green curve is the most appropriate data structure implemented in the scheme of an attribute update, if the settings of the *Key Chain* that might happen in a realistic environment are taken into account, as shown in the following,

- K is much likely greater than R;



(a) $K \in [0, 3000]$, $R = 1$ (b) $K \in [0, 300000]$, $R = 1000$

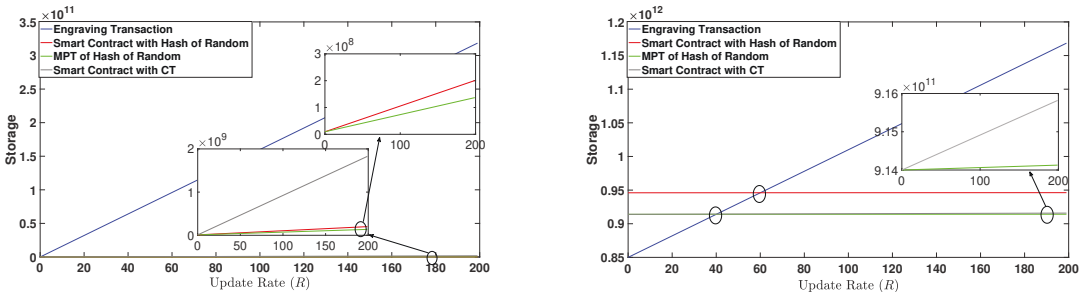
Figure 5.7 : The comparison among our three proposed structures and that of [154] regarding the overhead with respect to a constant $H = 10000$ and an upper bound of K in two different values of R. Note that the y-axis denotes the storage complexity.

- both values of K and R are likely within the range of $(0, 10000)$ due to the throughput restriction[8] of the commonly used consensus algorithm in a *Key Chain*, i.e., PBFT.

Therefore, it is unrealistic to set a sufficiently large value of K to support better performance of the blue curve among all these structures. On the other hand, the green curve has the most stable and smallest growth rate with respect to both R and K. It is expected to be the most appropriate structure to update the attribute in a *Key Chain*.

Searching Complexity: As Engraving Transaction has a searching complexity of at least $O(HR)$, which is significantly greater than the other three which are only in a logarithmic order, we only compare Smart Contract with Hash of Random (the red curve in Figs. 5.6-5.8), Smart Contract with CT (the gray curve in Figs. 5.6-5.8) and MPT of Hash of Random (the green curve in Figs. 5.6-5.8). Also because an inner MPT used in a smart contract is implemented in both Smart Contract with

[8]The throughput (transactions per second) is limited in the case where the Byzantine tolerance needs to be satisfied [133].



(a) $R \in [0, 200]$, $K = 1$

(b) $R \in [0, 200]$, $K = 100000$

Figure 5.8 : The comparison among our three proposed structures and that of [154] regarding the overhead with respect to a constant $H = 10000$ and an upper bound of R in two different values of K. Note that the y-axis denotes the storage complexity.

Hash of Random and Smart Contract with CT, which doubles the depth of the MPT, Smart Contract with Hash of Random and Smart Contract with CT have the same searching complexity, i.e., $O(log(HK + C))$, as shown by the blue curve in Fig. 5.9. On the other hand, MPT of Hash of Random has a searching complexity of $O(log(HK))$, as shown by the red curve in Fig. 5.9. It can be concluded that the red curve has a smaller growth rate than the blue curve, according to Fig. 5.9. As the value of C increases and asymptotically stabilizes, the performance of the blue curve converges to that of the red curve.

Therefore, it is the most appropriate to introduce a new dedicated MPT for storing $Hash(v(i, j))$ in a practical implementation of the *Key Chain* based on both the overhead and searching complexity (MPT of Hash of Random, the green curve in Figs. 5.6-5.8 and the red curve in Fig. 5.9). In contrast, if the unchanged structure of block headers is preferable, e.g., Ethereum, for backward compatibility with the existing Blockchain, $Hash(v(i, j))$ can be stored in the existing the world state MPT as a pure variable (Smart Contract with Hash of Random, the red curve in Figs. 5.6-5.8 and the blue curve in Fig. 5.9). Also, Engraving Transaction (the blue curve in Figs. 5.6-5.8) can be used in a testing environment where the complexity is not the

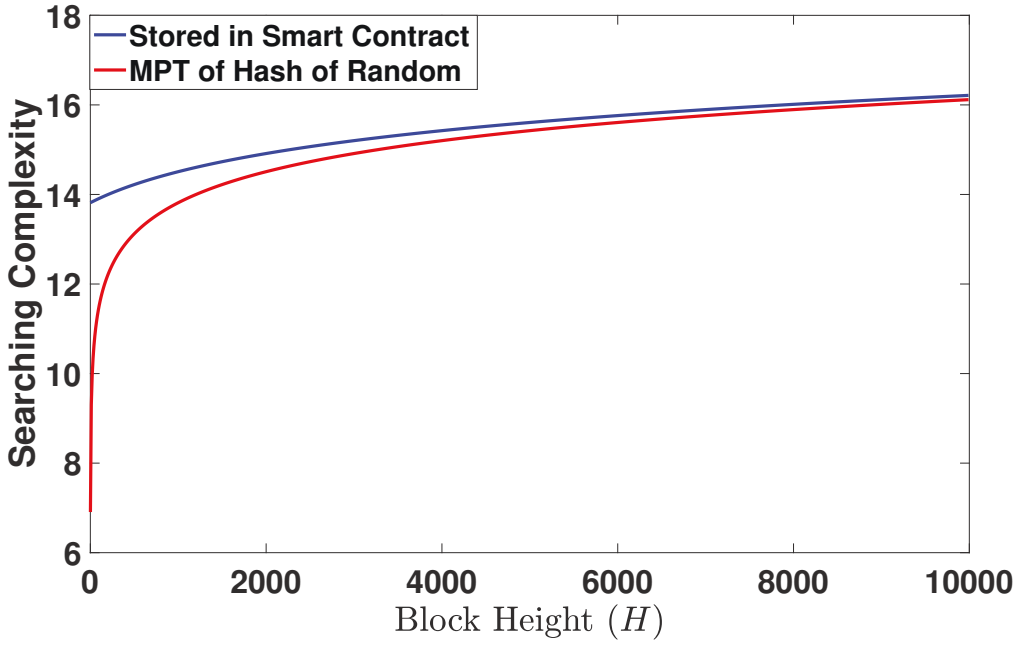


Figure 5.9 : The comparison among our proposed structures and that of [154] regarding the searching complexity with respect to H, where $R = K = 1000$, $C = 1000000$. Note that the y-axis denotes the searching complexity.

focus because of its simple implementation.

Compatibility: It is worth noting that our proposed system is a general decentralized framework and is generic to all the consensus protocols, cross-chain protocols, and cryptographic algorithms. Although we do not encourage completely relying on cloud services due to risks on the reliability, traceability, and transparency, the proposed structure and protocols complement to cloud services, rather than a competitor. In our scenario, Blockchain is considered to be a decentralized data structure, while part of the nodes storing a copy of data running on the cloud servers can be a better option. As such, either faulty cloud servers or attacks on cloud servers cannot have strong impacts on our proposed Blockchain system.

5.4.2 Security Analysis

This analysis focuses on the security of our entire multi-layer Blockchain system. The TA is supposed to have the highest security level and the immunity to cyber and physical attacks, as is typically assumed in ABE systems (this assumption still holds for multi-authorities as the trust is only alleviated for a single authority while the wholeness must be still trusted [42]).

Attack model: External and internal adversaries target to break the confidentiality and integrity of the data recorded in Blockchains. Here, the external adversaries are nodes which can only obtain the public information, such as the ciphertext of a message CT in the *Data Chain* and/or the public key of an attribute α, i.e., $ABE_{\alpha,PK}$. The internal adversaries can be categorized into two types,

- the nodes which used to be legitimate and now are revoked in regards to a specific attribute, and have not had their ABE_{SK} updated;
- the nodes which have been assigned with a specific attribute *Miner* (*Key Chain*).

In an update process, the internal adversaries can conduct a false update to compromise the integrity (or in other words, tamper-resistance) of the redactable *Key Chains* with legal identities and valid keys, i.e., ABE_{SK}. The proposed architecture is expected to be secure against the two types of adversaries in terms of the confidentiality of the published messages in the *Data Chain* and the integrity of the updated ciphertext $\mathbb{CT}_{Update}$ in the *Key Chain*s.

Confidentiality: We consider that an external or internal adversary targets to obtain the encrypted message CT stored in the *Data Chain* which must not be accessible to the adversary, i.e., the adversary does not own any attribute α satisfying the access policy $\{\mathbf{A}\}$ of CT. In the proposed system, the original message P is

encrypted to CT and stored in the *Data Chain*, while the decryption key $PK_{\mathbf{A},P}$ is encrypted by an ABE algorithm to be $\mathbb{CT} = \mathbb{ABE}_{\mathbf{A}}(PK_{\mathbf{A},P})$, as described in Section 4.2.

For an external adversary whose attribute α has never satisfied the access policy $\{\mathbf{A}\}$ before, the adversary cannot decrypt $\mathbb{CT} = \mathbb{ABE}_{\mathbf{A}}(PK_{\mathbf{A},P})$ to obtain $PK_{\mathbf{A},P}$, as can be guaranteed by the security of the ABE algorithm. Therefore, the external adversary does not have the decryption key $PK_{\mathbf{A},P}$ and cannot extract the message P, as guaranteed by the security of the (a)symmetric encryption algorithm.

For an internal adversary whose attribute β has once satisfied the access policy $\{\mathbf{A}\}$, the attribute β and the encrypted conversation key $\mathbb{CT}$ are updated. As a result, the revoked adversary without the correspondingly updated version of $ABE_{\beta,SK}$ cannot decrypt $\mathbb{CT}$ to obtain the decryption key $PK_{\mathbf{A},P}$; see the revocation design in Section 4.4.3. As a result, the internal adversary cannot directly access the message P, as it is unaware of the decrypted key $PK_{\mathbf{A},P}$. [9]

Integrity/Anti-Tampering: The integrity can be guaranteed as long as the lower bounds of faulty tolerance of consensus protocols (e.g., 33% for PBFT [40] or 50% for PoW [135]) are all satisfied among the *Data Chain* and *Key Chains*. The history of updates is immutable and verified during the consensus. Because of the synchronization among the miners [40] in a permissioned Blockchain where a PBFT (or PBFT-like) consensus is implemented, an internal adversary with an attribute *Miner* conducting a false update (it is possible due to identical CH values) and providing an outdated version of blocks to any DUs would fail to synchronize with other honest *Miner*s and would be detected by the system, as discussed in Section 4.3.2. Therefore, by means of punishment for the internal adversaries found

[9] A direct access indicates an on-chain access, i.e., data being looked up directly from the *Key Chain*, but not replicating a copy locally.

conducting a false-update attack, the integrity and tamper-resistance of Blockchains can still be ensured in our proposed system. In addition, the updated ciphertext, $\mathbb{CT}_{Update}$, is attached with the signature of the leader of the current consensus round. As a result, there is no need for the DPs to remain online, and the man-in-the-middle-attack can be prevented.

5.5 Conclusions

Remark that the content needs to be editable to fulfill the changes of attributes in an ABE-based Blockchain that provides fine-grained access control, how the contradiction can be solved between this requirement and the immutability of Blockchains has been resolved in this chapter. We proposed a new Blockchain system with fine-grained access control for IoT applications, where the membership and attributes of individuals, i.e., data owner, user, and miner, can be updated securely in anti-tamper Blockchains by integrating CH algorithms in a new multi-layer chain architecture. The system addresses direct data leakage caused by revoked members, and maintains good compatibility with major consensus protocols, cross-chain protocols, and encryption algorithms. The system can be further optimized by having multiple TAs in a decentralized manner to guarantee the superior security level of the TA. Our system analysis and simulation show that the proposed system can outperform existing solutions in terms of overhead and complexity.

By utilizing the proposed system, a secure, manageable and decentralized data access control which enables large-scale and high-precision data management can be readily deployed in an IoT network. The access control can, for the first time, stop revoked users or miners from accessing not only the future data but the past data in a Blockchain, thereby substantially improving the manageability without compromising the tamper-resistance of the Blockchain. This allows flexible and secure Blockchain-based IoT services and management to be adopted by regional/global

IoT leagues. Our analysis provides managerial guidelines for risk assessment, cost evaluation, and energy and storage requirements, so that appropriate technical solutions can be designed to meet specific business requirements.

Chapter 6

A Novel Dual-Blockchained Structure for Contract-Theoretic LoRa Networks

Many existing studies present simple and stilted integration between Blockchain and IoT technologies without dedicated optimization. However, such integration cannot take full advantage of both technologies. On the contrary, Blockchains might incur negative impacts from the inherent properties of IoT networks, while the inherent properties need to be secured by the global validation of Blockchains. How dedicated optimizations could be designed upon a Blockchain-based IoT application such that the combination could be natural, smooth, and effective becomes gradually substantial. Under this background, this chapter, taking Blockchain-based LoRa networks as a case study, proposes a novel Dual-Chain-based LoRa system that provides global cross-validated security. The optimized architecture can scale both the Dual-Chain system and the LoRa network.

6.1 Introduction

LoRa (Long Range), one of the most popular low-power wide-area network (LPWAN) technologies, has been reported to outperform cellular-based LPWAN (e.g., Narrow-Band IoT) in corporate or private realms, due to its self-driven public participation and comprehensive open-source community [169, 149]. Operating in an unlicensed band, several open-source protocols are presented (e.g., LoRaWAN [8]), where a star-of-stars network is established between end-devices and a LoRa Gateway, and between LoRa Gateways and a LoRa Controller (i.e., LoRa network server). It is reported that LoRa has held the highest market share in some countries and

accounted for the highest annual unit shipments along with an increasing projection of the entire LPWAN market [28, 128, 13].

The use of unlicensed bands and open-source platforms compel LoRa to stay on top of improving the scalability and flexibility among self-deployed gateways (e.g., relieving traffic congestion resulting from the channel limit) [58, 142], while incentivizing more private owners to increase coverage. To achieve a secure, effective, and fair incentive mechanisms, the contract-theoretic incentive mechanism [83, 31] is proposed to incentivize the private deployment of LoRa [198], densify technology adoption and improving coverage and spectrum utilization.

However, existing contract-theoretic designs neither consider a malicious LoRa Controller which censors the LoRa Gateways by paying less reward, nor apply a reliable mechanism to prove the validity of the incentive processing. Blockchain is considered to be suitable to address the issues of centralization by taking advantage of the decentralized architecture and tamper-resistant validation [140, 52]. However, traditional Blockchain technologies fail to handle massive data streams, incurring significant latency and low throughput [163, 188]. In addition, the one-device-to-many-gateway property (which is helpful for redundancy) of the most popular protocol, LoRaWAN, may compromise the scalability of Blockchain and the scalability of LoRa networks.

The above issues remain and a holistic solution is in demand. Specifically, the remaining challenges are: 1) how the contract-theoretic incentive mechanism can be integrated into the system, in which a new self-driven flow control protocol can relieve the traffic congestion and throughput loss resulting from duplicated data uploaded to the Blockchain; and 2) how the Blockchain can be integrated into the proposed contract-theoretic LoRa network to provide secure and scalable data storage services.

In this chapter, we develop a new Dual-Chain-based LoRa system by taking advantage of the Directed Acyclic Graph (DAG) structure. Such DAG is set along with a typical chain-based structure that provides identity registration and protocol monitoring in smart contracts. By interacting with both the DAG and identity chain and leveraging decentralized global cross-validation, a new contract-theoretic incentive mechanism can be secured. The contract-theoretic incentive mechanism consists of a new self-driven flow control protocol which scales the LoRa network by relieving traffic congestion, and improves the throughput of the Blockchain, scaling the Dual-Chain system. As a result, the system enables: 1) strong compatibility with typical LoRaWAN protocols; 2) the Proof-of-Task-Overhead (PoTO), a new spam protection dedicated for LoRa networks to reduce resource waste; 3) efficient and flexible data storage services at any time with high throughput; and 4) the transparency and fairness of incentives and data storage services because of the tamper-resistance property of the decentralized cross-validation protocol.

As revealed by our analysis and simulation results, the proposed Dual-Chain-based LoRa system can significantly improve the throughput of the Blockchain, while maintaining a high area utilization.

The rest of this chapter is organized as follows. In Section 6.2, we present the proposed system model about the new Dual-Chain-based structure and the new flow control protocol. The numerical simulation is shown in Section 6.3. Section 6.4 concludes the chapter.

6.2 System Model

As shown in Fig. 6.1, a number of LoRa Controllers are distributed, each of which runs a regional LoRa network supported by several LoRa Gateways. Many end-devices are distributed in each of the regions. Each region is covered by a LoRa Gateway which forwards data between the end-devices and the corresponding

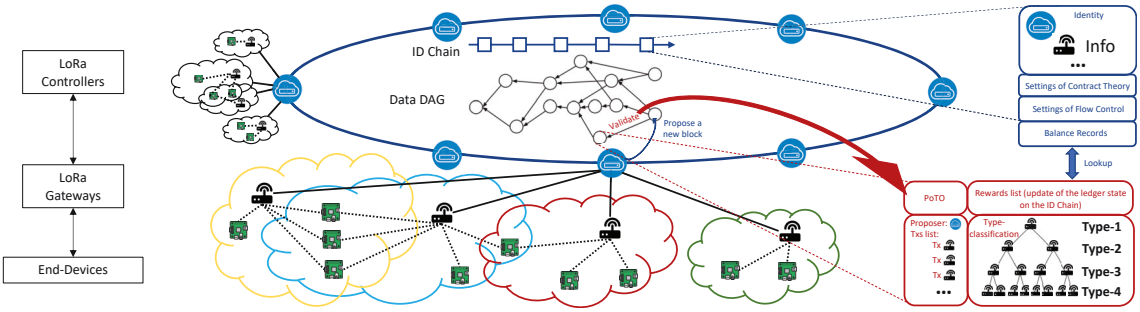


Figure 6.1 : The system architecture of the proposed Blockchain-based LoRa system with contract-theoretic incentive mechanism. A star-of-stars network is established between end-devices and a LoRa Gateway, and LoRa Gateways and a LoRa Controller. A group of controllers constitute a committee and maintain a Dual-Chain structure, i.e., the ID Chain providing registration and monitoring service, and Data DAG providing data storage service. Each controller as a task publisher collects data from its corresponding gateways and pays incentives. Such behaviors are included in a DAG block to be injected to the Data DAG, and cross-validated by other controllers.

LoRa Controller. There could be overlapping coverage areas among the gateways. Data sent from the end-devices situated in the overlapped area (the yellow, blue, and red regions in Fig. 6.1) is likely to be received by multiple gateways that have overlapping coverage.

To prevent these gateways from being congested or transmitting excessive duplicated data to their corresponding controller, a self-driven flow control protocol is applied to the gateways that have a large overlapping coverage area, i.e., the yellow and blue regions. Here, the term "self-driven" indicates that the flow control is not aimed to control traffic by cutting off the transmission. Instead, it is to incentivize the gateways to maximize their profit by complying with any preconcerted rules. A single epoch comprises several time-frames assigned to each gateway. Each authorized gateway is assigned to transmit within its own short time-frame based on the flow control protocol running on the controller.

A controller publishes a contract-theoretic task with a random timeout (i.e., the

controller decides the timeout point which is identical to a single epoch of the flow control) to all of its corresponding gateways. Based on the uploading data size from the gateways to their controller, the gateways are classified into different types in a tree structure managed by the controller, as shown in the type-classification in the red box at the bottom-right corner of Fig. 6.1. The discounted uploading data size as the penalty is applied to the unauthorized gateways transmitting beyond its time-frame. The larger data size a gateway contributes within the epoch, the higher likelihood the gateway is classified into a type that is closer to the root of the tree, and the more incentive can be granted for this task.

Each controller is powerful and robust, and responsible for multiple gateways in a geographical region, and the controllers cooperate and interact with each other[1]. Specifically, all controllers maintain a Dual-Chain-based structure, i.e., an *ID Chain* and a *Data DAG*, based on which the data can be secured by conducting a tamper-resistant cross-validation among the controllers. Critical information is uploaded to the *ID Chain* as smart contracts by controllers during the registration phase, and updated periodically for the cross-validation, as shown in the blue box at the top-right corner of Fig. 6.1. Such information includes the settings and status of published contract-theoretic tasks, the flow control processing, and each gateway's

[1]The LoRa Controllers are typically owned and maintained by the service providers [63]. We consider a decentralized league of hundred-scale controllers in different controller platforms or different geographical regions, while each controller can be much powerful than a single end-device or gateway machine, such as enterprise data centres (e.g., The Things Network, TTN) and smaller-sized organizations using open-source platform [72, 34]/Software-as-a-service (SAAS)/Platform-as-a-service (PAAS) implemented at the cloud-based distributed cluster. The LoRa data originating from densely distributed gateways in the same region ends up at a single controller platform that manages this region. The physically damaged individual controller would be destructive to the regional LoRa service, but would not affect the data retrieval service thanks to the faulty tolerance of the dual-chain structure.

serial number and public key (from which the Blockchain address can be derived).

The *ID Chain* is a regular Blockchain only maintained by the controllers, while a DAG-based structure is used for data storage, i.e., the *Data DAG* managed by both the controllers and gateways. In a DAG-based structure inspired by [152], blocks can be proposed in parallel. Each proposed block needs to validate other pending blocks as a part of contribution to be proved. Such pending blocks are usually those previously proposed by other controllers and yet not validated by sufficient subsequent blocks. To introduce a DAG-based structure instead of a classical chain-based one for data storage is due to its high scalability by supporting parallel blocks/transactions. With its IOTA-like DAG-based solution, the *Data DAG* is particularly suitable for storing large amounts of data in parallel. A LoRa Controller does not need to wait until its previous block has been accepted or others have finalized their blocks, it can publish tasks anytime with any timeout based on its requirements. However, a single *Data DAG* is not as powerful as a chain-based solution in terms of the support for smart contracts [210, 121]. Thus, we propose a separate *ID Chain* which is superior in terms of handling this issue. The *ID Chain*, in spite of its relatively weak support for parallel blocks compared to a *Data DAG*, can be as scalable as a *Data DAG* in terms of transactions per second by implementing the advanced technologies, such as the HotStuff consensus algorithm [201] that enables a high transaction rate among large communities, and the sharding technology (as suggested in Chapter 4) that enables horizontal scalability.

By combining these two primitives, we are able to deliver a controllable and flexible dual-chain system. In particular, different functionalities are split between a *Data DAG* and *ID Chain*. The *Data DAG* enables the parallel data storage service, where LoRa Controllers can initiate contract-theoretic tasks and collect the LoRa data with no need of compliance with the sequence of the consensus process. The *Data DAG* also provides a flexible validation process for businesses

that requires lower security levels and latency. The *ID Chain* is adopted to conduct devices registration/monitoring and contract in initialization/status records. It is also responsible for the incentive payment and balance records of the whole system. Therefore, splitting the data storage and balance records between the *Data DAG* and *ID Chain*, respectively, enables the independence among controllers. Each controller is only responsible for the validity of on-chain LoRa data stored on the *Data DAG*, and not for the validity of the metadata, i.e., the physical meaning of the data stored on the *Data DAG*. This allows for improved controllability and flexibility.

By looking up the information of each gateway in the *ID Chain* and validating each transaction size, hash value, and timestamp in the tree, the other controllers can confirm a contribution of a particular gateway is valid. With the same method, the type-classification conducted by a controller and the transfer of the incentive can also be validated on the *ID Chain* by the other controllers, as shown in the red box at the bottom-right corner of Fig. 6.1. Thus, the cross-validation can be conducted among the LoRa Controllers to prevent malicious behaviors in the *Data DAG*, such as an invalid process of the PoTO protocol, invalid published contract-theoretic task, invalid rewards transfer, and invalid source data from some unknown controllers or gateways.

The proposed system focuses on the uplink transmission, because 1) the uplink transmission is strongly favored in LoRa networks [4]; 2) the downlink transmission can only happen after a successful uplink transmission [105]; and 3) the duty cycle of uplink transmission is regulated by several governments agencies and departments due to the default ALOHA access and limited channel resource, thus limiting the usage of LoRa networks [162].

6.2.1 Dual-Chain Structure and Cross-Validation

The proposed Dual-Chain structure consists of two types of Blockchains, *ID Chain* for gateways registration, contract initialization, and status updating, and *Data DAG* for efficient data storage service, as shown in Fig. 6.1. The Dual-Chain structure is the foundation of the cross-validation conducted among all participating LoRa Controllers, for instance, validating the behaviors of controllers.

Registration and Initialization on ID Chain - The *ID Chain* with a typical chain-based structure is maintained by all participating controllers to conduct the registration of the gateways, contract initialization services, and subsequent status updating. Specifically, the *ID Chain* consists of two operations (see Lines 1 and 2 of Algo. 4):

- *Devices Registration:* Each gateway needs to register its identity to its corresponding controller, including its unique serial number and public key (from which the Blockchain address can be derived). Thus, the controller collects a list of gateways it is responsible for, and can subsequently upload the list to a smart contract on the *ID Chain*. Note that we consider a controller that owns the sufficient performance to maintain multiple Blockchains and manipulate a large amount of data among the gateways and the other controllers.

- *Contract Initialization:* Besides the gateways registration, the smart contract also records the rule of flow control and type-classification for each registered controller based on their own requirements; see Section 6.2.3. The rule includes but not limited to 1) the number of types; 2) the number of gateways in each type; and 3) the amount of incentive paid to each type.

By interacting with such an *ID Chain*, each controller fetches and updates the information of the gateways and types secured by the tamper-resistance property,

and thus a reliable reference can be provided to the controllers during the cross-validation phase.

Data Uploading to Data DAG - A typical chain-based structure incurs poor scalability and flexibility for LoRa networks due to its sequential generation of blocks one after another. Different from the typical chain-based *ID Chain*, a *Data DAG* features a DAG-based structure enabling parallel blocks/transactions to replace the traditional single block. In this chapter, the *Data DAG* features a general DAG-based structure where a block can be injected into the network at any time with an upper-bounded frequency and PoTO certified. Such a block is designed to verify a certain number of pending blocks (e.g., two blocks in IOTA [152]), and will be accepted by the network with a high likelihood if the block has been verified by a sufficient number of forthcoming blocks. Note that the *Data DAG* is compatible with the common tools used in typical DAG-based structures (e.g., the trunk/branch transaction process for bundling in IOTA). It is generic and not limited to any specific tools, protocols, or algorithms. A weighting factor is applied to encourage blocks to verify the most recent pending blocks (i.e., the tits of the DAG), and verify pending blocks proposed by the other controllers (i.e., the cross-validation). Consequently, the system can achieve:

- high throughput of the *Data DAG* (transactions/second) as processing parallel blocks/transactions are now allowed simultaneously;
- instead of a single block generator per slot, a controller does not have to wait until its own previous blocks has been accepted or the other controllers have finalized their own blocks.

Based on the above, high throughput can be achieved in the network in which the controllers can publish a contract-theoretic incentive task for their gateways, and

upload the result at any time with an upper-bounded frequency[2], as well as a timeout parameter on demand. This significantly contributes to an effective mechanism for LoRa networks in which high scalability and flexibility are important.

Data uploading is conducted for the controllers to upload the transactions sent from the corresponding gateways to the *Data DAG*. It consists of the following steps; see Lines 3-11 of Algo. 4).

Data collection at LoRa Controllers: At anytime a controller decides, it publishes a task inquiring LoRa Data forwarded from its corresponding gateways in the form of transactions. The task is subject to a pre-defined timeout and the allocation of the flow control if enabled. The controller marks full contribution only for transactions received from assigned gateways during a time-frame based on the allocation of the flow control (i.e., a discount is applied as a penalty for those breaching the rule; see Section 6.2.2). Note that collecting data from end-devices on the gateways' sides is independent of the flow control. A gateway can suspend the data forwarding while continuing to buffer the received LoRa data, if the gateway is unauthorized. The transaction sent from a gateway is secured by using its private key associated with the public key stored on the *ID Chain*, and also specifically contains the data size and data hash. Any received transactions which 1) belong to unregistered gateways, 2) present unmatched data size or data hash, or 3) display incorrect data format (e.g., not comply with the pre-defined upper-bounded size of uploading data; see Section 6.2.3), are considered to be invalid and discarded by the controller.

[2]Strictly, there should be a lock-phase between each task, i.e., an upper-bounded frequency. The upper-bounded frequency ensures that the previous blocks have been accepted by the *Data DAG* with high likelihood. This prevents the out-of-order of transactions [152], and unexpected rollbacks or forks by leveraging a block-ordering in the case where the data uploading is conducted in a very high rate. In this chapter, the flexibility of publishing tasks without having to limited by a certain period is our focus.

Algorithm 4: Decentralized data uploading mechanism

▷ **Define**

Controller$_i$ and Gateway$_{i,j}$; // The i-th LoRa Controller and its corresponding j-th LoRa Gateway.

Receiver ← Sender.***Send***(Message); // A sender sends msgs to a receiver.

Blockchain ⇐ Sender.***Upload***(Message); // A sender uploads msgs to the chain.

Gateway$_{i,j}$ ← Controller$_i$.***Incentive***(data$_{i,j,k}$); // The i-th LoRa Controller pays incentive to j-th LoRa Gateway on the ID Chain upon the data uploaded in task$_k$.

incentive to j-th LoRa Gateway on the ID Chain upon the data uploaded in task$_k$.

FlowControl // Enabling the flow control

▷ **Registration**

1 Controller$_i$ Gateway$_{i,j}$.***Send***(info);// where info contains the information of Gateway$_{i,j}$ (e.g., Blockchain address).

2 Blockchain$_{ID}$ ⇐ Controller$_i$.***Upload***(info);

▷ **Uploading data**

3 Controller$_i$ publishes $Task_{i,k}$ with $Timeout_{i,k}$, records $TimeStamp_{start}$.

4 **if** *FlowControl is enabled* **then**

5 Invoke **Algo. 3.Defining Congestion**

6 **while** $Timeout_{i,k}$ *not yet reached* **do**

7 Controller$_i$ Gateway$_{i,j}$.***Send***(LoRa_data); // data_size and data_hash are included in LoRa_data apart from the data itself. LoRa_data is secured by the asymmetric encryption.

8 **if** *the sending data from Gateway$_{i,j}$ is invalid* **then**

9 **alarm and drop**

10 **else**

11 Controller$_i$ records $TimeStamp_{j,t}$, i.e., the actual time that Gateway$_{i,j}$ sends LoRa_data.

// The controller ranks and pays Gateway$_{i,j}$ with different types in $Tree_{i,k}$ based on the inbound data size |data$_{i,j,k}$| for $Task_{i,k}$.

12 **if** *FlowControl is enabled* **then**

13 Invoke **Algo. 3.Incentive Allocation**

14 **else**

15 Gateway$_{i,j,k}$ ←Controller$_i$.***Incentive***(|data$_{i,j,k}$|)

// Invoke **Algo. 2** where $i' \neq i$, to validate a pending block proposed by another controller.

16 block$_{i,k}$ Controller$_i$.***Validate***(block$_{i',k}$).*Result*;

17 Data DAG ⇐ Controller$_i$.***Upload***(block$_{i,k}$);// where $Tree_{i,k}$ is contained in the block$_{i,k}$.

18 The uploading succeeds only if **Algo. 2** returns TRUE during the cross-validation of other forthcoming blocks.

Algorithm 5: Decentralized cross-validation mechanism

Output: TRUE or FALSE

▷ **Define**

Result Validator.***Validate***(Message); `// A validator validates some messages and return a bool result.`

▷ **Cross-Validation**

1 Result Controller$_i$.***Validate***(block$_{i',k}$.$Tree$), $i' \neq i$.

{

2 **if** *block$_{i',k}$ self-validating or lazy-validating* **then**

3 return FALSE

4 **for each** LoRa_data$_{i',j,k}$ in block$_{i',k}$.$Tree$ **do**

5 Result Controller$_i$.***Validate***(LoRa_data$_{i',j,k}$) {

`// Asymmetric-decrypting` LoRa_data$_{i',j,k}$

6 **if** *Gateway$_{i',j}$ info* $\notin$ *Blockchain$_{ID}$* **then**

7 **return** FALSE

`// Validate the PoTO protocol`

8 **if** $|data_{i,j,k}| <$ *lower-bound OR* $|data_{i,j,k}| \neq data\_size_{i',j,k}$ **then**

9 **return** FALSE

10 **if** $Hash(data_{i,j,k}) \neq data\_hash_{i',j,k}$ **then**

11 **return** FALSE

}

12 **end for each**

`// Validate the type-classification`

13 **if** *FlowControl is enabled* **then**

14 Invoke **Algo. 4.Cross-Validation for Flow Control**

15 **else**

16 **if** *Line 16 of Algo. 1 is NOT properly executed* **then**

17 **return** FALSE

18 **otherwise return** TRUE

}

Data uploading from LoRa Controllers to Data DAG: A controller first classifies the gateways which have completed the task into different types based on the data size that gateways have forwarded. The types are designed to fit in a tree structure, implying that type-1 contains the smallest number of gateways, and the number of gateways increases with the type index. The larger amount of data is received by the controller, the higher probability the gateway is classified in a type that is closer to the root of the tree. With a closer position to the root of the tree, the gateways are offered a higher incentive by the controller. The incentive is subsequently applied in a block which the controller has generated and broadcast. The block verifies a certain number of pending blocks in the *Data DAG* by pointing to their block hash values. The typical PoW is replaced by a more efficient protocol, PoTO; see cross-validation in the following.

Cross-Validation - The data uploading succeeds only if the pending block passes the cross-validation by all the participating controllers (or more specifically, validated by sufficient numbers of forthcoming blocks proposed by other controllers); see Algo. 5. By introducing a weighting factor to encourage blocks to cross-validate the tits of the DAG, the risk of a malicious controller conducting self-validation (i.e., validating its own previous blocks) and lazy-validation (i.e., validating any ancient blocks) can be reduced. Also, a minimum size of payloads is compulsory to prevent distributed denial-of-service (DDoS) attacks and Sybil attacks. Each transaction of the tree in a pending block can be associated with the LoRa data forwarded from each gateway to a specific controller for a published task. In regards to each transaction, other controllers can retrieve information, including the data size, data hash, and data payload after identifying the transaction, if the information of the gateway has been recorded on the *ID Chain*. Thus, the cross-validation can be conducted associated with the following aspects.

Identity (Lines 6-7 in Algo. 5): The information of each gateway and controller

is secured by the tamper-resistance of the *ID Chain*. The missing information of a gateway on the *ID Chain* leads to a possibility of the controller colluding with an unregistered gateway by packetizing its transaction into a pending block.

Proof-of-Task-Overhead (PoTO) (Lines 8-11 in Algo. 5): The PoW used in typical IOTA is a simple computational operation dedicated for spam protection and the defense to DDoS attacks [22]. Our proposed PoTO protocol, featuring the integration of Blockchain-LoRa, can delivery the same protection without need of additional computational operations required by the PoW used in IOTA. In particular, the computational operations can be omitted, as the limited data source transmitted in a physical channel, the limited data size, and the limited number of gateways inherently extend a period of time for every task. Even the malicious controllers would have to comply with the system restriction, thus leading to a controllable growth rate of the *Data DAG*. Based on this property, the amount of data received by the controllers (the data refers to the size of data payload of each transaction in a pending block on the *Data DAG*) are enough to be the amount of "work" done during the data collection and validation process. As a result, the PoTO protocol compromises the motivation to launch attacks. This is because controllers are not responsible for the validity of the metadata based on our Dual-Chain-based structure. Malicious controllers spending time and consuming communication (downloading/uploading) and computation (verifying/smart contract operations) resources to upload local corrupted LoRa data do not affect the interests of others and reap no profits for itself (See Section 6.3.1.3 for more details on the security analysis).

Type-classification (Lines 13-17 in Algo. 5): Recall that the rule of type-classification for each registered controller is recorded in the smart contract on the *ID Chain*. Global cross-validation regarding the type-classification can also be conducted to avoid any cheating, for example accepting bribes from a gateway and assigning it with a higher type than it deserves. Based on the data size from the last valida-

tion item, the other controllers can verify whether a specific controller has complied with the type-classification rule stored on the *ID Chain*. Thus, the incentive offered to each type of the gateways can subsequently be cross-validated by checking the balance of the gateways claimed by the pending block.

6.2.2 Flow Control Protocol in a self-driven way

In a typical LoRaWAN network, end-devices are connected to multiple LoRa Gateways, and hence the end-devices transmit data to multiple connected gateways [18]. The gateways transmit transactions to the corresponding controller, resulting in duplicated data at the controller. The duplicated data needs to be handled specifically because: 1) the *Data DAG* incurs a throughput loss and a data storage waste; and 2) the duplicated data still accounts for the uploading data size of the gateway during the type-classification, which can result in unfair competition. Thus, a discount applied to the duplicated data as the penalty is introduced to reduce unfair competition.

We propose a new flow control protocol which is designed to effectively allocate a fair and proper amount of incentive to each contributing gateway based on its uploading data size. Being part of the contract-theoretic incentive mechanism (see Section 6.2.3) to reduce the unfairness, the flow control is not aimed to control the transmission in a compulsory way (e.g., cutting off the transmission). Instead, it is a management tool to incentivize the gateways and make them comply with any preconcerted rules in a self-driven way, thus achieving the scalability as expected. Exceptions (e.g., sending urgent messages) are allowed. In other words, a gateway can still upload data anytime without caring about the flow control if the gateway has a strong wish to do so. The controller maintains an internal clock which determines a time-frame for every single gateway, while the controller can still receive the transmission from all the gateways. This motivates the gateway to transmit as

much as possible in their own time-frames, in order to reduce the ratio of duplicated data and maintain the participation of an adequate number of gateways in a specific region.

Algorithm 6: Flow Control: Congested LoRa Gateways and Incentive Allocation

▷ **Defining Congestion**

1 Controller$_i$ uploads randomness for $Task_{i,k+1}$ to the *ID Chain* using Blockchain-based randomness generator.

2 **while** [$Gateway_{i,j}$, ... ,$Gateway_{i,n}$] **do**

`// $\lambda_i$ denotes the overlapped proportion`

3 **if** ***Overlap***($Gateway_{i,j}$, $Gateway_{i,j+1}$)>λ_i **then**

4 Controller$_i$ records Gateway$_{i,j}$.$DAGAddr$, count the number of congested gateways n.

5 **for** *each* j *of* $Gateway_{i,j}$ **do**

6 $TimeStamp_j = TimeStamp_{start} + Timeout_{i,k} \times \frac{j}{n}$

7 Scheduler$_{i,k}$ Controller$_i$.***Scheduler***(Gateway$_{i,j}$.$DAGAddr$, $TimeStamp_j$)

▷ **Incentive Allocation**

if $TimeStamp_{j-1}$<$TimeStamp_{j,t} \leq TimeStamp_{j+1}$ **then**

8 Gateway$_{i,j,k}$ ←Controller$_i$.***Incentive***(|data$_{i,j,k}$|)

9 **else**

10 Gateway$_{i,j,k}$ ←Controller$_i$.***Incentive***($\frac{|\text{data}_{i,j,k}|}{n}$)

6.2.2.1 Congested LoRa Gateways

(Lines 1-7 of Algo. 6) Each gateway has a specific coverage area to serve end-devices. Multiple gateways in a region may have some overlapping coverage. Within the overlapping coverage, the end-devices connect to multiple gateways and are most likely to transmit duplicated data. Thus, we define the congested gateways as the gateways which have a specific common overlapping coverage area. A gateway registered on the *ID Chain* leads to the essential information, including the geographical location, device model, coverage area stored in a smart contract. The overlapped proportion is also defined in the smart contract, defined as the overlapped area divided by the total coverage area of a single gateway.

Through the overlapped proportion and the coverage area of each gateway recorded on the *ID Chain*, the corresponding controller is able to determine the congested gateways as a list $[Gateway_{i,j}, Gateway_{i,j+1}, \ldots, Gateway_{i,n}]$ where i indicates the i-th controller; j indicates the j-th gateway; and n indicates the number of congested gateways within a specific overlapping coverage). Also, a scheduler is defined in a smart contract and secured by the *ID Chain*. The scheduling for $Task_{i,k}$ is based on the pre-defined randomness announced on the *ID Chain* by using the Blockchain-based randomness generator such as RANDAO [2]. The scheduler records the identity and the assigned time-frame of each congested gateway. The controller refers to the scheduler determining whether a gateway transmits data within its own time-frame hence marked as a full contribution.

The contract-theoretic task-k published by the controller-i includes a timeout $Timeout_{i,k}$ which is a single epoch of the flow control. At the same time of publishing a task, a local timer is triggered on the controller and the initial timestamp $TimeStamp_{start}$ is recorded. Meanwhile, the controller checks the congested gateways based on the information (e.g., the overlapped proportion) of the gateways in the smart contract. The period of an epoch, $Timeout_{i,k}$, is divided into n time-frames which equals to the number of congested gateways. Therein, each boundary point is defined as the unique timestamp for each gateway-j $TimeStamp_j$ (Line 6 of Algo. 6)[3]. Finally, the controller updates every single timestamp $TimeStamp_j$ to the scheduler based on the identity of the congested gateway $Gateway_{i,j}.DAGAddr$. Note that using a single randomness for multiple tasks (i.e., multiple epochs) is permitted to prevent updating smart contracts for every task, and reduce the transaction load on the *ID Chain* which is typically poor at scalability.

[3]Generally, multiple gateways as a group can be allocated in one time-frame based on the live requirements. For simplicity, this chapter considers a one-gateway-one-timeframe scheme.

6.2.2.2 Incentive allocation

(Lines 8-10 of Algo. 6) Recall that the incentive allocated to each gateway is based on the uploading data size as the contribution. The flow control is an internal management tool for controllers to adjust the contribution of each gateway during a task, in order to allocate a proper incentive to the gateways and provide a fair type-based classification. The gateway can either check the *Data DAG* before transmitting transactions to reduce the ratio of duplicated data, or transmit without checking due to reasons such as saving query resources, enhance the data redundancy, or the willingness for sharing the coverage. Thus, transmitting data within its own time-frame is strongly encouraged with a high incentive, thus motivating gateways to reduce the ratio of duplicated data. In other words, if the transaction is uploaded by a gateway within its own time-frame, a full incentive is offered to the gateway based on its uploading data size. While if the transaction is uploaded by a gateway out of its corresponding time-frame, a partial incentive is offered based on the data size as the penalty. Here, the partial incentive is calculated by the original size of the uploading data divided by the number of congested gateways, n.

Algorithm 7: Flow control: decentralized cross-validation mechanism

Output: TRUE or FALSE

▷ **Cross-Validation for Flow Control**

Result Controller$_i$.***Validate***($Scheduler_{i',k}$), $i' \neq i$.
{
  if $Gateway_{i,j}.DAGAddr \notin Scheduler_{i,k}$ **then**
    return FALSE
  if *the pre-defined randomness is NOT matched with* $Scheduler_{i,k}$ **then**
    return FALSE
  if ***Algo. 3.Incentive Allocation*** *is NOT properly executed* **then**
    return FALSE
  otherwise return TRUE
}

6.2.2.3 Cross-validation of Flow Control

The above incentive allocation will not be conducted until passing the global cross-validation of flow control among other controllers. Recall that finalizing a pending block needs to validate a sufficient number of other pending blocks. By specifying the transactions in an arbitrary pending block uploaded to the *Data DAG*, all the other controllers can validate the transactions via their own proposed pending blocks, through the following aspects:

- *Identity of gateway* (Lines 2-3 of Algo. 7): The controllers validate whether the identity (e.g. Blockchain Address) belongs to one of the congested gateways based on the scheduler.
- *Scheduler* (Lines 4-5 of Algo. 7): The controllers validate whether the presented scheduler for a task is matched with the publicly pre-defined randomness.
- *Timestamp and Incentive* (Lines 6-7 of Algo. 7): The controllers compare the timestamp of the transaction with the time-frame of the gateway that is recorded in the scheduler. After that, the controllers calculate the amount of incentive and compare it with the intended amount.

6.2.3 Practical Implementation of the Incentive Mechanism

In regard to motivating more to participate in the deployment of LoRa Gateways to expand the coverage, we aim to design an incentive mechanism where the corresponding LoRa Controller will offer the token reward based on the amount of each gateway has contributed. However, a controller does not have any prior knowledge about the performance of gateways (e.g., the hardware performance) and the amount of LoRa data each gateway is willing to forward as a contribution. The information asymmetry between the controller and gateways needs to be tackled to

reduce the cost of the incentive while maximizing the utility of both controllers and gateways. In this chapter, we adopt contract theory [31] in the incentive mechanism design that can be integrated in a Blockchain-based LoRa system. In other words, the reward can be matched with how much each gateway should deserve in terms of the contribution without any bias, which guarantees efficiency and fairness. The whole process of the adoption of the contract theory is conducted during the contract initialization in the smart contract of *ID Chain* (referred to Section 6.2.1), and is discussed in the following.

During packetizing the block with task index k by a controller i chosen by the consensus process, a monopoly market [31] is considered. Therein, the market consists of a controller acting as the task publisher and a set of gateways $\mathbb{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_j, \ldots, \mathcal{N}_N\}, 1 \leq j \leq N$. For a gateway $\mathcal{N}_j$, the total amount of receiving data forwarded from the end-devices to the controller is denoted as $\mathbb{Q}_j^{i,k}$ for task-k published by controller-i. Any $\mathbb{Q}$ is resource-intensive in terms of the bandwidth, the performance of receiver, the number of supported bands, etc. To be specific, $\mathbb{Q}$ can be abstracted into $\mathbb{R}$ and t, denoting the single channel bandwidth (bps) and the usage (second) of this channel on this bandwidth during the maximum task period of task-k, respectively. Note that, (6.1) represents the vector of the logic channel characterized by a pair BW and SF (denoting the bandwidth in Hertz and the spreading factor), while CR denotes the code rate defining the level of tolerance to signal interference [18].

$$\mathbb{R} = \mathbb{R}(SF, BW, CR) = SF \times \frac{BW}{2^{SF}} \times CR. \tag{6.1}$$

Thus, we define F types based on the heterogeneous willingness in terms of the total uploading data size B. We assume that the gateways are sorted in an ascending order of the contributed data: $\theta_1 < \cdots < \theta_f < \cdots < \theta_F$. The greater $\theta_f, f \in \{1, \ldots, F\}$ implies the more LoRa data that this type of gateways have forwarded from the end-devices to the controller [91, 207]. Thus, each task $\mathbb{Q}_j^{i,k}$ is denoted by a two-tuple

$(\mathbb{R}_j^{i,k}, t_j^{i,k})$.

Each controller needs to encounter the information asymmetry while they aim to minimize the economic loss. For all F types θ_f customized by a controller in a pre-defined contract by a controller, the contract is a series of uplink-data-reward bundles denoted by $(T_f(B_f), B_f)$. B_f denotes the total uploading data size B of the type-θ_f gateways and $T_f(B_f)$ is the incentive offered to the gateways. Also note that, the gateways which forward more data from the end-devices to the corresponding controller under a valid threshold upper-bounding the uploading data size B (i.e., B_{max}) can be rewarded more.

6.2.3.1 Uploading data size in LoRa Gateways

We define the uploading data size B as the *significant* data size. Here, the term "*significant*" indicates that the received LoRa data is de-duplicated (i.e., repeated data being discounted as the penalty) after decoded at the controller side. This is due to the fact that multiple gateways may receive and forward identical messages sent from a single end-device either with the same or different logic channels [18]. In addition, we define B_f taking the following factors into consideration:

- the number of end-devices served by the type-θ_f gateways;
- the latency for the type-θ_f gateways to complete the task (receiving data and forwarding out) associated with both inbound and outbound bandwidth;
- the coverage area and the overlapping coverage area subject to the proposed flow control protocol (see Section 6.2.2);
- the amount of duplicated data.

We consider a specific region. For the j-th gateway completing the task-k, the significant data size $B_f(\mathbb{R}_j^k, t_j^k)$ for data forwarding task $\mathbb{Q}_j^k$ can be given by

$$B_f(\mathbb{R}_j^k, t_j^k) = \sum^{l} (\mathbb{R}_j^k t_j^k)_l, \tag{6.2}$$

where $\mathbb{R}_j^k$ and t_j^k denote the vector of logic channels and the time of transmission on each discrete logical channel out of total l channels during the task $\mathbb{Q}_j^k$, respectively. B_f denotes the size of uploading data from the end-devices to the gateways and should be upper-bounded by $B_{max} = \sum \mathbb{R}_j^k \mathbb{T}_{max}$, where $\mathbb{T}_{max}$ is the task period (i.e., an epoch period). B_f is equivalent to $|\text{data}_{i,j,k}|$ used in Algo. 6 for a specific significant data size sent from j-th gateway to its matched i-th controller during task-k. In (6.2), $t_j^k \leq \mathbb{T}_{max}$ and $B_f \leq B_{max}$. Note that the actual significant data size $B_f > B_{max}$ may happen due to maliciously forwarding accumulated data to the controller, which is detectable at the controller side by comparing B_{max} with the actual inbound uploading data size from gateways. This is because the link between gateways and controllers can be a fixed, wired, or backbone network. Thus, the total size of the accumulated data can be much bigger than B_{max}.

6.2.3.2 Utility of LoRa Controllers

Based on the signed contract (T_f, B_f) between a controller and the type-θ_f gateways, the utility of the controller earned through the task $\mathbb{Q}^k$ from the type-θ_f gateways is given by

$$U_c(\theta_f) = \mu(B_f) - \omega T_f, \tag{6.3}$$

where

$$\mu(B_f) = \begin{cases} \frac{\sigma}{(B_{max} - B_f)} & \text{if } B_{max} \geq B_f, \\ 0 & \text{otherwise.} \end{cases} \tag{6.4}$$

There is a penalty applied to the gateways uploading excessive data to the controller (i.e., $T_f(B_{max} < B_f) = 0$). Note that $\sigma > 0$ is a pre-defined parameter and (6.4)

implies B_f (closer to B_{max}) can attain a larger $\mu(B_f)$. Thus, the goal of the controller is to maximize its total utility all through the F types of gateways, as given by

$$\max_{(T_f,B_f)} \quad U_c = \sum_{f=1}^{F} p_f N(\mu(B_f) - \omega T_f), \forall f \in \{1,\ldots,F\}, \tag{6.5}$$

where N is the total number of gateways that the controller is in charge of; ω is a pre-defined parameter; and p_f is the prior probability of the type-θ_f gateways with $\sum_{f=1}^{F} p_f = 1$. Note that the controller can attain the distribution based on the historical statistics, and we assume a uniform distribution among the F types gateways which the controller is aware of [91, 207, 83]. The controller is also aware of the value of N, as a pre-registration on the *ID Chain* is needed to activate gateways in the coverage.

6.2.3.3 Utility of LoRa Gateways

For the type-θ_f gateways completing the task-k based on the signed contract (T_f, B_f), the utility function is given by

$$U_f = \theta_f \nu(T_f) - \phi B_f, \forall f \in \{1,\ldots,F\}, \tag{6.6}$$

where ϕ is the unit resource cost of data forwarding; $\nu(T_f)$ is the evaluation function of the type-θ_f gateways in terms of the incentive T_f. The evaluation function $\nu(T_f)$ monotonically increases with the following properties [31]:

- $\frac{\partial \nu}{\partial T_f} > 0$, monotonically increasing;
- $\frac{\partial^2 \nu}{\partial T_f^2} < 0$, concavity;
- $\frac{\partial \nu}{\partial \theta_f} > 0$, positive correlation of data contribution; and
- $\nu(0) = 0$, scheme for non-incentive.

The goal of all F types of gateways is to maximize the utility earned by data forwarding, as given by

$$\max_{(T_f,B_f)} \quad U_f = \theta_f \nu(T_f) - \phi B_f, \forall f \in \{1,\ldots,F\}. \tag{6.7}$$

Based on (6.5) and (6.7), our objective is to maximize the utility of the controllers and the utility of the gateways at the same time, while they are, in fact, contradictory. To solve the conflicting problem, the contract theory is used to design a series of optimal type specific contract (T_f^*,B_f^*); see more details on our published article [67].

6.2.3.4 Interaction between Dual-Chain to secure incentive

The following steps are conducted for the practical implementation of the contract-theoretic incentive mechanism, as shown in Fig. 6.2. A LoRa Controller, acting as the task publisher, obtains the values of any relevant information based on the historical data. Such information includes the LoRa settings i.e., (SF, BW, and CR), the number of LoRa Gateways under the management, and the expected value of significant uploading data size. Thus, the controller can calculate the optimal contract along with the types, and subsequently broadcasts this term to the gateways via the internet. By evaluating the utility based on the contract received from the controller, each of the gateways decides whether to participate in the task, and chooses one option in the contract, i.e., (T_f, B_f), by sending back feedback to claim its willingness for signing the contract with the controller. The above is recorded and updated with the status of smart contract in the *ID Chain*. Finally, after the gateways establish the task to forward the LoRa data from the end-devices to the controller with the agreed value of significant uploading data size, the controller proposes a new DAG block. Therein, the preconcerted incentive to each type of gateways can be paid based on the corresponding contractual obligation recorded in the contract of *ID Chain*.

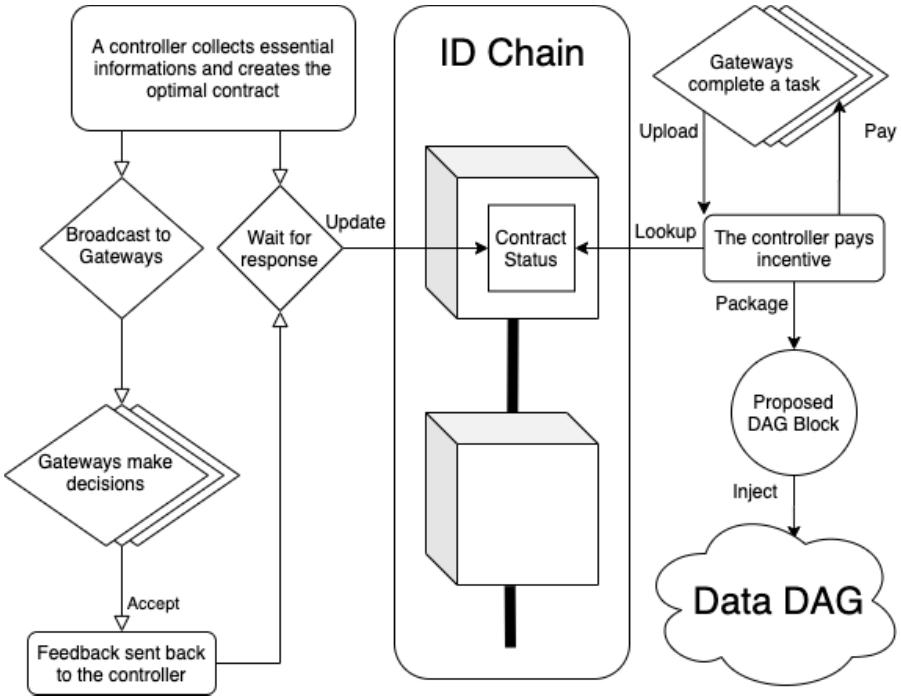


Figure 6.2 : The flow chart of practical implementation of the contract-theoretic incentive mechanism to relieve the information asymmetry

6.3 Simulation and Discussion

We carry out Monte-Carlo simulations in Python-3.8 to evaluate the impact on a LoRa network from the proposed flow control and incentive mechanism in a systematic view. We hereby define the following three terms:

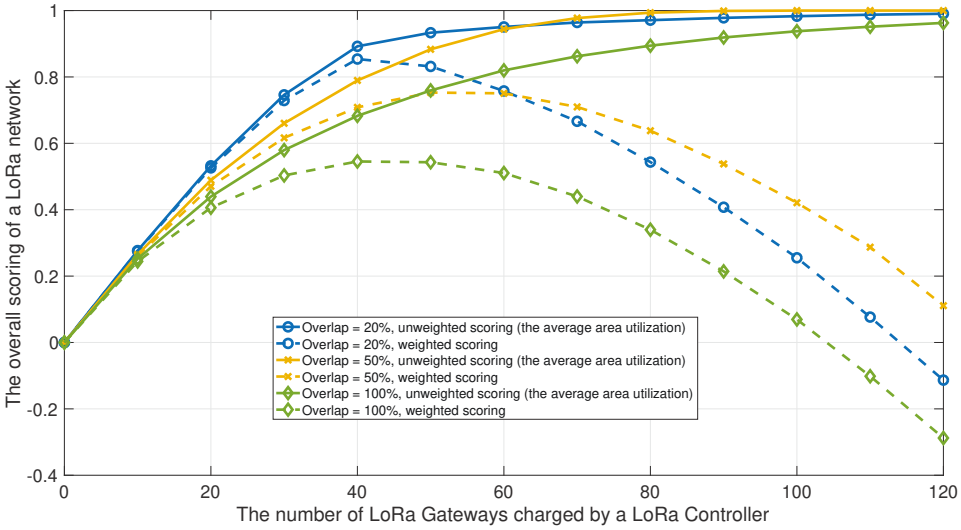
(Un)weighted Scoring: Excessive duplicated data severely degrades the throughput of *Data DAG* [107]. Specifically, transmitting a message to n different congested gateways results in the throughput of the *Data DAG* being reduced by n. In order to mitigate the loss, weighted scoring is introduced to assess the overall system by taking into account an expected loss rate associated with the overlapping coverage area among congested gateways, i.e., the total profit of the whole system. Accordingly, unweighted scoring is also introduced to represent the area utilization of a specific region managed by a single controller, i.e., the total coverage ratio.

Expected Loss Rate: The expected loss rate is the average loss rate applied to the gateways sending duplicated data to the controller. It is used to assess the impact of the proposed incentive mechanism applied to the overlapping coverage area in order to mitigate the throughput loss on the *Data DAG*. The overlapping coverage area is multiplied by the loss rate, which reflects a rapid decline in the overall weighted scoring because of the possible overlapping coverage areas of more than two gateways. A zero loss rate indicates that the negative impacts of the overlapping coverage area are not considered[4]. A lower loss rate implies a larger impact of the incentive mechanism applied to the overlapping coverage area.

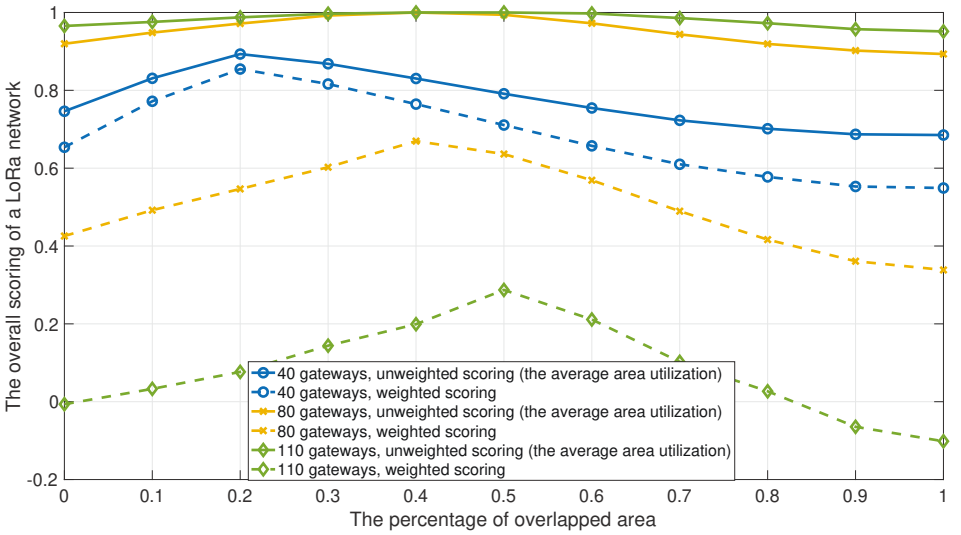
Ideal Overlapped Proportion: The maximum overlapping coverage area which can be accepted by a gateway being deployed in the case where there is enough leftover space within the region. Any owners will prefer to deploy their own gateways apart from an existing one in order to achieve the ideal overlapped proportion, until the gateways cannot be situated in such a position anymore due to the insufficient space. Gateways distributed in an overlapping coverage area have to comply with the flow control protocol conducted by the corresponding controller. A gateway tends to carefully maintain an overlapping coverage area with others such that it can enjoy more exclusive use of the significant throughput while still having a chance to compete with others. As such, traffic congestion can be relieved without having an excessive amount of duplicated data.

We consider an expected loss rate = 80% in the simulations. Figs. 6.3(a) and 6.3(b) show the trajectory of the overall (un)weighted scoring with the increasing number of gateways and the percentage of overlapping coverage area in a

[4]An expected loss rate = 1 indeed results in the overlook of negative effects of the overlapping coverage area. This is because, intuitively, the existing LoRa network designs take the data redundancy into account while rarely considering the throughput loss in a Blockchain-based LoRa network.



(a)



(b)

Figure 6.3 : (Left) The trajectory of the overall scoring along with the increasing number of LoRa Gateways changed by a LoRa Controller. (Right) The trajectory of the overall scoring along with the increasing percentage of the overlapping coverage area among gateways.

specific region, respectively. Every single scenario determined by the overlapping coverage area or the number of gateways is illustrated in a specific color, with the unweighted scoring and weighted scoring plotted as solid curve and dotted curve, respectively.

It is concluded in Fig. 6.3(a) that the unweighted scoring of overlap of 20% (the solid blue curve) reaches 90% area utilization the fastest at around 40 gateways while its weighted scoring (the dotted blue curve) remains the highest. In contrast, overlap of 100% (the solid green curve) implies the system takes no consideration about the negative effects of the overlapping coverage area. It reaches 90% area utilization at around 80 gateways and incurs the fastest decline of the weighted scoring (the dotted green curve). The unweighted scoring of overlap of 50% (the solid orange curve) reaches 90% area utilization by having only around 10 gateways more (at around 50 gateways) with the longest tail (the dotted orange curve). The longest tail implies that the scenario can afford the most number of gateways within the region while having the slowest decline of the weighted scoring (in other words, the total profit of the whole system). This encourages an energetic ecosystem where more users are willing to participate.

Fig. 6.3(b) reveals the same concept from another perspective. The scenario with 40 gateways exhibits the peaks on both the unweighted scoring (the solid blue curve) and weighted scoring (the dotted blue curve) at overlap of 20%. As the percentage of overlapping coverage area increases to around overlap of 50%, the peaks gradually shift to the point with overlap of 50% wherein there is an increasing number of participating gateways (from the orange curve to the green curve).

Along with the findings in Fig. 6.3, we can conclude that, introducing our new flow control and incentive mechanism can satisfy a Dual-Chain-based LoRa network requesting high throughput and flexibility. A smaller upper-bound of the overlapping

coverage area which needs to comply with the flow control (a more strict flow control) can result in a fast convergence to high area utilization while the total profit of the whole system remains high. On the other hand, if the goal is to encourage more gateways to participate, an upper-bound of the overlapping coverage area approaching 50% can balance the area utilization and the system profit very well.

6.3.1 Security Analysis

The security analysis focuses on the dual-chain-based structure and the proposed new PoTO protocol.

6.3.1.1 Dual-Chain Structure: Why Splitting Functions?

We split the functions between the *Data DAG* and *ID Chain*, respectively. The *Data DAG* is only responsible for the data storage, while the *ID Chain* is responsible for contract operations and payment functions. The proposed dual-chain strategy is different from those IOTA-like DAG-based structures where the payment functions is inherently supported. The strategy is particularly important in the context of LoRa. The reason is that a malicious controller may collude with all its gateways by sharing gateways' private keys, or simply generate a set of private keys and pretend to manage some gateways. As such, the controller can bypass the preconcerted incentive rule of the contract theory, and maximize its average block rate in the *Data DAG*, compared to other honest controllers which only initiate the transmission after collecting some data. As a result, the attacker can broadcast blocks which contain double-spending payment at the highest rate to induce the growth of *Data DAG* in its favour, thus breach the transaction order finality and compromise the balance system. It is thus significant to introduce a dual-chain structure that splits the payment function (whose physical meaning matters) out from the *Data DAG* and let the *ID Chain* handle the payment.

6.3.1.2 Dual-Chain Structure: The Performance Gap

The performance gap between a scalable DAG-based structure and an non-scalable chain-based structure may lead to network bottlenecks, data inconsistency, and corruption. Our proposed system enable the consistency between the *Data DAG* and *ID Chain*, even with an increasing number of end-devices and gateways. This can be achieved due to:

- **The scarcity of smart contract operations on the *ID Chain*:** Any smart contract operations on an *ID Chain* related to the devices registration/monitoring and contract initialization/status records are conducted much less frequently than publishing a single contract-theoretic task on an *Data DAG*. This is because these operations mostly send small data to update the status on smart contracts, and closely depend on the type-classification which can be conducted once and serve multiple tasks. Likewise, the randomness generated on the *ID Chain* can also be reused conservatively for multiple tasks during the self-driven flow control process.
- **The limited amounts of data transmitted on the *Data DAG*:** The block rate of *Data DAG* in LoRa networks is restrained by the task period, the data size of a task, and the number of gateways involved in a task.
  - **Limited task period:** The period of any smart contract operations related to incentive payment and balance records correspond to the unit period of a published task, as the functionality is in the charge of the *ID Chain*. Such period is restrained by an upper-bounded frequency of publishing new blocks to the *Data DAG* based on the anti-spam features, i.e., PoTO protocol.
  - **Limited data size:** The data size uploaded by a gateway during a task is restrained by the physical channel resource according to Equations (6.1)

and (6.2), including spreading factor, bandwidth, code rate, duty cycle, etc.

  - **Limited number of gateways:** It is found in Figure 4 that the number of self-deployed gateways in a region tends to fall into a certain range and maximize the utility of the LoRa network, considering that the (un)weighted scoring takes effect for the penalty of the uploaded data size.

- **The advanced technologies implemented to improve the *ID Chain*:** The performance gap can be eliminated by implementing advanced technologies to improve the performance of the *ID Chain*, such as the HotStuff consensus algorithm that enables high transaction rate among large communities, and the sharding technology that enables the horizontal scalability. Moreover, requesting a higher security level on the *Data DAG* needs each node to download as deep *Data DAG* as possible for more secure validation. This makes the *Data DAG* no better than the improved *ID Chain*, and stops the *Data DAG* from being scaled out. One feasible solution is to implement the sharding technology (as suggested in Chapter 4) to both the *Data DAG* and *ID Chain*.

6.3.1.3 PoTO Protocol: The Improved Spam Protection

The proposed system inherently enables the spam protection which, otherwise, would have to be achieved by a typical PoW process in IOTA. The PoW used in IOTA is not a typical PoX-based consensus algorithm (as suggested in Chapter 3). It is a comparably simple computational operation which differs from the expensive PoW conducted in many miner-based Blockchains. Instead of racing for the winner of each consensus round, the defense to DDoS attacks and spam protection are the key goals of the PoW in IOTA. Our proposed PoTO protocol can deliver the same protection without need of additional computational operations required by

the PoW used in IOTA, as clarified below from the perspectives of system restriction and attack motivation.

From the perspective of system restriction,

- **Limited data source:** Each LoRa Gateway needs to be registered on the *ID Chain* and generate the unique private key for digital signature, prior to granting permission to participate in the LoRa networks. The controllers can mutually verify each of the identities of transactions in a DAG block to ensure the transactions are indeed sourced from a certain and finite gateway set.
- **Limited data size and number of gateways:** The data size uploaded by a gateway during a task is restrained by the physical channel resource according to Equations (6.1) and (6.2). Also, the number of self-deployed gateways in a region falls into a certain range to maximize the entire utility.

In our system, the controllers incentivize the self-deployment of gateways. The controllers and gateways maximize the utilities, i.e., Equations (6.5) and (6.7). A controller would not be able to spam the network unless it colludes with all its gateways. In the worst-case scenario where the collusion happens, the system restriction can extend a period of time before a task is finalized, thus significantly reducing the risk of DDoS attacks with a compulsory minimum size of payloads. This leads to a controllable growth rate of the DAG , since the block rate depends on the data source, data size, and the number of gateways.

From the perspective of attack motivation, as suggested in Section 6.2, the function of incentive payment is conducted on the *ID Chain* in our proposed dual-chain system. This leads to independence among controllers, and the controllers are only responsible for the validity of on-chain LoRa data stored on the *Data DAG*, and not for the validity of the metadata. Each individual controller aims to enhance the quality of local LoRa business. Spending time and consuming communication

(downloading/uploading) and computation (verifying/smart contract operations) resources to upload local corrupted LoRa data do not affect the interests of others and reap no profits for itself. Therefore, for LoRa Controllers, the communication and computation overhead is enough to be the amount of "work" done during the data collection and validation process, i.e., PoTO, without need of additional computation operations.

6.4 Conclusions

Remark that the increasing number of studies which presented simple and stilted integration between Blockchain and IoT technologies without dedicated optimization has hindered the adoption of Blockchain-based IoT applications. As a solid study to demonstrate how the optimizations can be designed, in this chapter, we proposed a new Dual-Chain-based LoRa system where the state-of-the-art contract-theoretic incentive mechanism for self-driven deployment, and data storing service can be secured by a decentralized global cross-validation with the tamper-resistance of Blockchain. The contract-theoretic incentive mechanism consists of a new self-driven flow control protocol that mitigates the traffic congestion and throughput loss of the Blockchain due to uploading the duplicated data, scaling the Dual-Chain system to achieve high throughput. As a result, each gateway is motivated to be deployed in its appropriate place to maximize the entire system throughput while reducing the data replication without the need to modify existing LoRaWAN protocols.

Chapter 7

Contributions and Future Work

This chapter summarizes the research in this thesis ahead of the contributions of each chapter. Finally, we point out the future work that potentially improves and extends our current work.

7.1 Contributions

In regard to the four highlighted aspects, this thesis has presented the corresponding novel achievements ranged from new analytic models to new system architecture designs. A new unified analytical model for PoX consensus scheme in large-scale networks was proposed in Chapter 3, while a comprehensive comparison and evaluation among existing major sharding mechanisms was presented in Chapter 4. This thesis also introduced new system architecture designs for Blockchain-based fine-grained access control and Blockchain-based LoRa networks in Chapters 5 and 6, respectively. To be specific, the novelty and contributions of the thesis is summarized as follows.

A new unified analytical model was proposed to fill in some piece of gaps in the literature of PoX consensus scheme used in large-scale IoT networks. The new unified analytical model was able to quantify the profit of individual miners in any of the popular permissionless PoX-based Blockchains. The new model captured proposed changes in the system resource distribution of PoX schemes by designing an infinite-dimensional Markov chain. A set of expressions was established to efficiently evaluate the mining probability of a miner, given the amount of system re-

source owned by the miner. The type and distribution of system resources could be customized in line with system requirements. As revealed by our analysis, in PoX-based consensus algorithms where the monopoly of block generation is prevented and diversity is maintained, miners can maximize the profits with strong double-spending-resistance and controllable cost-risk assessment, thereby contributing to a healthy and sustainable mining ecosystem.

The sharding technology has proven its importance solving the scalibility issue of Blockchains. A comprehensive comparison and evaluation was presented to summarize the pros and cons of existing major sharding mechanisms, as well as the possible future improvements. To be specific, Nakamoto-based sharding mechanism (Monoxide) stands on the stage, for the first time pointing out a different direction compared with the BFT-based sharding mechanisms. OmniLedger, Rapidchain, and Ethereum 2.0 attempted to implement scalable or high-FT BFT-based consensus algorithms rather than the traditional PBFT for the intra-consensus safety. Also, a smart-contract-oriented sharding mechanism was proposed by Chainspace, for the first time supporting the complicated conditional transactions. All considered sharding mechanisms also introduced the optimizations to address the new challenges their proposed schemes pose to the system, e.g., latency, storage, fair randomness generator, and replay-attack defense, but further improvements are necessary. Finally, we pointed out the common pitfalls in existing sharding mechanisms, including the use of a global chain, the needs for validators to store unrelated records, and business-oriented members allocation. We also pointed out the future trend, including the scaling of randomness generator, extending Nakamoto-based sharding to general PoX schemes, balancing the uses of stochastic and biased members allocation, and enabling more effective scheme for smart contracts.

A novel multi-layer Blockchain-IoT data service system was proposed to enable secure attribute updates in an ABE-based fine-grained access control mechanism

for Blockchains. We developed a redactable key chain along with a standard data chain to secure and control the access to the data chain. Empowered by redactable hash functions, the redactable key chain allowed the access policies of ABE to be updated by key chain miners. The data chain could be any existing Blockchain with any scalable structure and preserves the immutability of the IoT data. Collectively, these crytographic primitives addressed the inherent incompatibility between the immutability of Blockchains and the indispensable need of updating attributes to manage the access to Blockchains. The system was compatible with the major types of cryptographic primitives, consensus algorithms, and cross-chain protocols in Blockchains. The analysis and simulation results showed that our proposed mechanism is able to outperform existing solutions in terms of overhead, searching complexity, and security.

A novel Dual-Chain-based LoRa system was proposed as a case study of optimizing Blockchain-based IoT system architecture. The system took advantage of the DAG structure along with a typical chain-based structure that provides identity registration and protocol monitoring in smart contracts. By interacting with both the DAG and identity chain and leveraging decentralized global cross-validation, the processing of any incentive mechanisms (including the proposed contract-theoretic incentive mechanism) can be secured. The contract-theoretic incentive mechanism featured a new self-driven flow control protocol that scales the LoRa network by relieving traffic congestion, and improves the throughput of the Blockchain, scaling the Dual-Chain system. As a result, the system enables: 1) strong compatibility with typical LoRaWAN protocols; 2) the PoTO, a new spam protection dedicated for LoRa networks that wastes fewer resource; 3) efficient and flexible data storage services at any time with high throughput; and 4) the transparency and fairness of incentive mechanism and data storage services because of the tamper-resistance property of the decentralized cross-validation protocol. As revealed by our analysis

and simulation results, the proposed Dual-Chain-based LoRa system can significantly improve the throughput of the Blockchain, while maintaining a high area utilization.

7.2 Future Work

With a series of proposed models, surveys, and designs, this thesis has analyzed key stages of Blockchain-based IoT systems and deduces significant results. However, extensions and improvements are seen as the future work to cover more aspects in this field.

The unified PoX schemes analytical model cannot capture the non-negligible delay. In the future, we will optimize the margin of error of the model and study the short-term impact of the accumulated resource of each miner upon the entire system. Moreover, we will involve more existing PoX schemes in the analysis and investigate new PoX schemes based on the benchmark to meet diversified contexts.

The sharding technology has been well developed for the past years and still being rapidly evolved. Our proposed survey needs to consider more perspectives, such as the sharding-related replay attacks and defenses, the pros and cons of using business-driven member allocation, etc. We will also investigate new sharding mechanisms based on the trends/suggestions to meet diversified contexts.

Current proposed ABE-based fine-grained access control mechanism for Blockchains still relies on a centralized TA. One potential solution is to decentralize the TAs in the system such as investigating new decentralized key generation and distribution dedicated for Blockchain contexts.

Several studies regarding the decentralized TA have been proposed to avoid the reliance on a centralized TA in conventional ABE-based fine-grained access control mechanisms [108, 78, 79, 44]. All of them follow similar concepts that multiple

authorities can independently issue keys to users without needs of knowing other authorities and cooperate with others, while users can encrypt data in terms of any boolean formula over attributes issued from any chosen set of authorities. However, none of them are able to tackle the issue of exposing the secret information of a TA. If one assigned with only one attribute that comes from a single TA, the TA exposing the secret information risks the user being attacked, which would not be capable of Blockchain contexts. One potential Blockchian-driven solution is to decentralize any TAs by using secure Multi-Party Computation (sMPC) to split a TA's secret information among multiple parties, and conduct functions without exposing the actual information [88], which desires for more future studies and investigation.

For optimizing Blockchain-based IoT architecture, our Blockchain-based LoRa system can be applied to a concrete application and evaluate its performance in the future. Our system can also be extended to more generalized LPWAN networks. In addition, the overhead would have to increase to $O(n)$ (a network sized of n nodes) as the Data DAG grows. Possible optimization of computation, communication, and storage complexity by using sharding technologies discussed in Chapter 4 can be integrated to the system. However, any potential implications of the cross-shard transactions in a DAG-driven sharding mechanisms are still desperate for further studies.

Bibliography

[1] "Raiden network," 2015. [Online]. Available: https://raiden.network/

[2] (2017) Randao: Verifiable Random Number Generation. [Online]. Available: https://www.randao.org/whitepaper/Randao_v0.85_en.pdf

[3] "Reddcoin," 2018. [Online]. Available: https://wiki.reddcoin.com/Main_Page

[4] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[5] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, "Chainspace: A sharded smart contracts platform," *CoRR*, vol. abs/1708.03778, 2017. [Online]. Available: http://arxiv.org/abs/1708.03778

[6] M. Al-Bassam, A. Sonnino, and V. Buterin, "Fraud proofs: Maximising light client security and scaling blockchains with dishonest majorities," *CoRR*, vol. abs/1809.09044, 2018. [Online]. Available: http://arxiv.org/abs/1809.09044

[7] Alex, "How much power does the pi4b use? power measurements." [Online]. Available: https://www.raspberrypi-spy.co.uk/2018/11/raspberry-pi-power-consumption-data

[8] L. Alliance, "LoRaWAN Specification v1.0," 2015, accessed on 16.04.2020. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-05/2015_-_lorawan_specification_1r0_611_1.pdf

[9] N. Alon, H. Kaplan, M. Krivelevich, D. Malkhi, and J. Stern, "Scalable secure storage when half the system is faulty," in *Automata, Languages and Programming*, U. Montanari, J. D. P. Rolim, and E. Welzl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 576–587.

[10] ——, "Addendum to "scalable secure storage when half the system is faulty"[inform. comput. 174 (2)(2002) 203–213]," *Information and Computation*, vol. 205, no. 7, pp. 1114–1116, 2007.

[11] M. Ambrosin, A. Anzanpour *et al.*, "On the feasibility of attribute-based encryption on internet of things devices," *IEEE Micro*, vol. 36, no. 6, pp. 25–35, Nov 2016.

[12] M. Ambrosin, M. Conti, and T. Dargahi, "On the feasibility of attribute-based encryption on smartphone devices," in *Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems*, ser. IoT-Sys '15. New York, NY, USA: ACM, 2015, pp. 49–54. [Online]. Available: http://doi.acm.org/10.1145/2753476.2753482

[13] I. Analytics, "LPWAN Market Report 2018-2023," 2018, accessed on 16.04.2020. [Online]. Available: https://iot-analytics.com/product/lpwan-market-report-2018-2023/

[14] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger Fabric: a Distributed Operating System for Permissioned Blockchains," in *Proc. of the 13th EuroSys Conf. (EuroSys '18)*. ACM, Apr. 2018, p. 30.

[15] B. Anggorojati, P. N. Mahalle *et al.*, "Capability-based access control delegation model on the federated iot network," in *The 15th International*

Symposium on Wireless Personal Multimedia Communications, Sep. 2012, pp. 604–608.

[16] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, "Redactable blockchain – or – rewriting history in bitcoin and friends," in *2017 IEEE European Symposium on Security and Privacy (EuroS P)*, April 2017, pp. 111–126.

[17] G. Ateniese and B. de Medeiros, "On the key exposure problem in chameleon hashes," in *Security in Communication Networks*, C. Blundo and S. Cimato, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 165–179.

[18] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of lora: Long range amp; low power networks for the internet of things," *Sensors*, vol. 16, no. 9, 2016. [Online]. Available: https://www.mdpi.com/1424-8220/16/9/1466

[19] B. Awerbuch and C. Scheideler, "Towards a scalable and robust dht," *Theory of Computing Systems*, vol. 45, no. 2, pp. 234–260, Aug 2009. [Online]. Available: https://doi.org/10.1007/s00224-008-9099-9

[20] AWS, "Summary of the amazon s3 service disruption in the northern virginia (us-east-1) region," https://aws.amazon.com/message/41926/, 2017.

[21] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille, "Enabling blockchain innovations with pegged sidechains," *URL: http://www. opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains*, p. 72, 2014.

[22] B. Baek, "Iota: A cryptographic perspective," 2019.

[23] L. Baird, "The swirlds hashgraph consensus algorithm: Fair, fast, byzantine

fault tolerance," *Swirlds Tech Reports SWIRLDS-TR-2016-01, Tech. Rep.*, 2016.

[24] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake [Extended Abstract]Y," *SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, pp. 34–37, Dec. 2014. [Online]. Available: http://doi.acm.org/10.1145/2695533.2695545

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

[26] E. Bertino and N. Islam, "Botnets and internet of things security," *Computer*, vol. 50, no. 2, pp. 76–79, 2017.

[27] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, May 2007, pp. 321–334.

[28] A. Bhutani and P. Wadhwani, "Global LPWAN Market Size worth over $65 Bn by 2025," 2019, accessed on 16.04.2020. [Online]. Available: https://www.gminsights.com/pressrelease/lpwan-market

[29] G. BitFury, "Proof of stake versus proof of work," *White paper, Sep*, 2015.

[30] block.one. (2018, Mar.) Eos.io technical white paper v2. [Online]. Available: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md

[31] P. Bolton, M. Dewatripont *et al.*, *Contract theory*. MIT press, 2005.

[32] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch, "Verifiable delay functions," Cryptology ePrint Archive, Report 2018/601, 2018, https://eprint.iacr.org/2018/601.

[33] D. Boneh, M. Drijvers, and G. Neven, "Compact multi-signatures for smaller blockchains," in *Advances in Cryptology – ASIACRYPT 2018*, T. Peyrin and S. Galbraith, Eds. Cham: Springer International Publishing, 2018, pp. 435–464.

[34] O. Brocaar, "ChirpStack Network Server," 2020, accessed on 16.04.2020. [Online]. Available: https://github.com/brocaar/chirpstack-network-server

[35] V. Buterin, "The Problem of Censorship," 2015, accessed on 01.08.2019. [Online]. Available: https://blog.ethereum.org/2015/06/06/the-problem-of-censorship/

[36] ——, "Chain interoperability," *R3 Research Paper*, 2016.

[37] ——, "Ethereum sharding FAQ," Apr. 2019, accessed on 01.08.2019. [Online]. Available: https://github.com/ethereum/wiki/wiki/Sharding-FAQ

[38] V. Buterin *et al.*, "Ethereum white paper: a next generation smart contract & decentralized application platform," *First version*, 2014.

[39] J. Camenisch, D. Derler *et al.*, "Chameleon-hashes with ephemeral trapdoors," in *Public-Key Cryptography – PKC 2017*, S. Fehr, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 152–182.

[40] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[41] R. Cattell, "Scalable sql and nosql data stores," *Acm Sigmod Record*, vol. 39, no. 4, pp. 12–27, 2011.

[42] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography*, S. P. Vadhan, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 515–534.

[43] H. Chen and Y. Wang, "Sschain: A full sharding protocol for public blockchain without data migration overhead," *Pervasive and Mobile Computing*, vol. 59, p. 101055, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119218306370

[44] J. Chen and H. Ma, "Efficient decentralized attribute-based access control for cloud storage with user revocation," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 3782–3787.

[45] S. Chen, W. Dai, Y. Dai, H. Fu, Y. Gao, J. Guo, H. He, and Y. Liu, "Thinkey: A scalable blockchain architecture," *CoRR*, vol. abs/1904.04560, 2019. [Online]. Available: http://arxiv.org/abs/1904.04560

[46] Y. Chen, W. Sun *et al.*, "Towards efficient fine-grained access control and trustworthy data processing for remote monitoring services in iot," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1830–1842, July 2019.

[47] A. Chepurnoy, "Interactive proof-of-stake," *arXiv preprint arXiv:1601.00275*, 2016.

[48] A. Churyumov, "Byteball: A decentralized system for storage and transfer of value," *URL https://byteball. org/Byteball. pdf*, 2016.

[49] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, and D. Woodford, "Spanner: Google’s globally distributed database," *ACM Trans. Comput. Syst.*, vol. 31, no. 3, pp. 8:1–8:22, Aug. 2013. [Online]. Available: http://doi.acm.org/10.1145/2491245

[50] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains," in *Financial Cryptography and Data Security*, J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 106–125.

[51] J. Daemen and V. Rijmen, "Aes proposal: Rijndael," 1999.

[52] H. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.

[53] G. Danezis and S. Meiklejohn, "Centrally banked cryptocurrencies," *CoRR*, vol. abs/1505.06895, 2015. [Online]. Available: http://arxiv.org/abs/1505.06895

[54] S. M. Danish, M. Lestas, W. Asif, H. K. Qureshi, and M. Rajarajan, "A lightweight blockchain based two factor authentication mechanism for lorawan join procedure," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2019, pp. 1–6.

[55] S. R. Danning Sui and J. Pfeffer, "Are Miners Centralized? A Look into Mining Pools," 2018, accessed on 22.10.2019. [Online]. Available: https://media.consensys.net/are-miners-centralized-a-look-into-mining-pools-b594425411dc

[56] J. Debus, "Consensus Methods in Blockchain Systems," Frankfurt School of Finance & Management, Blockchain Center, Frankfurt am Main, Germany, Tech. Rep., 2017.

[57] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *IEEE P2P 2013 Proceedings*, 2013, pp. 1–10.

[58] S. Delbruel, N. Small, E. Aras, J. Oostvogels, and D. Hughes, "Tackling contention through cooperation: A distributed federation in lorawan space," in *Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks*. Junction Publishing, 2020, pp. 13–24.

[59] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*, ser. SIGMOD '17. New York, NY, USA: ACM, 2017, pp. 1085–1100. [Online]. Available: http://doi.acm.org/10.1145/3035918.3064033

[60] J. Dutta and S. Roy, "Iot-fog-cloud based architecture for smart city: Prototype of a smart building," in *2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence*, 2017, pp. 237–242.

[61] W. F. Elsadek, "Toward hyper interconnected iot world using sdn overlay network for ngn seamless mobility," in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2016, pp. 460–463.

[62] I. Eyal, A. E. Gencer, E. G. Sirer, and R. V. Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. Santa Clara, CA: USENIX Association, Mar. 2016, pp. 45–59. [Online]. Available: https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eyal

[63] I. Factory, "Best LORAWAN Network Servers," 2020, accessed on 16.04.2020. [Online]. Available: https://iotfactory.eu/products/software-platform/best-lorawan-network-servers/

[64] G. Fedrecheski, L. C. C. De Biase *et al.*, "Attribute-based access control for the swarm with distributed policy management," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 1, pp. 90–98, Feb 2019.

[65] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, Oct 1987, pp. 427–438.

[66] L. D. Feo, S. Masson, C. Petit, and A. Sanso, "Verifiable delay functions from supersingular isogenies and pairings," Cryptology ePrint Archive, Report 2019/166, 2019, https://eprint.iacr.org/2019/166.

[67] X. W. K. Y. N. W. J. Z. G. Yu, L. Zhang and R. P. Liu, "A novel dual-blockchained structure for contract-theoretic lora-based information systems," 11 2010.

[68] J. Garay, A. Kiayias, and N. Leonardos, "The Bitcoin Backbone Protocol: Analysis and Applications," in *Annu. Int. Conf. on the Theory and Appl. of Cryptographic Techn. (EUROCRYPT '15)*. Berlin, Heidelberg: Springer, Apr. 2015, pp. 281–310.

[69] J. Garzik, "Bip102: Block size increase to 2mb," 2015. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0102.mediawiki

[70] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 3–16. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978341

[71] 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, 2017, pp. 51–68.

[72] P. Gotthard, "Compact server for private LoRaWAN networks," 2020, accessed on 16.04.2020. [Online]. Available: https://github.com/gotthardp/lorawan-server

[73] V. Gramoli, "From blockchain consensus back to byzantine consensus," *Future Generation Comput. Syst.*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17320095

[74] J. Gray *et al.*, "The transaction concept: Virtues and limitations," in *VLDB*, vol. 81, 1981, pp. 144–154.

[75] J. Guo, I. Chen, J. J. P. Tsai, and H. Al-Hamadi, "A hierarchical cloud architecture for integrated mobility, service, and trust management of service-oriented iot systems," in *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, 2016, pp. 72–77.

[76] K. Gupta and S. Shukla, "Internet of things: Security challenges for next generation networks," in *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, 2016, pp. 315–318.

[77] T. Haerder and A. Reuter, "Principles of transaction-oriented database recovery," *ACM computing surveys (CSUR)*, vol. 15, no. 4, pp. 287–317, 1983.

[78] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2150–2162, 2012.

[79] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. Au, "Ppdcp-abe: Privacy-preserving decentralized ciphertext-policy attribute-based

encryption," in *Computer Security - ESORICS 2014*, M. Kutyłowski and J. Vaidya, Eds. Cham: Springer International Publishing, 2014, pp. 73–90.

[80] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C.: USENIX Association, Aug. 2015, pp. 129–144. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/heilman

[81] S. Hohenberger and B. Waters, "Realizing hash-and-sign signatures under standard assumptions," in *Advances in Cryptology - EUROCRYPT 2009*, A. Joux, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 333–350.

[82] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, "Zcash Protocol Specification," Zerocoin Electric Coin Company, Lakewood, CO, USA, Tech. Rep., Jan. 2016.

[83] Z. Hou, H. Chen, Y. Li, and B. Vucetic, "Incentive mechanism design for wireless energy harvesting-based internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2620–2632, Aug 2018.

[84] K. Huang, X. Zhang *et al.*, "Building redactable consortium blockchain for industrial internet-of-things," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.

[85] IBM, "Transaction confidentiality," 2016. [Online]. Available: https://openblockchain.readthedocs.io/en/latest

[86] IDC, "Number of internet of things (iot) connected devices worldwide from 2019 to 2030," https:

//www.statista.com/statistics/1183457/iot-connected-devices-worldwide/, 2020.

[87] ——, "Prognosis of worldwide spending on the internet of things (iot) from 2018 to 2023," https://www.statista.com/statistics/668996/worldwide-expenditures-for-the-internet-of-things/, 2020.

[88] B. Jayaraman, H. Li, and D. Evans, "Decentralized certificate authorities," *CoRR*, vol. abs/1706.03370, 2017. [Online]. Available: http://arxiv.org/abs/1706.03370

[89] R. Jhawar and V. Piuri, "Fault tolerance management in iaas clouds," in *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*, Oct 2012, pp. 1–6.

[90] M. Jourenko, K. Kurazumi, M. Larangeira, and K. Tanaka, "Sok: A taxonomy for layer-2 scalability related protocols for cryptocurrencies." *IACR Cryptology ePrint Archive*, vol. 2019, p. 352, 2019.

[91] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, March 2019.

[92] A. Kiayias, N. Lamprou, and A.-P. Stouka, "Proofs of proofs of work with sublinear complexity," in *Financial Cryptography and Data Security*, J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 61–78.

[93] A. Kiayias, A. Miller, and D. Zindros, "Non-interactive proofs of proof-of-work." *IACR Cryptology ePrint Archive*, vol. 2017, no. 963, pp. 1–42, 2017.

[94] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual International Cryptology Conference*. Springer, 2017, pp. 357–388.

[95] ——, "Ouroboros: A provably secure Proof-of-Stake Blockchain Protocol," in *The 37th Annu. Int. Cryptology Conf. (CRYPTO '17)*. Springer, 2017, pp. 357–388.

[96] S. King and S. Nadal, "PPcoin: peer-to-peer crypto-currency with proof-of-stake," Aug. 2012. [Online]. Available: https://pdfs.semanticscholar.org/0db3/8d32069f3341d34c35085dc009a85ba13c13.pdf

[97] ——, "PPcoin: peer-to-peer crypto-currency with proof-of-stake," Aug. 2012. [Online]. Available: https://pdfs.semanticscholar.org/0db3/8d32069f3341d34c35085dc009a85ba13c13.pdf

[98] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 279–296. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kogias

[99] E. Kokoris-Kogias, P. Jovanovic *et al.*, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in *2018 IEEE Symposium on Security and Privacy (SP)*, May 2018, pp. 583–598.

[100] H. Kopp, C. Bösch, and F. Kargl, "Koppercoin–a distributed file storage with financial incentives," in *The 12th Int. Conf. on Inform. Security Practice and Experience (ISPEC 2016)*. Cham: Springer, Nov. 2016, pp. 79–93.

[101] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: speculative byzantine fault tolerance," in *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6. ACM, 2007, pp. 45–58.

[102] ——, "Zyzzyva: Speculative byzantine fault tolerance," *ACM Trans. Comput. Syst.*, vol. 27, no. 4, Jan. 2010. [Online]. Available: https://doi.org/10.1145/1658357.1658358

[103] J. Lai, R. H. Deng, and Y. Li, "Fully secure cipertext-policy hiding cp-abe," in *Information Security Practice and Experience*, F. Bao and J. Weng, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 24–39.

[104] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

[105] A. Lavric and V. Popa, "Internet of things and lora™ low-power wide-area networks: A survey," in *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*, 2017, pp. 1–5.

[106] O. Leiba, Y. Yitzchak, R. Bitton, A. Nadler, and A. Shabtai, "Incentivized delivery network of iot software updates based on trustless proof-of-distribution," *CoRR*, vol. abs/1805.04282, 2018. [Online]. Available: http://arxiv.org/abs/1805.04282

[107] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, "Inclusive block chain protocols," in *Financial Cryptography and Data Security*, R. Böhme and T. Okamoto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 528–547.

[108] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2011, pp. 568–588.

[109] C. Li, P. Li, D. Zhou, W. Xu, F. Long, and A. Yao, "Scaling nakamoto consensus to thousands of transactions per second," *arXiv preprint arXiv:1805.03870*, 2018.

[110] R. Li, H. Asaeda, and J. Li, "A distributed publisher-driven secure data sharing scheme for information-centric iot," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 791–803, June 2017.

[111] R. Li, T. Song *et al.*, "Blockchain for large-scale internet of things data storage and protection," *IEEE Transactions on Services Computing*, 2018.

[112] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[113] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 2017, pp. 297–315.

[114] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[115] J. Lin, Z. Shen, and C. Miao, "Using blockchain technology to build trust in sharing lorawan iot," in *Proceedings of the 2nd International Conference on Crowd Science and Engineering*, ser. ICCSE'17. New York, NY, USA: Association for Computing Machinery, 2017, p. 38–43. [Online]. Available: https://doi.org/10.1145/3126973.3126980

[116] Q. Lin, H. Yan *et al.*, "An id-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, vol. 6, pp. 20 632–20 640, 2018.

[117] LinkTime. Justin Drake-Ethereum, Sharding. Youtube. [Online]. Available: https://www.youtube.com/watch?v=J4rylD6w2S4

[118] S. Lo and J. C. Wang, "Bitcoin as money?" Federal Reserve Bank of Boston, Current Policy Perspectives 14-4, 2014. [Online]. Available: https://EconPapers.repec.org/RePEc:fip:fedbcq:2014_004

[119] E. Lombrozo, J. Lau, and P. Wuille, "Bip141: Segregated witness (consensus layer)," 2015.

[120] 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*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 17–30. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978389

[121] D. M and N. B. Biradar, "Iota-next generation block chain," *International Journal of Engineering and Computer Science*, vol. 7, no. 04, pp. 23 823–23 826, Apr. 2018. [Online]. Available: http://103.53.42.157/index.php/ijecs/article/view/4007

[122] M. Ma, G. Shi, and F. Li, "Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the iot scenario," *IEEE Access*, vol. 7, pp. 34 045–34 059, 2019.

[123] S. Marco Colino, "The antitrust f word: Fairness considerations in competition law," *Journal of Business Law, Forthcoming*, 2018.

[124] P. Massonet, L. Deru, A. Achour, S. Dupont, A. Levin, and M. Villari, "End-to-end security architecture for federated cloud and iot networks," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2017, pp. 1–6.

[125] MATT, "Raspberry pi power consumption data." [Online]. Available: https://raspi.tv/2019/how-much-power-does-the-pi4b-use-power-measurements

[126] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. London, UK, UK: Springer-Verlag, 2002, pp. 53–65. [Online]. Available: http://dl.acm.org/citation.cfm?id=646334.687801

[127] O. Mazhelis and P. Tyrväinen, "A framework for evaluating internet-of-things platforms: Application provider viewpoint," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 147–152.

[128] R. Merritt, "LoRa Taps New Chips, Smart Homes," 2018, accessed on 16.04.2020. [Online]. Available: https://www.eetimes.com/lora-taps-new-chips-smart-homes/

[129] I. Miers, C. Garman *et al.*, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *2013 IEEE Symposium on Security and Privacy*, May 2013, pp. 397–411.

[130] J. Miles, *R Squared, Adjusted R Squared*. American Cancer Society, 2014. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat06627

[131] A. Miller and J. J. LaViola Jr, "Anonymous byzantine consensus from moderately-hard puzzles: A model for Bitcoin," 2014. [Online]. Available: http://nakamotoinstitute.org/research/anonymous-byzantine-consensus

[132] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of bft protocols," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 31–42.

[133] D. Mingxiao, M. Xiaofeng *et al.*, "A review on consensus algorithm of blockchain," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2017, pp. 2567–2572.

[134] S. Moss, "Microsoft azure suffers outage after cooling issue," https://https://www.datacenterdynamics.com/news/microsoft-azure-suffers-outage-after-cooling-issue/, 2018.

[135] S. Nakamoto. (2008) Bitcoin: A peer-to-peer electronic cash system. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[136] M. Naor, "Bit commitment using pseudorandomness," *Journal of Cryptology*, vol. 4, no. 2, pp. 151–158, Jan 1991. [Online]. Available: https://doi.org/10.1007/BF00196774

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

[138] K. Nikitin, E. Kokoris-Kogias, P. Jovanovic, N. Gailly, L. Gasser, I. Khoffi, J. Cappos, and B. Ford, "CHAINIAC: Proactive software-update transparency via collectively signed skipchains and verified builds," in *26th USENIX Security Symposium (USENIX Security 17).* Vancouver, BC: USENIX Association, Aug. 2017, pp. 1271–1287. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/nikitin

[139] S. R. Niya, S. S. Jha, T. Bocek, and B. Stiller, "Design and implementation of an automated and decentralized pollution monitoring system with blockchains, smart contracts, and lorawan," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–4.

[140] O. Novo, "Blockchain meets iot: An architecture for scalable access management in iot," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, April 2018.

[141] O. Novo, "Blockchain meets iot: an architecture for scalable access management in iot," *IEEE Internet of Things Journal*, 2018.

[142] M. N. Ochoa, A. Guizar, M. Maman, and A. Duda, "Toward a self-deployment of lora networks: Link and topology adaptation," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2019, pp. 1–7.

[143] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "Towards a novel privacy-preserving access control model based on blockchain technology in iot," in *Europe and MENA Cooperation Advances in Information and Communication Technologies*, Á. Rocha, M. Serrhini, and C. Felgueiras, Eds. Cham: Springer International Publishing, 2017, pp. 523–533.

[144] ——, "Towards a novel privacy-preserving access control model based on blockchain technology in iot," in *Europe and MENA Cooperation Advances in Information and Communication Technologies*. Springer, 2017, pp. 523–533.

[145] K. R. Ozyilmaz and A. Yurdakul, "Designing a blockchain-based iot with ethereum, swarm, and lora: The software solution to create high availability with minimal security risks," *IEEE Consumer Electronics Magazine*, vol. 8, no. 2, pp. 28–34, 2019.

[146] R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *Advances in Cryptology – EUROCRYPT 2017*, J.-S. Coron and J. B. Nielsen, Eds. Cham: Springer International Publishing, 2017, pp. 643–673.

[147] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology — CRYPTO '91*, J. Feigenbaum, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 129–140.

[148] K. Pietrzak, "Simple verifiable delay functions," Cryptology ePrint Archive, Report 2018/627, 2018, https://eprint.iacr.org/2018/627.

[149] D. Poluektov, M. Polovov, P. Kharin, M. Stusek, K. Zeman, P. Masek, I. Gudkova, J. Hosek, and K. Samouylov, "On the performance of lorawan in smart city: End-device design and communication coverage," in *Distributed Computer and Communication Networks*, V. M. Vishnevskiy, K. E. Samouylov, and D. V. Kozyrev, Eds. Cham: Springer International Publishing, 2019, pp. 15–29.

[150] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," *White paper*, pp. 1–47, 2017.

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

[152] S. Popov, "The tangle," *cit. on*, p. 131, 2016.

[153] J. Prestwich, "What to Expect When ETH's Expecting," Jan. 2019, accessed on 01.08.2019. [Online]. Available: https://hackernoon.com/what-to-expect-when-eths-expecting-80cb4951afcd

[154] Z. C. Q. Wen, Y. Guo and D. Wu, "A blockchain-based data sharing scheme in the supply chain by iiot," in *Industrial Cyber-Physical Systems - IEEE ICPS 2019*, 2019, pp. 683–688.

[155] M. S. Rahman, N. C. Peeri, N. Shrestha, R. Zaki, U. Haque, and S. H. Ab Hamid, "Defending against the novel coronavirus (covid-19) outbreak:

How can the internet of things (iot) help to save the world?" *Health Policy and Technology*, 2020.

[156] Y. Rahulamathavan, R. C. . Phan *et al.*, "Privacy-preserving blockchain based iot ecosystem using attribute-based encryption," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Dec 2017, pp. 1–6.

[157] P. P. Ray, "A survey of iot cloud platforms," *Future Computing and Informatics Journal*, vol. 1, no. 1, pp. 35 – 46, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2314728816300149

[158] E. Regnath and S. Steinhorst, "Leapchain: Efficient blockchain verification for embedded iot," in *Proceedings of the International Conference on Computer-Aided Design*, ser. ICCAD '18. New York, NY, USA: ACM, 2018, pp. 74:1–74:8. [Online]. Available: http://doi.acm.org/10.1145/3240765.3240820

[159] L. Ren, K. Nayak, I. Abraham, and S. Devadas, "Practical synchronous byzantine consensus," *CoRR*, vol. abs/1704.02397, 2017. [Online]. Available: http://arxiv.org/abs/1704.02397

[160] Z. Ren and Z. Erkin, "Vapor: A value-centric blockchain that is scale-out, decentralized, and flexible by design," in *Financial Cryptography and Data Security*, I. Goldberg and T. Moore, Eds. Cham: Springer International Publishing, 2019, pp. 487–507.

[161] S. Roy, A. K. Das *et al.*, "Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 457–468, Jan 2019.

[162] M. Saelens, J. Hoebeke, A. Shahid, and E. De Poorter, "Impact of eu duty cycle and transmission power limitations for sub-ghz lpwan srds: an overview and future challenges," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 219, 2019.

[163] M. Salimitari, M. Chatterjee, and Y. Fallah, "A survey on consensus methods in blockchain for resource-constrained iot networks," Apr 2020. [Online]. Available: https://www.techrxiv.org/articles/A_Survey_on_Consensus_Methods_in_Blockchain_for_Resource-constrained_IoT_Networks/12152142/1

[164] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, "An architecture for the internet of things with decentralized data and centralized control," in *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, 2015, pp. 1–8.

[165] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 515–532.

[166] S. Sen and M. J. Freedman, "Commensal cuckoo: Secure group partitioning for large-scale services," *SIGOPS Oper. Syst. Rev.*, vol. 46, no. 1, pp. 33–39, Feb. 2012. [Online]. Available: http://doi.acm.org/10.1145/2146382.2146389

[167] P. Singh and K. Deshpande, "Performance evaluation of cryptographic ciphers on iot devices," *CoRR*, vol. abs/1812.02220, 2018. [Online]. Available: http://arxiv.org/abs/1812.02220

[168] R. P. Singh, M. Javaid, A. Haleem, and R. Suman, "Internet of things (iot) applications to fight against covid-19 pandemic," *Diabetes Metabolic Syndrome: Clinical Research Reviews*, vol. 14, no. 4, pp. 521 – 524, 2020.

[Online]. Available:
http://www.sciencedirect.com/science/article/pii/S1871402120301065

[169] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on lpwa technology: Lora and nb-iot," *ICT Express*, vol. 3, no. 1, pp. 14 – 21, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405959517300061

[170] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre: A fast and scalable cryptocurrency protocol." *IACR Cryptology ePrint Archive*, vol. 2016, p. 1159, 2016.

[171] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," in *Financial Cryptography and Data Security*, R. Böhme and T. Okamoto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 507–527.

[172] ——, "Phantom: A scalable blockdag protocol." *IACR Cryptology ePrint Archive*, vol. 2018, p. 104, 2018.

[173] A. Sonnino, S. Bano, M. Al-Bassam, and G. Danezis, "Replay attacks and defenses against cross-shard consensus in sharded distributed ledgers," *CoRR*, vol. abs/1901.11218, 2019. [Online]. Available: http://arxiv.org/abs/1901.11218

[174] J. Sousa and A. Bessani, "From byzantine consensus to bft state machine replication: A latency-optimal transformation," in *2012 Ninth European Dependable Computing Conference*, May 2012, pp. 37–48.

[175] M. Stadler, "Publicly verifiable secret sharing," in *Advances in Cryptology — EUROCRYPT '96*, U. Maurer, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 190–199.

[176] T. Subashri, R. Arunachalam *et al.*, "Pipelining architecture of aes encryption and key generation with search based memory," in *Recent Trends in Network Security and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 224–231.

[177] E. Syta, P. Jovanovic, E. K. Kogias, N. Gailly, L. Gasser, I. Khoffi, M. J. Fischer, and B. Ford, "Scalable bias-resistant distributed randomness," in *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017, pp. 444–460.

[178] E. Syta, I. Tamas, D. Visher, D. I. Wolinsky, P. Jovanovic, L. Gasser, N. Gailly, I. Khoffi, and B. Ford, "Keeping authorities "honest or bust" with decentralized witness cosigning," in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 526–545.

[179] 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, thirdquarter 2016.

[180] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2084–2123, thirdquarter 2016.

[181] N. Van Saberhagen, "Cryptonote v 2.0," 2013.

[182] R. Venkatesan, M. V. Raghavan, and K. S. S. Prakash, "Architectural considerations for a centralized global iot platform," in *2015 IEEE Region 10 Symposium*, 2015, pp. 5–8.

[183] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *International workshop on open problems in network security*. Springer, 2015, pp. 112–125.

[184] J. Wang and H. Wang, "Monoxide: Scale out blockchains with asynchronous consensus zones," in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. Boston, MA: USENIX Association, Feb. 2019, pp. 95–112. [Online]. Available: https://www.usenix.org/conference/nsdi19/presentation/wang-jiaping

[185] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38 437–38 450, 2018.

[186] W. Wang *et al.*, "A Survey on Consensus Mechanisms and Mining Management in Blockchain Networks," *arXiv preprint arXiv:1805.02707*, 2018.

[187] X. Wang, J. Zhang *et al.*, "Performance evaluation of attribute-based encryption: Toward data privacy in the iot," in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 725–730.

[188] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng, "Survey on blockchain for internet of things," *Computer Communications*, vol. 136, pp. 10 – 29, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366418306881

[189] B. Wesolowski, "Efficient verifiable delay functions," Cryptology ePrint Archive, Report 2018/623, 2018, https://eprint.iacr.org/2018/623.

[190] R. Williams, E. McMahon, S. Samtani, M. Patton, and H. Chen, "Identifying vulnerabilities of consumer internet of things (iot) devices: A scalable approach," in *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2017, pp. 179–181.

[191] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[192] P. Wuille, "Bip103: Block size following technological growth," 2015. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0103.mediawiki

[193] C. Xu, K. Wang, P. Li, S. Guo, J. Luo, B. Ye, and M. Guo, "Making big data open in edges: A resource-efficient blockchain-based approach," *IEEE Transactions on Parallel and Distributed Systems*, pp. 1–1, 2018.

[194] ——, "Making big data open in edges: A resource-efficient blockchain-based approach," *IEEE Transactions on Parallel and Distributed Systems*, pp. 1–1, 2018.

[195] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[196] Y. Xu and Y. Huang, "An n/2 byzantine node tolerate blockchain sharding approach," *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, Mar 2020. [Online]. Available: http://dx.doi.org/10.1145/3341105.3374069

[197] D. Yang, J. Gavigan, and Z. Wilcox-O'Hearn, "Survey of confidentiality and privacy preserving technologies for blockchains," *R3, Zcash Company, Res. Rep*, 2016.

[198] E. Yang, "The challenges of implementing LoRa and LoRaWAN in industries worldwide," Oct. 2019, accessed on 15.05.2020. [Online]. Available: https://www.asmag.com/showpost/30700.aspx

[199] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[200] L. Yeh, P. Chiang *et al.*, "Cloud-based fine-grained health information access control framework for lightweightiot devices with dynamic auditing andattribute revocation," *IEEE Transactions on Cloud Computing*, vol. 6, no. 2, pp. 532–544, April 2018.

[201] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: Bft consensus with linearity and responsiveness," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, ser. PODC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 347–356. [Online]. Available: https://doi.org/10.1145/3293611.3331591

[202] ——, "Hotstuff: Bft consensus with linearity and responsiveness," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, ser. PODC '19. New York, NY, USA: ACM, 2019, pp. 347–356. [Online]. Available: http://doi.acm.org/10.1145/3293611.3331591

[203] S. Yin, "Design and implementation of iot centralized management model with linkage policy," *IET Conference Proceedings*, pp. 5 .–5 .(1), January 2015. [Online]. Available: https://digital-library.theiet.org/content/conferences/10.1049/cp.2015.0859

[204] F. Yingwei, "Alibaba cloud reports io hang error in north china," https://equalocean.com/technology/20190303-alibaba-cloud-reports-io-hang-error-in-north-china, 2019.

[205] G. Yu, X. Wang, K. Yu, W. Ni, J. A. Zhang, and R. P. Liu, "Survey: Sharding in blockchains," *IEEE Access*, vol. 8, pp. 14 155–14 181, 2020.

[206] 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*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 931–948. [Online]. Available: http://doi.acm.org/10.1145/3243734.3243853

[207] M. Zeng, Y. Li, K. Zhang, M. Waqas, and D. Jin, "Incentive mechanism design for computation offloading in heterogeneous fog computing: A contract-based approach," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[208] R. Zhang and B. Preneel, "Lay down the common metrics: Evaluating proof-of-work consensus protocols' security," in *2019 IEEE Symposium on Security and Privacy (SP). IEEE*, 2019.

[209] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang, "Blockchain challenges and opportunities: A survey," in *Work Pap.*, 2016.

[210] M. Zichichi, S. Ferretti, and G. D'angelo, "A framework based on distributed ledger technologies for data management and services in intelligent transportation systems," *IEEE Access*, vol. 8, pp. 100 384–100 402, 2020.

[211] R. Zwetsloot, "Raspberry pi 4 specs and benchmarks." [Online]. Available: https://magpi.raspberrypi.org/articles/raspberry-pi-4-specs-benchmarks