Post-Quantum Blockchain for Internet of Things Domain

A thesis submitted in partial fulfilment of the requirements for the degree

of Doctor of Philosophy

by

Bakhtiyor Yokubov

Department of Electronic and Electrical Engineering

College of Engineering, Design and Physical Sciences

Brunel University London

United Kingdom

October 2023

Abstract

In the evolving realm of quantum computing, emerging advancements reveal substantial challenges and threats to existing cryptographic infrastructures, particularly impacting blockchain technologies. These are pivotal for securing the Internet of Things (IoT) ecosystems. The traditional blockchain structures, integral to myriad IoT applications, are susceptible to potential quantum computations, emphasizing an urgent need for innovations in post-quantum blockchain solutions to reinforce security in the expansive domain of IoT.

This PhD thesis delves into the crucial exploration and meticulous examination of the development and implementation of post-quantum blockchain within the IoT landscape, focusing on the incorporation of advanced post-quantum cryptographic algorithms in Hyperledger Fabric, a forefront blockchain platform renowned for its versatility and robustness. The primary aim is to discern viable post-quantum cryptographic solutions capable of fortifying blockchain systems against impending quantum threats enhancing security and reliability in IoT applications.

The research comprehensively evaluates various post-quantum public-key generation and digital signature algorithms, performing detailed analyses of their computational time and memory usage to identify optimal candidates. Furthermore, the thesis proposes an innovative lattice-based digital signature scheme Fast-Fourier Lattice-based Compact Signature over NTRU (Falcon), which leverages the Monte Carlo Markov Chain (MCMC) algorithm as a trapdoor sampler to augment its security attributes.

The research introduces a post-quantum version of the Hyperledger Fabric blockchain that integrates post-quantum signatures. The system utilizes the Open Quantum Safe (OQS) library, rigorously tested against NIST round 3 candidates for optimal performance. The study highlights the capability to manage IoT data securely on the post-quantum Hyperledger Fabric blockchain through the Message Queue Telemetry Transport (MQTT) protocol. Such a configuration ensures safe data transfer from IoT sensors directly to the blockchain nodes, securing the processing and recording of sensor data within the node ledger.

The research addresses the multifaceted challenges of quantum computing advancements and significantly contributes to establishing secure, efficient, and resilient post-quantum blockchain infrastructures tailored explicitly for the IoT domain. These findings are instrumental in elevating the security paradigms of IoT systems against quantum vulnerabilities and catalysing innovations in post-quantum cryptography and blockchain technologies.

Furthermore, this thesis introduces strategies for the optimization of performance and scalability of post-quantum blockchain solutions and explores alternative, energy-efficient consensus mechanisms such as the Raft and Stellar Consensus Protocol (SCP), providing sustainable alternatives to the conventional Proof-of-Work (PoW) approach.

A critical insight emphasized throughout this thesis is the imperative of synergistic collaboration among academia, industry, and regulatory bodies. This collaboration is pivotal to expedite the adoption and standardization of post-quantum blockchain solutions, fostering the development of interoperable and standardized technologies enriched with robust security and privacy frameworks for end users.

In conclusion, this thesis furnishes profound insights and substantial contributions to implementing post-quantum blockchain in the IoT domain. It delineates original contributions to the knowledge and practices in the field, offering practical solutions and advancing the state-of-the-art in post-quantum cryptography and blockchain research, thereby paving the way for a secure and resilient future for interconnected IoT systems.

Declaration

I declare that this PhD thesis, titled “Post-Quantum Blockchain for Internet of Things Domain”, is my original work, except where acknowledged. It has not been submitted for any degree or examination at another university.

Date: April 2023

Place: London, United Kingdom

Bakhtiyor Yokubov

Publications

1. Bakhtiyor Yokubov and Lu Gan "*A Performance Comparison of Post-Quantum Algorithms in Blockchain*." The Journal of The British Blockchain Association, September 2022.

2. Bakhtiyor Yokubov and Lu Gan, "*Comprehensive Comparison of Post-Quantum Digital Signature schemes in Blockchain*", 2021 IEEE International Conference on Electronic Communications, Internet of Things and Big Data, 10-12 December 2021.

3. Bakhtiyor Yokubov, Lu Gan and Cong Ling, "*Blockchain-Based System for IoT Devices Using Post-Quantum Cryptography*", 6th IMA Conference on Mathematics in Defence and Security, 30-31 March 2021.

Acknowledgements

I want to express my deepest gratitude and appreciation to my principal supervisor, Dr. Lu Gan, for her invaluable guidance, support, and encouragement throughout my PhD journey. Her expertise, constructive feedback, and unwavering commitment have been instrumental in completing this thesis. I am honoured to have had the opportunity to learn from her and work under her supervision.

I acknowledge the "El-Yurt Umidi" foundation's generous sponsorship and support. Their financial assistance enabled me to pursue my academic aspirations and research.

I am grateful to Prof. Cong Ling, his group, and students on post-quantum cryptography at Imperial College London for their valuable input, collaboration, and sharing of expertise, which has significantly enriched my understanding of the subject matter and has greatly benefited my research.

I would also like to thank my supervisor, Prof. Tatiana Kalganova, for her valuable insights, guidance, and support throughout my research journey. Her vast knowledge and experience have contributed significantly to the development of this thesis.

My heartfelt appreciation goes to my family for their unwavering love, support, and encouragement throughout my studies. Their belief in my abilities and their sacrifices have played a vital role in helping me persevere through challenging times. I dedicate this thesis to them as a testament to their unconditional love and support.

Furthermore, I would like to thank my friends and colleagues, who have provided me with valuable insights, camaraderie, and assistance during my academic journey. Their encouragement and support have been truly appreciated.

Finally, I am grateful to Brunel University London for providing me with the resources and facilities necessary to complete this research. Their support has allowed me to pursue my academic endeavours and contribute to the field of post-quantum blockchain in the IoT domain.

List of Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
B2B	Business to Business
BCSP	Blockchain Cryptographic Service Provider
BFT	Byzantine fault tolerant
CA	Certificate Authority
CLI	Command-Line Interface
DAC	Distributed Autonomous Corporations
DApps	Decentralized Applications
DBFT	Delegated Byzantine Fault Tolerance
DeFi	Decentralized Finance
DLT	Distributed Ledger Technologies
DPoS	Delegated Proof of Stake
DSA	Digital Signature Algorithm
DSR	Design Science Research
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
ECDH	Elliptic-Curve Diffie–Hellman
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm

EOV	Execute-Order-Validate
EU	European Union
e-voting	Electronic Voting
Falcon	Fast-Fourier Lattice-based Compact Signature over NTRU
FBA	Federated Byzantine Agreement
GDPR	General Data Protection Regulation
GPV	Gentry, Peikert, and Vaikuntanathan
GS	Gram-Schmidt
HFE	Hidden Field Equation-based
HIPAA	Health Insurance Portability and Accountability Act of 1996
HLF	Hyperledger Fabric
IF	Integer Factoring
IIoT	Industrial Internet of Things
IMHK	Independent Metropolis-Hastings-Klein
IoT	Internet of Things
IP	Internet Protocol
ISIS	Inhomogeneous Short Integer Solution
MCMC	Markov Chain Monte Carlo
MQTT	Message Queue Telemetry Transport
MSP	Membership Service Provider

MVCC	Multi-Version Concurrency Control
NFC	Near-Field Communication
NIST	National Institute of Standards and Technology
NP	Non-deterministic Polynomial-time
OQS	Open Quantum Safe
PBFT	Practical Byzantine Fault Tolerance
PHI	Protected Health Information
PoA	Proof of Authority
PoET	Proof of Elapsed Time
PoS	Proof of Stake
PoW	Proof of Work
RFID	Radio Frequency Identification Device
RSA	Rivest-Shamir-Adleman
SCP	Steller Consensus Protocol
Seth	Sawtooth Ethereum
SGX	Software Guard Extensions
SHA	Secure Hash Algorithm
SIS	Short Integer Solution
SMK	Symmetric Metropolis-Klein
SVP	Short Vector Problem

TPS	Transaction per Second
US	United States
WHO	World Health Organization
XMSS	Extended Merkle Signature Scheme
YAC	Yet Another Consensus

Chapter 1 Introduction

1.1. Research Background

Blockchain technology, originating from Bitcoin cryptocurrency [1], has been recognized for its secure connections, data protection, resiliency, and openness [2]. It preserves data within a sequence of interconnected blocks through cryptographic hashes, effectively addressing concerns such as double-spending [3],[4],[5]. The versatility and robustness of blockchain technology have led to its adoption across various domains, including smart factories [6],[7], smart health [8], logistics [9],[10], measurement systems [11], electronic voting [12], and IoT [13].

Blockchain's inherent strength lies in its ability to conduct secure and immutable transactions, eliminating the need for a trusted intermediary. This autonomy is achieved through advanced cryptographic protocols, enabling secure node communications and utilising digital signatures and cryptographic hash functions for peer-to-peer consensus.

IoT, another revolutionary technology, enriches everyday life by enabling connectivity between numerous devices equipped with sensors and actuators, serving diverse applications [14] ranging from constrained environments like homes to larger ecosystems like campuses and cities. Due to its immutable and secure nature, blockchain is ideally suited to enhance confidentiality, integrity, and availability of information within IoT frameworks [15].

Recent advancements in quantum computing have seen industry leaders such as Google, Microsoft, and IBM make significant strides in developing quantum computers. While achieving stable qubits capable of surpassing traditional public-key cryptography continues, developments like IBM's 127-qubit processor indicate significant progress [16]. Quantum computers, renowned for their computational efficacy, have the potential to address previously intractable problems by reducing time complexity [17], thus promising revolutionary transformations across various sectors.

However, the heightened computational capabilities of quantum computers pose significant risks, particularly to encryption protocols like public-key cryptography, which traditionally rely on the computational assumptions of classical computers [18]. Shor's algorithm, a quantum computing algorithm, can efficiently solve the Elliptic Curve (EC) and the Integer Factoring (IF) problems [19], threatening the security of Rivest-Shamir-Adleman (RSA) and Elliptic Curve Digital Signature Algorithm (ECDSA).

The accelerated advancement of quantum computing introduces substantial risks to current cryptographic systems, including blockchain. In the context of IoT, where secure communication is paramount due to the extensive exchange of information between interconnected devices, the potential impact of quantum attacks is particularly alarming. Integrating post-quantum cryptography in blockchain is imperative to uphold the security and resilience of IoT networks in the long term. This research endeavours to explore the vulnerabilities quantum computing introduces and proposes pragmatic solutions to shield blockchain systems within the IoT domain from prospective quantum threats, thereby fortifying the overall security framework of these systems.

1.2. Research Motivation

The emergence of quantum computing delineates profound challenges to the security infrastructures of prevailing cryptographic systems, including those within blockchain technology. It becomes even more consequential in IoT applications, where myriad interconnected devices that exchange information and rely on secure communication channels operate with specific constraints and finite resources, affecting the formulation and deployment of post-quantum cryptographic resolutions.

The impetus for this research is primarily driven by the looming threats quantum computing inflicts upon current cryptographic infrastructures. The consequences of quantum infiltrations may not be instantaneous, but the accelerated advancement of quantum computers accentuates the imperative to fortify cryptographic systems against future vulnerabilities. It holds relevance for blockchain technology, a foundational component for many IoT applications, where the integrity and robustness of the system are intrinsically linked to cryptographic security.

Moreover, existing literature lacks an exhaustive comparison and critical appraisal of post-quantum algorithms within blockchain structures. Much of the existing research centres around individual post-quantum algorithms, yielding a knowledge deficit concerning the optimal algorithms tailored for distinct blockchain applications and IoT environments. Undertaking a thorough evaluation of post-quantum algorithms is crucial, considering their operational efficiency, resource constraints, and compatibility with blockchain infrastructures and IoT deployments.

Additionally, a conspicuous void exists in the tangible, real-world applications of post-quantum blockchain frameworks in the IoT sphere. Validating the practicality and efficacy of these systems in tangible scenarios is paramount for fostering their assimilation and mitigating the security risks emanating from quantum computing advancements.

This research aspires to insulate blockchain technology from quantum threats, pinpoint apt post-quantum algorithms for incorporation into blockchain structures and scrutinize the critical considerations in deploying these algorithms within the IoT realm. This endeavour seeks to augment the existing corpus of knowledge by introducing a fortified security methodology for post-quantum signature algorithms.

By navigating through these challenges and filling the existing knowledge gaps, this study stands to propel the development of post-quantum cryptography in blockchain and IoT frameworks, thereby enhancing the overall security posture and resilience of these systems against impending quantum threats.

1.3. Research Questions

The primary objective of this PhD thesis is to architect a blockchain system fortified with post-quantum cryptography tailored explicitly for the IoT landscape. To achieve this, the research delves into the following research questions:

- How can blockchain technology be reinforced to counteract prospective quantum threats effectively?
- Given the vast array of available post-quantum algorithms, which are optimal for integration into blockchain architectures without compromising functionality and integrity?

- What are the principal considerations and intrinsic challenges when embedding post-quantum cryptographic algorithms into blockchain structures, particularly those developed for IoT applications?
- How can innovative approaches be utilized to amplify the robustness of post-quantum signature algorithms further?

1.4. Research Methodology

This study employs the Design Science Research (DSR) methodology as its foundational investigative approach. DSR is renowned for enabling the development of practical solutions to real-world problems through iterative cycles of design and evaluation while concurrently distilling design principles and guidelines with broad applicability beyond the immediate context of research. The selection of DSR methodology in this research is premised on the following grounds:

- The core aim of utilizing this strategy is to develop an artifact that addresses identified research challenges, subsequently contributing to the enrichment of collective knowledge [20]. The artifact resulting from the DSR approach can take various forms, such as design principles, constructs, methodologies, design theories, models, technology norms, or other types of knowledge that contribute to resolving research challenges [21]. In this context, the artifacts manifest as comprehensive guiding frameworks for developing and implementing post-quantum blockchain in the IoT domain, offering actionable insights and resolutions to research problems.
- The methodology offers a clear and structured pathway, elucidating distinct roles, methods, and artifacts, ensuring the derivation of substantial and credible outcomes [22]. It enables the application of analytical and synthetic perspectives and methodologies to comprehensively represent the study's breadth.
- DSR is adept at addressing challenges at any level within the information system spectrum researchers encounter.

This research follows the DSR process model propounded by Peffers et al. [20], consisting of the following six activities:

1. **Problem identification:** This activity is foundational, outlining the research problems. Here, challenges within current blockchain and IoT technologies were discerned, and

strategies for mitigating challenges of post-quantum blockchain within the IoT domain were conceptualized.

2. **Objectives Definition:** This stage involves pinpointing plausible solutions to the discerned problems. Here, proposals for an architecture amalgamating post-quantum cryptography and blockchain were developed, addressing the security constraints of current blockchain applications. A meticulous comparison of various post-quantum cryptography signature schemes was also undertaken.
3. **Design and development:** The conceptualized artifact was materialized and integrated with this stage. A novel architectural embodiment of post-quantum blockchain for IoT applications was designed, juxtaposing the prevailing technologies.
4. **Demonstration:** This phase is pivotal for substantiating the efficacy of the proposed solution in addressing the research problem. A deployment of post-quantum blockchain in IoT was undertaken to validate the conceptual proof and the artifact's applicability.
5. **Evolution:** The developed artifact was critically assessed for its contribution to resolving the research problem. The analysis of proof-of-concept scenarios was conducted using external software tools, and the subsequent findings were documented.
6. **Communication:** This final phase involved the dissemination of the design and the efficacy of the artifact to the research community. This study manifested in various scholarly publications and presentations at international scientific symposiums, fostering knowledge sharing within the academic fraternity.

1.5. Research Contributions

This thesis provides the following significant contributions to the field of post-quantum cryptography and blockchain technology in the context of the IoT:

1. An approach has been proposed for assessing post-quantum algorithms' efficiency and operational attributes within blockchain environments. This approach is pivotal for understanding these algorithms' computational and memory demands, aiming to fill the existing knowledge gap by promoting a more comprehensive and multi-faceted evaluation. This contribution is crucial as it empowers practitioners to identify the most compatible post-quantum algorithms for distinct blockchain applications, enabling a meticulous examination of current post-quantum blockchain solutions and enhancing preparedness for impending quantum computing advancements.

2. A framework has been developed to establish a quantum-resistant Hyperledger Fabric system specifically designed for the IoT. This framework addresses the vulnerabilities emanating from the convergence of IoT, blockchain, and prospective quantum risks, placing a premium on crypto-agility to facilitate seamless transitions to quantum-safe environments. This framework allows nodes to choose from various post-quantum signature algorithms, reflecting the diverse needs of IoT infrastructures. Additionally, the framework incorporates an algorithm leveraging the MQTT protocol to safeguard the integrity of IoT data during transmission. It is seminal in the intersection of blockchain and IoT, presenting a novel strategy to mitigate the risks posed by the evolution of quantum computing.
3. A method to reinforce the Falcon post-quantum signature scheme has been introduced, integrating Monte Carlo Markov Chain (MCMC) sampling. This method addresses the inherent variance in Falcon's discrete Gaussian trapdoor sampler by synergizing Independent Metropolis-Hastings-Klein (IMHK) and Symmetric Metropolis-Klein (SMK) algorithms with Falcon's existing process, thus diminishing the standard deviation. The integration of MCMC sampling with cryptographic methods in this contribution is groundbreaking, paving the way for subsequent research on bolstering cryptographic resilience against quantum assaults.

Together, these contributions underscore the innovative and profound nature of the research conducted, highlighting the advanced methodologies, strategies, and frameworks developed to navigate the complex challenges in post-quantum cryptography and blockchain within the IoT domain. The contributions delineated herein possess intrinsic merit and hold substantial implications for future research trajectories and practical applications in related domains.

1.6. Thesis Structure

This thesis is organized as follows:

Chapter 1: This initial chapter sets the stage, introducing the central research problem and emphasizing the urgency and relevance of the study. It outlines the research objectives and poses the crucial questions driving this investigation. The research methodology adopted is detailed, illustrating the robust approach to derive reliable results. This chapter also highlights

the significant contributions of this study and concludes by providing an overview of the thesis structure.

Chapter 2: This chapter comprehensively reviews the foundational principles of blockchain technology, post-quantum cryptography, and the IoT. It explores the diverse applications of blockchain across various industries and examines the compatibility of different blockchain platforms with IoT deployments.

Chapter 3: This chapter presents a detailed analysis of various post-quantum cryptographic algorithms. It evaluates the performance of different post-quantum public-key generation and digital signature protocols in blockchain contexts, using computation time and memory consumption as critical metrics. The chapter also features case studies illuminating existing post-quantum blockchain technology implementations.

Chapter 4: This chapter presents a novel blueprint for constructing a post-quantum Hyperledger Fabric blockchain system. It provides in-depth discussions on the design components and their implementations. The efficacy of the post-quantum Hyperledger Fabric is then demonstrated in real-world IoT settings, focusing on temperature and humidity sensors.

Chapter 5: This chapter explores the complexities of the Falcon post-quantum signature scheme and proposes an innovative integration of the MCMC algorithm into Falcon's trapdoor sampling process. This integration is described in detail, highlighting its implications and significance in post-quantum cryptography.

Chapter 6: The concluding chapter synthesizes the essential findings and insights obtained from the research. Based on the study results, it provides thoughtful recommendations for future work in this field. Finally, the chapter discusses the limitations and scope of the research, offering a balanced and transparent closure to the thesis.

1.7. Summary

This introductory chapter provides a comprehensive overview of the research problem, emphasizing the significance of the study in the context of quantum computing challenges to the cryptographic systems underpinning blockchain technology, particularly within the IoT framework. The research's importance is heightened by the potential vulnerabilities that

quantum computing presents to existing cryptographic systems and the evident gaps in contemporary literature.

The research trajectory has been meticulously outlined, starting with its motivation, leading to the pertinent questions it aims to address, and elucidating the methodological approach that steers this study. The research underscores the risks associated with quantum computing and highlights the need for holistic post-quantum solutions tailored for blockchain in the IoT domain.

Moreover, this chapter highlights the primary contributions that emerge from this study, giving readers an insight into the innovative solutions and methods that will be explored in subsequent chapters. An outlined thesis structure then offers a clear roadmap, ensuring clarity and logic as readers delve deeper into the research.

This chapter lays the foundational groundwork, setting the stage for the following detailed explorations and discussions. The primary goal is to bridge the identified knowledge gaps and push the field of post-quantum cryptography in blockchain technology, especially within the IoT domain, towards greater security and resilience. The subsequent chapters take on this endeavour, diving into the intricacies of the problem while presenting innovative solutions.

Chapter 2 Literature Review

2.1. Introduction

With the advancement and rapid assimilation of technologies like blockchain and IoT, a notable paradigm shift is evident across varied industries, paving the way for groundbreaking solutions to intricate problems. Blockchain technology has emerged as a revolutionary tool because it can reshape diverse sectors, including finance, healthcare, supply chain, and IoT. The fusion of blockchain with IoT heralds enhanced security, privacy, and scalability within interconnected ecosystems. Nevertheless, the advent of quantum computing introduces newfound security conundrums to classical cryptographic methods prevalent in blockchain protocols.

This chapter extensively surveys the literature surrounding blockchain technology, IoT, and their convergence. It seeks to elucidate the core principles, distinctive attributes, and inherent challenges attributed to blockchain and IoT. Furthermore, this chapter scrutinizes various blockchain frameworks suited to IoT deployments and explores the limitations of traditional cryptographic methodologies within these realms, highlighting the susceptibility of existing cryptographic elements to quantum onslaughts. Subsequently, it investigates post-quantum cryptography as a feasible countermeasure to the inadequacies of classical cryptographic systems, examining diverse post-quantum signature protocols ensuring fortified defence against quantum adversaries.

The insights from this literary exploration will lay the bedrock for the forthcoming chapters, emphasizing the conceptualization and assessment of a quantum-proof blockchain infrastructure tailored for IoT solutions. The overarching aim of this chapter is to pinpoint pivotal domains of innovation and scholarly inquiry that are crucial for the anticipated security, scalability, and compatibility of integrated blockchain and IoT platforms through an in-depth analysis of contemporary scholarly works and technological developments.

2.2. Blockchain Technology

2.2.1. Overview of Blockchain

In recent years, academia and industry have observed significant interest in cryptocurrency. Bitcoin, the world's first cryptocurrency, has gained extraordinary success since its launch in 2009. By 2016, its market capitalization surpassed $10 billion, solidifying its position as the most valuable and widely used cryptocurrency [23]. Blockchain technology, which underpins Bitcoin's decentralized architecture, was first proposed in 2008 and actualized in 2009 [1].

Blockchain is a distributed ledger that stores a record of all committed transactions made within the network, organized in a chronological chain of blocks [24]. The chain continuously expands with the addition of new blocks. Several essential features characterize blockchain technology, including decentralization, anonymity, persistence, and auditability, facilitated by integrating core technologies like cryptographic hash, digital signatures, and distributed consensus mechanisms, such as PoW and PoS. Blockchain's decentralized environment significantly reduces costs, enhances efficiency, and builds participant trust.

Although most known for its use in Bitcoin and other cryptocurrencies, blockchain technology has numerous applications beyond digital assets and online payments. In the financial services sector, blockchain can facilitate secure and efficient transactions without the need for intermediaries or traditional banks [25]. Furthermore, the technology holds great promise for revolutionizing a wide range of industries, including healthcare [8], where it can improve data security and patient privacy; logistics [9],[10] by streamlining supply chain management and tracking goods; e-voting [12], enhancing the transparency and integrity of electoral processes; and smart factories [6],[7], optimizing production and resource management, among others. With its potential to enable secure, transparent and decentralized interactions, blockchain is increasingly recognized as a crucial component of the next generation of internet-based systems, including IoT [13]. The following sections provide a deeper understanding of blockchain technology's core components and principles.

2.2.2. Blockchain Architecture

The blockchain is a decentralized public ledger that maintains a comprehensive record of transactions. It comprises a series of blocks connected in a chain-like structure. Each block, excluding the initial block called the genesis block, refers to its immediate predecessor through an inverse reference, essentially the parent block's hash value (see Figure 2.1). The structure of a block contains several vital pieces of information, which together ensure the integrity, immutability, and chronological order of the transactions:

- **Block version.** Within a blockchain, the block version is an attribute in the block header that denotes the version of the protocol rules utilized to create the block. This attribute is an integer value that increments whenever the protocol rules change. By referring to the block version, nodes in the network can identify the appropriate set of validation rules to apply to the block.
- **Parent block hash.** The parent block hash is a header attribute of a block in a blockchain, containing a unique 256-bit hash value that identifies the block's immediate predecessor. Generated using a cryptographic hashing algorithm, this hash value establishes an unbroken chain of blocks in the blockchain. Nodes in the network reference the parent block hash to verify a block's inclusion in the blockchain and confirm its correct chronological placement.
- **Merkle tree root.** The Merkle tree root is a header attribute in a blockchain block that represents the root of a Merkle tree. This tree is created by hashing all the transactions in the block, forming pairs of hashes, and hashing them again until a single hash, the Merkle root, is obtained. As a unique identifier for all the transactions within the block, the Merkle root verifies that the transactions have not been tampered with. Nodes can quickly validate a block's contents by comparing its Merkle root with those of other nodes in the network, circumventing the need to download and verify each transaction individually.
- **Timestamps.** Timestamps, a header attribute of a block in a blockchain, indicate the block's creation time. The timestamp is the number of seconds elapsed since January 1st, 1970, at 00:00 UTC. This information helps determine the order of blocks in the blockchain and ensures that new blocks are not generated too quickly or too slowly. Nodes in the network use the timestamp information to verify that blocks are created within a specific time range and to synchronize their clocks with the rest of the network.

- **Nonce.** The nonce, a header attribute in a block of a blockchain, is a 32-bit arbitrary number used during the mining process of a new block. Miners must find a nonce that, combined with the other information in the block header and hashed, produces a hash value that meets the network's difficulty target. Miners generally start with a nonce of zero and increment it for each hash attempt until they find a valid hash. The nonce helps create a unique hash value for each block and prevents duplicate blocks from being added to the chain.

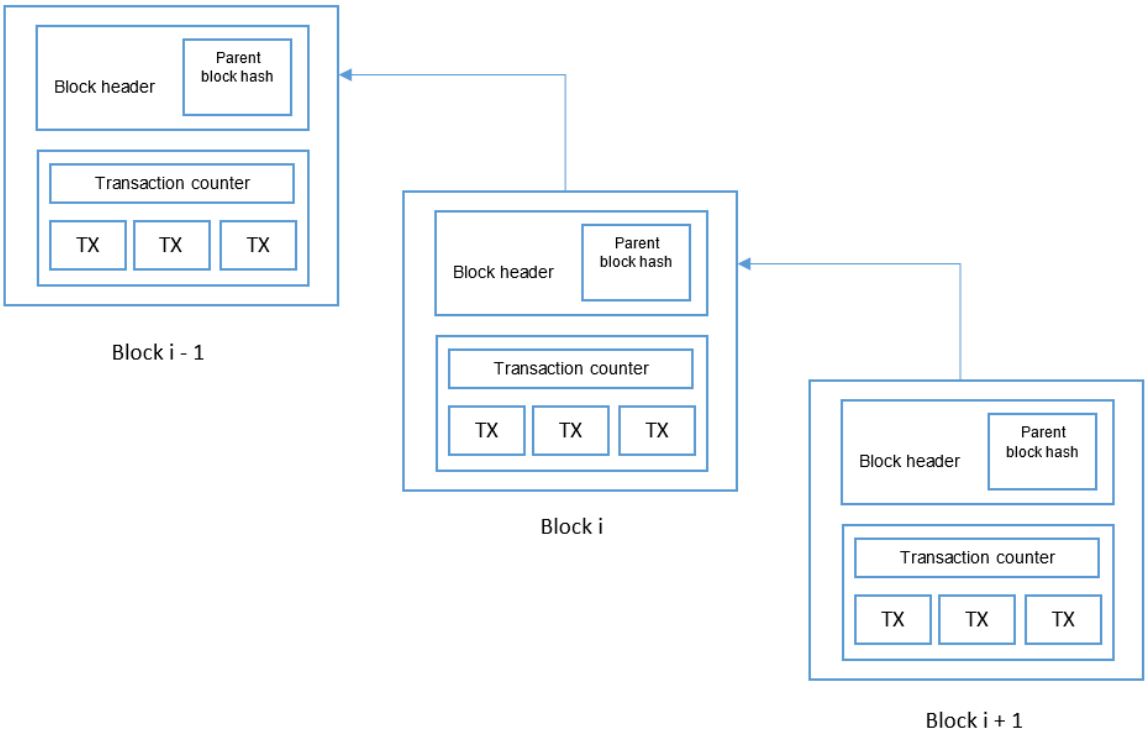

Figure 2.1: Block structure in blockchain.

As new transactions are processed, the blockchain continuously expands. When a new block is created, all nodes in the network collaborate to validate it. Once validated, the block is added to the end of the blockchain with a reference pointing back to its parent block. This process simplifies the detection of unauthorized alterations to previous blocks, as the hash value of a tampered block will significantly differ from that of an unaltered one. Moreover, the blockchain's distributed nature across the network enables rapid identification of data manipulation attempts by other participating nodes. This feature enhances the overall security and integrity of the system.

2.2.3. Digital Signature

Blockchain predominantly comprises transaction records, requiring verification for each newly added record. Blockchain transaction authentication is achieved through a digital signature scheme grounded in asymmetric cryptography. This scheme unfolds in two key phases:

1. The signing phase.
2. The verification phase.

The signing process commences with the generation of a private and a corresponding public key. The private key, produced randomly utilizing an appropriate distribution, is kept secret from other users. With this private key in hand, a corresponding public key is generated. The verification process employs both the public key and the transaction details. While the public key is accessible to all users, the private key remains confidential. In asymmetric cryptography, every user possesses a pair of keys: a private and a public key, which are collectively referred to as a key pair. The following example will illuminate these keys' roles in the digital signature scheme.

Suppose Alice wishes to send a message to Bob. During the signing phase, Alice encrypts (or "signs") the original message using her private key before sending it to Bob. Upon receipt of this encrypted message, Bob utilizes Alice's public key to decrypt it. Successful decryption confirms the message's origin (Alice) and guarantees that the original content remains intact; this concludes the verification phase [26].

2.2.4. Working Flow of Blockchain

A blockchain-based system conducts transactions within a decentralized, trustless network, accommodating various participants. To illustrate the operational process of blockchain, consider a hypothetical scenario involving two parties, Alice and Bob:

1. Alice transfers a specific amount of cryptocurrency (e.g., Bitcoin) to Bob.
2. Alice initiates the transaction by indicating Bob's public address as the recipient and specifying the amount she wishes to transfer. She then signs the transaction using her private key.
3. The transaction is broadcast to the network, reaching an assembly of nodes known as miners or validators.

4. Miners validate the transaction by confirming the correctness of Alice's signature (using her public key) and ensuring that she has sufficient funds to complete the transaction. If the transaction is valid, it is included in a new block within each miner's local blockchain copy.
5. To append the new block to the blockchain, miners compete to solve a cryptographic puzzle (PoW) requiring substantial computational resources. The first miner to solve the puzzle is given the privilege of adding the block to the chain and receives a reward in the form of newly minted cryptocurrency.
6. The new block is disseminated throughout the network, prompting all other nodes to update their local blockchain copies accordingly.
7. Bob can now verify that he has received the cryptocurrency from Alice. He can access and utilize the received funds for future transactions using his private key.

This example illustrates how blockchain technology facilitates a secure, transparent, and tamper-resistant transaction between Alice and Bob. Doing so eliminates the need for a central authority (like a financial institution) to mediate or authenticate the transaction.

2.2.5. Key Characteristics of Blockchain

To provide a comprehensive understanding of blockchain technologies, the following main attributes are highlighted:

1) *Decentralization:* Traditional centralized transaction management systems rely on a trusted third-party entity, such as a bank or government agency, to validate and process transactions. Such centralization often leads to increased costs, performance bottlenecks, and a single point of failure. However, blockchain technology enables a decentralized approach, allowing peer-to-peer validation of transactions without needing a centralized authority for authentication or intervention. This results in reduced costs, improved performance, and a diminished risk of a single point of failure.
2) *Immutability:* Each block is linked to its predecessor via a hash reference in a blockchain. Any modifications to a previous block would invalidate all succeeding blocks. Moreover, the Merkle tree's root hash encompasses the hashes of all committed transactions. Any slight alteration to a transaction would generate a new root hash,

making attempts to tamper with data easily detectable. The blend of hash references and the Merkle tree structure ensures data integrity.

3) *Nonrepudiation:* Using a private key to sign a transaction enables others to verify its authenticity using the corresponding public key. This cryptographic signature ensures that the transaction initiator cannot deny initiating the transaction, providing nonrepudiation.
4) *Transparency:* Public blockchain systems, like Bitcoin and Ethereum, provide all network users equal access and interaction opportunities. Validated transactions are stored in the blockchain and become accessible to every user, rendering the data within a blockchain highly transparent.
5) *Pseudonymity:* Although blockchain data is transparent, user privacy can be maintained to a certain extent by anonymizing blockchain addresses. As described in [3], some blockchain applications seek to protect personal data privacy. However, complete privacy is impossible in blockchain, as addresses can still be traced through inference [27]. Research shows that analysing blockchain data can help identify fraudulent and illegal transactions, suggesting blockchain offers pseudonymity rather than absolute privacy.
6) *Traceability:* Each transaction within a blockchain includes a timestamp, enabling users to trace and verify previous data items' origins within the blockchain, ensuring an accessible and detailed transaction history.

Incorporating these key characteristics, blockchain technology offers a secure and robust foundation for various applications, particularly in decentralized and transparent transaction systems.

2.2.6. History of Blockchain

Casino et al. [28] divided the evolution of blockchain technology into three distinct stages in their 2019 publication: Blockchain 1.0, Blockchain 1.0, related to cryptocurrency transactions initiated with Bitcoin; Blockchain 2.0, encompassing the broader application of blockchain technology across various sectors such as healthcare, governance, and smart cities. A fourth generation of blockchain technology, introduced in 2020 [29], supports all decentralized Blockchain 3.0 applications within the framework of Industry 4.0. These generations can

coexist and interact as blockchain technology continuously evolves rapidly. The following sections delve into the specific details of these different stages.

2.2.6.1. Blockchain 1.0

Although vital technical components of blockchain, such as public key infrastructure, smart contracts, and Byzantine fault tolerance, have existed since the 1980s [30]. They were not incorporated into a single architecture until Satoshi Nakamoto introduced Bitcoin in 2008 [1]. Nakamoto's invention of the "chain of blocks" ledger allowed direct peer-to-peer transactions without central financial institutions or intermediaries. Initially intended to create a new digital currency and payment system, controlling transactions presented a significant challenge. Blockchain technology has since evolved and been improved through various investigations and enhancements.

2.2.6.2. Blockchain 2.0

Between 2010 and 2013, blockchain technology was widely adopted for digital payments, currency transfers, and cryptocurrency applications. During this period, the second generation of blockchain emerged with the unveiling of the Ethereum platform [31]. This platform introduced the concept of a digital ledger, also known as a "smart contract". A smart contract is a computer code or software that acts as a self-executing agreement between parties in a decentralized network. It automatically executes upon meeting predetermined conditions and enforces the agreement's terms without intermediaries. The smart contract encapsulates all the relevant terms and conditions for financial services and applications, ensuring transparency for all parties involved. Nevertheless, the scalability and robustness of this blockchain generation pose significant challenges that demand further attention.

2.2.6.3. Blockchain 3.0

The impact of blockchain technology now reaches beyond the financial services industry into business sectors [32], healthcare [33], and security [34]. Smart contracts have advanced, paving the way for blockchain technology to evolve into a decentralized internet by integrating open standard platforms, data storage, communication networks, and smart contracts. Platforms like the Hyperledger framework [35] have been designed to support multiple decentralized applications (DApps). These DApps operate with blockchain networks running in the

background, while their frontend user interfaces can be developed using any programming language capable of accessing the backend blockchain.

2.2.6.4. Blockchain 4.0

Decentralized applications necessitate multiple architectures and services to communicate within a unified platform. As these systems work in unison, users from different platforms can interact cross-chain. Industry 4.0 requires seamless platform integration with enhanced privacy and trust while addressing scalability issues [29]. Blockchain 4.0 caters to Industry 4.0 by integrating business processes across chains and ensuring security. This iteration of blockchain technology merges Blockchain 3.0, distributed databases, and a public ledger, facilitating real-time business and fulfilment of Industry 4.0 logistics. The SEELE platform of this generation [36] connects independent blockchain systems, achieving linear scalability through a neural consensus methodology and a combination of on-chain and off-chain computations.

2.2.7. Blockchain Applications

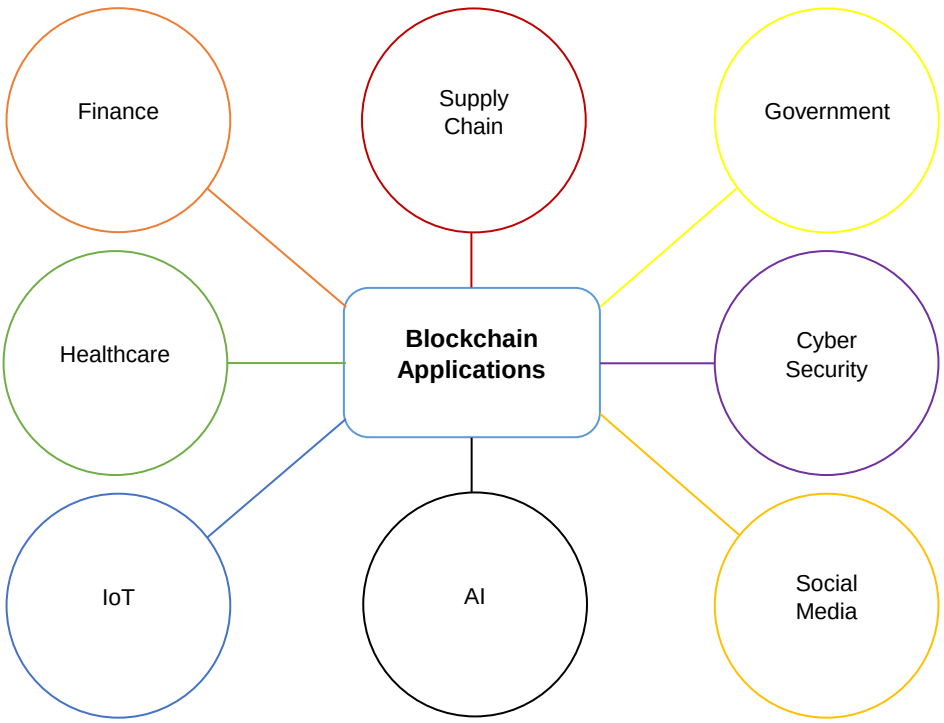

Figure 2.2: Blockchain application domains.

The potential applications of blockchain technology extend across various domains, including finance, government, IoT, healthcare, Artificial Intelligence (AI), supply chain, and more. This section explores numerous blockchain application cases scholars propose worldwide, as illustrated in Figure 2.2.

2.2.7.1. Finance

The global financial system, which processes trillions of dollars and serves billions of people daily, faces several challenges, including cost escalations through fees and delays, friction due to paperwork, and opportunities for fraudulent activities. Blockchain technology has the potential to streamline business operations in the banking and financial service sectors while ensuring a high level of security and reliable records of agreements and transactions. Initially developed to support cryptocurrencies like Bitcoin, blockchain functions as a colossal, distributed ledger capable of securely recording and verifying valuable information across millions of devices globally. Blockchain technology enables secure peer-to-peer storage of money and various assets, eliminating the need for intermediaries such as banks, governments, and financial institutions.

2.2.7.2. Healthcare

In the healthcare industry, blockchain technology has numerous applications, including the traceability of medicine and patients' medical data records. Medicine counterfeiting is a significant issue in the pharmaceutical industry. According to the World Health Organization (WHO), counterfeit or substandard medicines account for approximately 50% of the global market, with 25% consumed in developed or developing countries [37]. Instead of treating diseases, these medicines can cause severe problems for patients. Blockchain technology can address this challenge by ensuring that all transactions are immutable and timestamped, allowing medicine to be tracked and making information tamper-proof.

Maintaining patient data integrity is a primary concern in healthcare. Individualized treatment strategies are necessary for patients with common diseases due to their physical variability, requiring access to their complete medical history. However, medical data is sensitive and requires a secure sharing platform. The current medical record-keeping system lacks both privacy and interoperability. Ensuring the safety and security of patients' medical data is a crucial blockchain application. Blockchain can create a secure and robust framework for

storing patients' medical data, ultimately improving service quality and reducing treatment costs.

Regulatory compliance requirements for healthcare blockchains vary depending on the nature of the sensitive data stored on the blockchain, data usage agreements, and the physical locations of the blockchain nodes and decentralized ledgers holding the information. For example, if blockchain stores Protected Health Information (PHI) about United States (US) citizens, it must adhere to the Health Insurance Portability and Accountability Act of 1996 (HIPAA) [38] regulations. Conversely, if the blockchain contains sensitive information about patients living in the European Union (EU), it must comply with the General Data Protection Regulation (GDPR) [39].

2.2.7.3. Internet of Things

IoT is a transformative technology that connects the physical world to a vast information system, enabling various applications in logistics, food industry, manufacturing, and beyond. The primary goal of IoT is to increase performance and efficiency, reduce machine downtime, and enhance product quality. However, IoT systems face several challenges: heterogeneity, poor interoperability, resource constraints, and security and privacy vulnerabilities. The distributed architecture of IoT presents a significant challenge, as each node in an IoT network is a potential point of failure, susceptible to cyber-attacks like distributed denial-of-service [40]. Additionally, centralized communication of IoT devices may lead to a single point of failure. Data confidentiality, integrity, and authentication are crucial in IoT environments [41].

To overcome these drawbacks, researchers have suggested using blockchain technology to tackle the challenges of IoT systems. Blockchain-based IoT offers numerous advantages:

- **Interoperability:** Blockchain enables connectivity among IoT devices across various systems, promoting seamless communication and data exchange.
- **Data Reliability:** Blockchain incorporates public key cryptography and digital signatures, ensuring the accuracy and authenticity of IoT data.
- **Traceability:** With the help of historical timestamps for each block, blockchain enables tracking and verifying all information, ensuring IoT data traceability.
- **Decentralized Interactions:** IoT systems and devices can interact without intermediaries by utilizing smart contracts on the blockchain. Smart contracts, coded in

high-level programming languages, execute automatically when predetermined conditions are met.

Blockchain technology is an ideal complement to IoT, enhancing scalability, reliability, security, privacy, and interoperability. Section 2.3.3 further delves into blockchain IoT integration.

2.2.7.4. Supply Chain

A supply chain is a network that connects a business and its vendors to produce and distribute goods to buyers. Numerous companies can benefit from incorporating blockchain technology into their supply chains to store, monitor, and optimize immutable and reliable data. Implementing blockchain technology can create secure and transparent supply chains and eliminate counterfeit products by storing vital product information, such as serial numbers, price, location, date, and quality, on a blockchain. Moreover, blockchain allows for real-time monitoring and tracing of supply chains, from raw materials to finished goods, expediting recording and verification processes. A blockchain-based supply chain can foster trust between involved parties and end consumers by storing all immutable data on the blockchain.

Blockchain technology is particularly suited for establishing a chain of custody. Once recorded, chain-of-custody transactions become immutable, creating a tamper-proof record. This chain of custody is accessible to all parties on the blockchain, allowing them to verify the record by simply reading it. A chain-of-custody solution enhances transparency, efficiency, and accountability in often obscured supply chain processes.

By increasing transparency and improving product traceability, blockchain technology can help reduce or even prevent fraud in the supply chain. Manipulating the blockchain is challenging, as it is an immutable ledger that can only be updated and validated through network consensus. Furthermore, if a product’s information is recorded on the blockchain, determining its origin becomes straightforward since the data is stored on a shared, distributed ledger.

2.2.7.5. Electronic Voting

Numerous studies have been conducted on electronic voting systems, aiming to minimize the cost of elections while ensuring their integrity by meeting security, privacy, and compliance requirements. Replacing traditional pen-and-paper systems with innovative election systems can reduce fraud and make voting more traceable and verifiable.

Distributed Ledger Technologies (DLTs), such as blockchain, offer a decentralized platform for electronic voting systems through end-to-end verification processes [42]. Blockchain is an appealing alternative to conventional voting systems due to its decentralization, non-repudiation, and robust security features [12].

2.2.7.6. Energy Industry

One of the critical applications of blockchain technology in energy-related domains involves microgrids. Microgrids are localized networks of power sources and loads, integrated and controlled to optimize energy production and consumption efficiencies and improve reliability [43]. These power sources may encompass renewable energy stations, distributed power generators, and energy storage components owned by various organizations or energy providers. A significant advantage of microgrid technology is that it allows consumers to receive the energy they need while producing and selling excess energy to the grid. In microgrids, blockchain technology facilitates, records, and confirms power sales and purchases [44].

Blockchain technology can enable energy trade in smart grids, microgrids, and the Industrial Internet of Things (IIoT) [45]. With bidirectional information flow, blockchain facilitates secure and private energy monitoring and trade, eliminating the need for a central intermediary [46]. Smart contracts can verify programmatic descriptions of anticipated power flexibility, demand response agreements, power requirements and generation balance. Implementing blockchain in energy-related applications can reduce energy costs and enhance system resilience.

2.2.7.7. Insurance

Blockchain technology can provide a platform for insurance companies to streamline transactions with their clients, policyholders, and providers. It enables the negotiation, purchase, and registration of insurance policies, as well as the filing and handling of claims and reinsurance activities. By employing smart contracts, insurance policies can be automated, significantly reducing administrative costs [47]. Claims processing, typically complex and costly due to disputes and misinterpreting policy provisions, can be simplified by organizing insurance policies using precise if-then relationships implemented through digital protocols. This approach can reduce labour and expenses associated with policy implementation, enabling insurance companies to lower their prices and attract more customers. Furthermore, blockchain can help insurance companies develop new automated insurance products for their clients without incurring high administrative costs and overhead. In summary, blockchain can facilitate the global expansion of insurance companies by streamlining operations and reducing costs.

2.3. Internet of Things

The widespread adoption of the internet has enabled seamless connectivity and communication, giving rise to IoT. IoT integrates human interaction with technology, simplifying data sharing and allowing remote management of many “smart” devices, enabling access to specific services. This revolution paves the way for innovative products and business models. Developments in broadband communication, power management, microprocessors, and increasingly reliable memory have prompted digitising various functions and environments. This evolution has brought forth the concept of a “smart” world, generating valuable data applicable in various domains such as smart campuses, smart homes, and smart cities.

2.3.1. IoT Terminology

Researchers offer numerous interpretations of IoT from various perspectives [48]. Some focus on physical objects connected to networks, others on network technology and protocols, while some accentuate semantic elements, including data storage, information, and search. The

European Commission defines IoT as integrating physical and virtual domains to create intelligent environments [49]. Generally, IoT refers to a network of smart devices connected to the internet, capable of identifying and interacting with one another by transmitting and collecting data across the network [50]. This interaction facilitates connections and interactions among people, processes, and objects anytime, anywhere, across any network, and with any service.

Technically, devices embedded with sensors and chips, including NFC and RFID tags, communicate with each other through Internet Protocol (IP) addresses and communication protocols like MQTT, negating the necessity for human intervention [51]. Once these devices connect to the internet, they transmit data to computing solutions such as cloud systems, which amalgamate data storage, analytical tools, and delivery models to provide services for individuals and organizations. Upon reaching the cloud, the data is processed using appropriate software, thereby assisting in establishing an intelligent environment. It involves making informed decisions, predicting potential problems, saving time and money, and establishing perception, network, and intelligent processing as crucial components of IoT.

2.3.2. Challenges of IoT

Several research challenges arise from IoT's unique characteristics, such as the interconnection of multiple smart entities equipped with electronic or mechanical sensors, actuators, and software systems. These entities can perceive, collect, and process environmental data, ultimately acting based on this information, thereby making the management of IoT systems highly intricate. The most notable challenges associated with IoT systems include.

1. **Heterogeneity:** This refers to the diversity of IoT devices, communication protocols, and data types within the IoT ecosystem. Heterogeneity poses challenges for interoperability, privacy, and security, as different devices and protocols may not work harmoniously together, and maintaining data privacy and security can be difficult across multiple systems.
2. **Complexity of Networks:** IoT involves multiple communication and network protocols, creating a complex network environment. The diversity of these protocols can create challenges in efficiently managing the overall network [52], [53].

3. **Poor Interoperability:** This refers to the ability of IoT systems' hardware and software components to share and collaborate on information efficiently. Owing to the decentralisation and heterogeneity of IoT systems, data exchange among different industries, strategic centres, and IoT systems can be arduous, resulting in difficulties in achieving interoperability.
4. **Resource Constraints of IoT Devices:** IoT devices typically have limitations in computing, storage, and power capacities. These constraints implement robust security measures challenging and leave resource-constrained IoT devices vulnerable to malicious attacks.
5. **Privacy Vulnerability:** Privacy is a crucial concern in IoT systems, as they process sensitive user data. Due to the decentralization and complexity of IoT systems, ensuring data privacy is challenging. Additionally, while integrating IoT with cloud computing significantly enhances IoT systems' computing and storage capacities, uploading sensitive data to third-party cloud servers may introduce privacy risks [54]. Therefore, implementing appropriate measures to protect IoT data privacy, prevent unauthorized access, and prohibit disclosure is essential.
6. **Security Vulnerability:** The decentralization and heterogeneity of IoT systems present substantial challenges in ensuring their security. Despite the importance of security in enterprise settings, implementing security measures in resource-constrained IoT systems is difficult. Due to IoT devices' constraints, traditional security approaches such as authentication, authorization, and encryption may not be suitable. Moreover, the absence of timely security firmware updates makes IoT systems more prone to malicious attacks [55].

2.3.3. Opportunities Arising from the Integration of Blockchain and IoT

Integrating blockchain technology with IoT systems presents multiple opportunities, leading to substantial benefits across different industries. These opportunities include:

- **Enhanced interoperability of IoT systems**: Blockchain technology can significantly improve the interoperability of IoT systems by facilitating seamless data exchange between various IoT devices and systems. Blockchain achieves this by converting and

storing IoT data in a standardized format, which enables efficient processing and storage of diverse data types. Additionally, blockchains' peer-to-peer overlay network architecture supports universal internet access, allowing for easy traversal of disparate networks. By harnessing blockchain technology, IoT systems can achieve greater interoperability, leading to more efficient data exchange between devices and systems.

- **Improved security of IoT systems:** Incorporating blockchain technology can significantly enhance the security of IoT systems. This is primarily because blockchain can protect IoT data by storing it as encrypted and digitally signed transactions using cryptographic keys, such as ECDSA [26]. Moreover, integrating IoT systems with blockchain technologies like smart contracts enables automatic firmware updates for IoT devices to address security vulnerabilities, improving overall system security [56].
- **Traceability and Reliability of IoT data:** Storing IoT data on a blockchain ensures its traceability and reliability. Blockchain technology enables the identification and verification of data at any time and from any location. Furthermore, blockchains record all past transactions, rendering them traceable. As proposed by [57], a blockchain-based product traceability system can provide traceable services for suppliers and retailers, allowing them to inspect and verify product quality and authenticity. The immutability of blockchains is critical for maintaining the integrity of IoT data, as recorded transactions become nearly impossible to alter or forge.
- **Autonomic interactions of IoT systems:** Blockchain technology allows IoT systems to interact autonomously. This capability facilitates the automation of transactions without needing traditional intermediaries, such as governments or businesses, in the payment process. The concept of distributed autonomous corporations (DACs), as developed by [57], enables the automation of transactions through smart contracts, which can function without human intervention. This, in turn, leads to cost savings and increased efficiency in the interactions among IoT systems.

By leveraging these opportunities, integrating blockchain with IoT can bring significant advancements in various domains, contributing to a more interconnected, secure, and efficient world.

2.4. Analysis of Blockchain Platforms for Internet of Things

This section provides a comparative evaluation of numerous consensus algorithms that are currently in use. The primary objective of this analysis is to determine the most suitable consensus algorithm for IoT applications.

2.4.1. Classifications of Blockchain Networks

According to [58], blockchain networks can be classified into permissionless and permissioned blockchains. Each of these classifications has distinct characteristics, advantages, and applications. A detailed comparison between the two, along with a thorough examination of their features, is provided in this section.

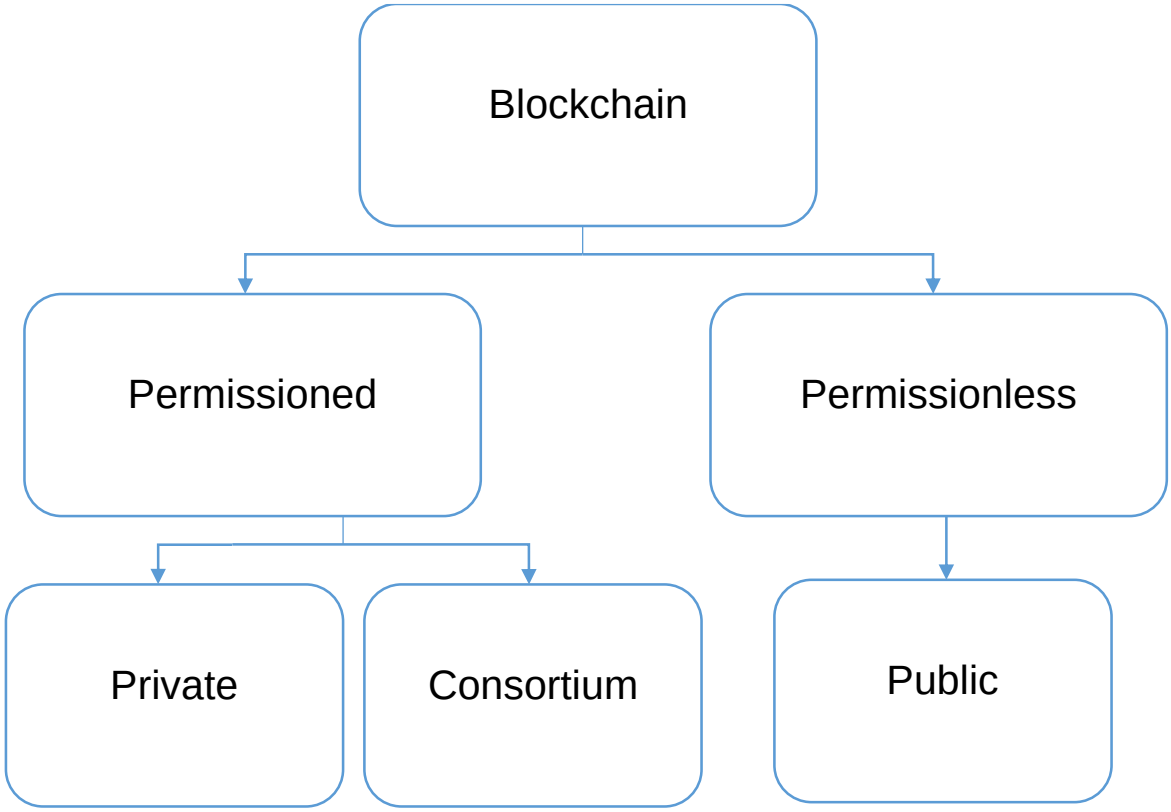

Figure 2.3: Blockchain network classifications.

Figure 2.3 depicts the classification of blockchain networks, with public blockchains falling under permissionless networks, while private and consortium blockchains come under the purview of permissioned networks. This section will cover each blockchain network's distinct features and characteristics (see Figure 2.4).

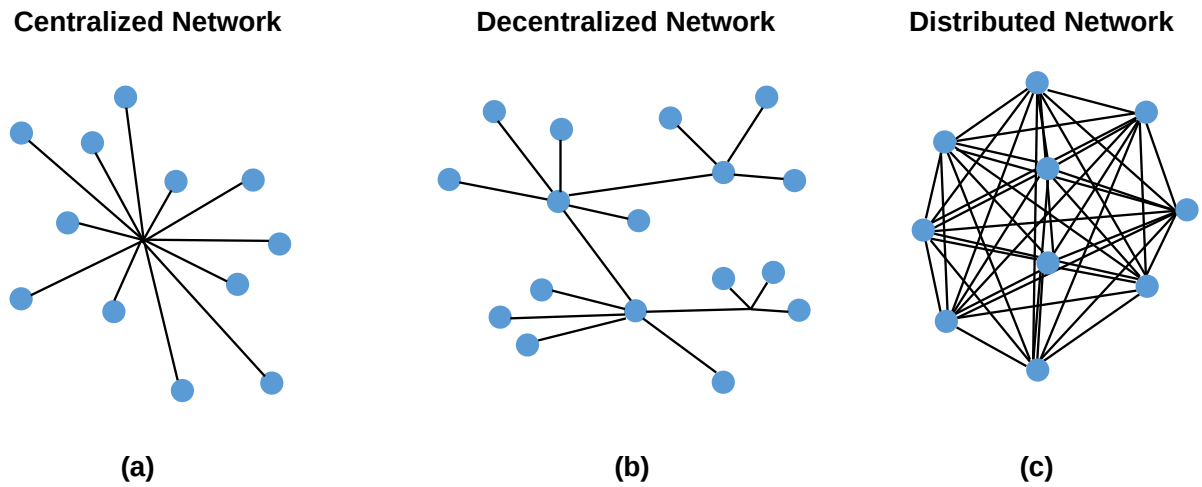

Figure 2.4: (a) Private blockchain network; (b) Consortium blockchain network; (c) Public Blockchain Network.

2.4.1.1. Permissionless (Public) Blockchain Network.

1. **Open Access:** Permissionless blockchain networks offer unrestricted access, allowing anyone to join without prior approval. Participants can freely validate transactions, engage in the consensus process, and access the data stored on the network, as shown in Figure 2.4 (c). This fosters an environment of inclusiveness and encourages participation from diverse entities globally.
2. **Decentralization:** Permissionless blockchains prioritize decentralization, ensuring no single entity controls the network. This fosters trustlessness, censorship resistance, and democratic decision-making within the ecosystem. Decentralization also helps eliminate single points of failure and enhances the network's overall security.
3. **Transparency:** Permissionless blockchains inherently display transaction data and network activity to the public. Such openness promotes transparency, enabling individuals to audit transaction history and foster trust and accountability among participants.

4. **Consensus Mechanisms:** Permissionless blockchains rely on more complex and resource-intensive consensus algorithms, such as PoW or PoS, to maintain security, fairness, and decentralization. These mechanisms help ensure that the network resists malicious attacks and fraudulent activities.
5. **Security:** Since permissionless blockchains are open and decentralized, they benefit from a more extensive network of participants, contributing to increased security through the consensus process. The distributed nature of these networks makes it challenging for bad actors to manipulate or compromise the system.
6. **Use Cases:** Permissionless blockchains are well-suited for DApps and cryptocurrencies like Bitcoin [59] and Ethereum [60], where trust minimization and open access are essential. They also find applications in decentralized finance (DeFi), digital identity, and decentralised data storage, among other use cases.

2.4.1.2. Permissioned (Private) Blockchain Network.

1. **Restricted Access:** In permissioned blockchains, participants require authorization to join the network. Only approved users can validate transactions, participate in the consensus process, or access data. Such a structure ensures that the network involves only trusted entities, heightening security and control (see Figure 2.4 (a)).
2. **Centralization:** Permissioned blockchains exhibit more excellent centralization than permissionless blockchains because a pre-selected group of participants governs them. A selected group's governance provides better control, enables streamlined decision-making, and more effective enforcement of rules and policies.
3. **Privacy:** Permissioned blockchains offer higher privacy levels because access to sensitive information is restricted to authorized users. Such limitations are crucial for organizations needing to safeguard proprietary information or adhere to data protection regulations. The structure of permissioned blockchains further enables confidential transactions and allows selective data disclosure to relevant parties.
4. **Consensus Mechanisms:** Permissioned blockchains often utilize more efficient and scalable consensus algorithms, such as Practical Byzantine Fault Tolerance (PBFT) or Proof of Authority (PoA), leading to faster transaction processing and lower energy consumption. These consensus mechanisms are better suited for environments where participants are known and trusted.

5. **Security:** While permissioned blockchains may not have as many participants as permissionless blockchains, they can still maintain a high level of security through the controlled participation of trusted entities. The authorization process helps ensure that only reputable participants can join the network, reducing the risk of malicious activities.
6. **Use Cases:** Permissioned blockchains are commonly employed in industries that require privacy, data control, and regulatory compliance, such as finance, supply chain management, and healthcare. They are also utilized in interbank transactions, asset tokenization, and enterprise blockchain solutions where control, governance, and collaboration are essential. Ripple [61] and Eris [62] are private blockchain networks.

2.4.1.3. Consortium Blockchain Network.

Consortium blockchain networks are categorized as permissioned blockchain networks. These networks offer a combination of features from private and public blockchain networks. Consortium blockchain networks allow a selected number of nodes to join, but all nodes can read, write, and verify transactions on the ledger, like public blockchains. Regarding transparency and immutability, consortium blockchains fall between private and public models.

A hybrid structure in consortium blockchains offers enhanced efficiency over public blockchains. Such efficiency arises because nodes can endorse peers for executing smart contracts or committing transactions, which reduces redundant computations and boosts network performance and throughput. Compared to private blockchains, consortium blockchains provide higher trustworthiness and security. The absence of governance by a single node for validation and authorization keeps the system decentralized, making it less vulnerable to security threats. Hyperledger [35] and Corda [63] are notable examples of consortium blockchain networks.

Table 2.1 summarizes the various types of blockchain networks and their features.

In addressing the inquiries in this section regarding the suitability of blockchain network platforms for the IoT domain, a comprehensive examination of the characteristics of various blockchain types has been conducted. Pahl et al. [64] developed a flowchart framework to facilitate the selection process, presented in Figure 2.5, which serves as a guide for determining

the most appropriate blockchain network platform. This framework has been utilized to assist in identifying the most suitable blockchain platform for the IoT domain.

Table 2.1: Comparisons among public blockchain, consortium blockchain and private blockchain.

Property	Public blockchain	Consortium blockchain	Private blockchain
Network Type	Distributed	Decentralized	Centralized
Consensus Process	Permissionless	Permissioned	Permissioned
Read Permission	Public	Could be public or restricted	Could be public or restricted
Transaction Validation	All pears	Selected peers	Single peer
Efficiency	Nearly impossible to tamper	Could be tampered	Could be tampered
Throughput rate	Low	High	High
Efficiency	Low	High	High
Centralized	No	Partial	Yes
Energy Consumption	High	Low	Low

The framework is designed to guide answering two crucial questions before incorporating blockchain technology into any software system. Firstly, it determines if blockchain technology is necessary for the software, and secondly, if deemed necessary, it identifies the most appropriate blockchain platform to implement.

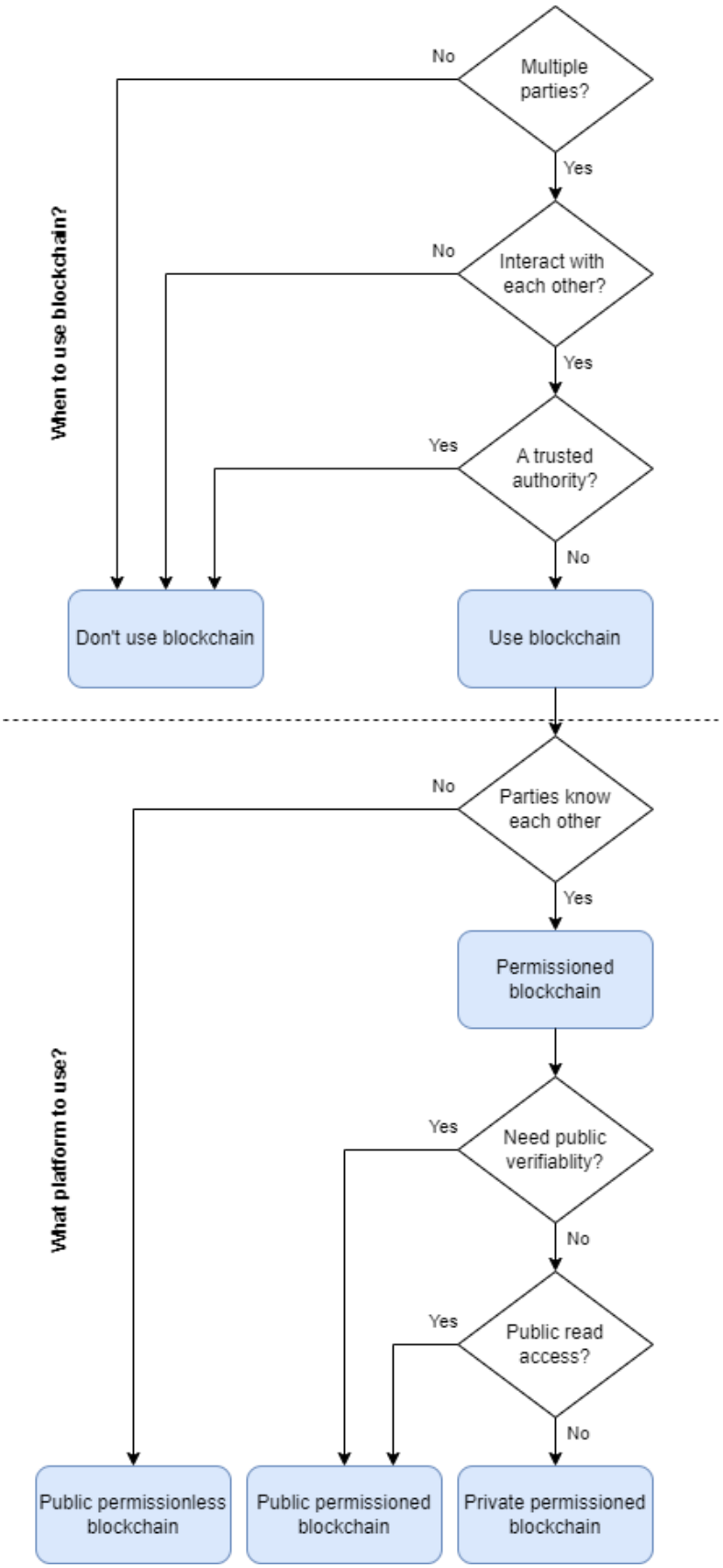

Figure 2.5: A decision flow of blockchain platform.

The flowchart is structured into two sections, with the first section addressing the first question and the second section focusing on determining the required properties of the blockchain platform.

The initial section of the flowchart helps determine if the use of blockchain technology is necessary for the application by assessing the system's requirements for multiple parties with distinct roles or if a single party can fulfil all roles. If it is determined that a blockchain is indeed necessary, the flow proceeds to the next section (see Figure 2.5).

Subsequent sections of the flowchart focus on differentiating between public, private, and consortium blockchains. These decisions are based on factors such as the level of trust among participants, the desired consensus mechanism, and the degree of access control required for the application. By answering these questions, users can navigate the decision flow and ultimately identify the most suitable type of blockchain for their specific use case. The flowchart analysis indicated that a consortium blockchain network would be the optimal fit for IoT applications.

The potential applications of blockchain technology extend across various domains, including finance, government, IoT, healthcare, Artificial Intelligence (AI), supply chain, and more. This section explores numerous blockchain application cases scholars propose worldwide, as illustrated in Figure 2.2.

2.4.2. Consensus Algorithms

A vital characteristic of a blockchain is its decentralized architecture. Contrary to traditional transactions requiring a central authority such as a central bank for financial transactions or a governance entity for public decisions, a blockchain does not need such centralization to validate transactions and maintain an official ledger. This decentralization provides advantages such as eliminating unnecessary costs involving third-party organizations, lessening computational bottlenecks, and enabling all nodes on the network to validate and store blockchain records [65]. Without a central body, blockchain networks require an alternate technique to validate transactions and ensure consistent records across all participants. Consensus algorithms serve that purpose.

Researchers developed various consensus mechanisms and deployed them on open-source platforms. This section compares these consensus algorithms to determine a suitable blockchain platform for a given system. This analysis is needed because many blockchain platforms possess adaptable capabilities to run multiple consensus processes. Moreover, no algorithm is perfect, as each has advantages and disadvantages. Consensus protocols form the backbone of blockchain platforms, making it crucial to investigate them and select the most appropriate one for the specific system under consideration.

Consensus mechanisms ensure blockchain networks' security, reliability, and functionality. They help maintain consistency and agreement among the network's nodes, facilitate transaction validation, and prevent malicious activities such as double-spending. With various consensus algorithms available, it is essential to understand their unique features, strengths, and weaknesses to make an informed decision when selecting a suitable blockchain platform.

The following subsections will discuss the most prominent consensus algorithms, including their key characteristics, advantages, and disadvantages.

2.4.2.1. Proof of Work

To explain PoW, the following terms must be understood:

- A node is a device on the blockchain network that can utilize its computing capacity to validate transactions and communicate information regarding new blocks in the blockchain.
- • A miner is a node in the blockchain network that can mine blocks by calculating an appropriate nonce.

Note that while all miners are nodes, not all nodes are miners.

The fundamental concept of the PoW consensus mechanism is that miners must demonstrate that a significant amount of work has been performed in mining the block. Before publishing a block to the blockchain, miners must solve a cryptographic problem. This cryptographic problem requires calculating a sufficient nonce (number only used once) and adding it to the block data to obtain an adequate block hash according to the difficulty level. Since solving the cryptographic problem necessitates a certain amount of processing power, a significant amount of effort has been invested in mining the block. Nodes in the network will then verify published

blocks using the block data, transaction data, nonce, and output hash, ensuring that the output hash conforms to the current difficulty level.

The PoW consensus process is implemented to accomplish the following:

1. Reduce the likelihood of blocks being simultaneously published.
2. Reduce the likelihood that transactions can be tampered with.

First, signed transactions (using the digital signature scheme described in Section 2.2.3) are broadcast to the network. Nodes on the blockchain network validate these transactions using the transaction data, the sender's signature, and their public key. Verified transactions are then added to the pending transactions' memory pool.

Miners then compile the memory pool's transactions into a block. Without PoW, miners could rapidly and easily create and publish blocks to the network. Multiple blocks might be published simultaneously due to the large number of miners on the network, resulting in multiple forks in the blockchain list. As miners publish blocks rapidly, these forks will continue to split into several forks, making it challenging to identify a single valid blockchain among them. In contrast to PoW, finding an appropriate nonce in the Bitcoin implementation results in a block being mined, on average, every 10 minutes [66]. Such a time frame reduces the chances of two miners solving for a nonce and publishing their block to the network simultaneously. When they happen, the result is a blockchain fork (see Figure 2.6).

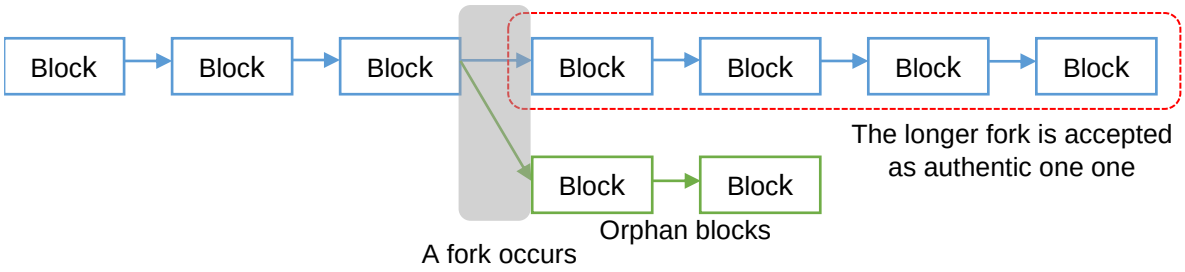

Figure 2.6: Example of a fork in the blockchain.

Nonetheless, it is doubtful that a second block will be uploaded to both forks simultaneously. Over time, the longer fork will be deemed authentic by the PoW protocol. The miners who had been building on the blocks in the shorter fork will subsequently transition to the longer

legitimate fork. The shorter fork is now an orphan block because it no longer advances in length [65].

Additionally, using PoW, after a certain number of blocks have been added to the blockchain, it becomes nearly impossible to alter the transactions by reversing and rebuilding the blockchain. When an attacker attempts to modify the destination of a prior transaction, the block contents and hash are altered accordingly. The block header includes the hash of the preceding block. Therefore, if an attacker attempts to modify a prior block and its hash, they must recalculate all subsequent blocks. PoW makes such an attack infeasible because it demands vast processing power and significantly increases the time required to mine each block. When an attacker manages to mine a changed block, and its succeeding blocks, more verified blocks will have been added to the original blockchain fork. Since the longer fork is deemed authentic, it would be difficult for an attacker to hijack a prior block and mine enough blocks to replace its faulty blocks with valid blocks to authenticate the longer fork.

A 51% attack represents the primary theoretical method for introducing fraudulent transactions in a blockchain. In this scenario, an attacker who controls 51% of the network's hashing power can mine blocks faster than the rest. By doing so, they can create a fork with invalid blocks longer than the legitimate fork. As a result, the network may recognize the attacker's fork as the genuine chain. However, this is not feasible in the real world, as the hardware cost alone would reach $5.46 billion [67]. Even in the unlikely event that an attacker could take control of 51% of hashing power, the financial gain from rewriting the blocks might not outweigh the expense, and it might be more profitable to collect mining rewards by using the hashing power to generate legal blocks. A 51% attack would also reduce trust in the blockchain, lowering the currency's value and decreasing the incentive and value of such an attack.

2.4.2.2. Proof of Stake

PoW scheme necessitates considerable computational resources to resolve intricate mathematical problems, leading to high energy usage. PoS proposes a low-energy alternative to PoW, requiring users to exhibit a certain amount of currency ownership, assuming that individuals with more extensive currency holdings are less inclined to compromise the network. As stake-based selection can lead to inequalities, several alternatives have been suggested that utilize stake size in deciding the next block's forger. For instance, Blackcoin

[68] employs a randomized mechanism to predict the following generator, applying a formula that considers the smallest hash value in conjunction with stake size. Peercoin [69] supports age-based coin selection, where older and larger coin sets have a higher probability of mining the next block.

2.4.2.3. Proof of Elapsed Time (PoET)

Intel initially proposed PoET as an energy-efficient alternative to resource-intensive consensus algorithms, such as PoW. Leveraging Intel's Software Guard Extensions (SGX) technology, PoET aims to maintain PoW's security and decentralization levels while drastically reducing the energy and computational power necessary for consensus.

In the PoET consensus process, participating nodes, also called validators, must wait a randomly assigned period before proposing a new block. The waiting time is produced by a secure and trusted function within the SGX-enabled environment and is kept confidential. Upon completion of their waiting period, the validator can propose a new block, and the first to complete the waiting time and broadcast the proposal earns the right to add the block to the blockchain.

However, PoET has limitations, such as dependence on Intel's SGX technology, which may raise trust issues and potential vulnerabilities as not all hardware supports SGX. PoET's applicability is restricted and potentially unsuitable for the framework under consideration.

2.4.2.4. Practical Byzantine Fault Tolerance (PBFT)

PBFT is a consensus algorithm designed to address Byzantine Faults' challenges in distributed systems, referring to instances where some network nodes behave unpredictably or maliciously, possibly causing inconsistencies and errors. Introduced by Castro and Liskov in 1999 [70], PBFT offers a solution that ensures the reliability and consistency of a distributed system, even in the presence of faulty nodes. PBFT provides a solution ensuring a distributed system's reliability and consistency, even when some nodes are faulty. PBFT operates on a state machine replication model, where all nodes keep a copy of the system's state and execute the same operation sequence. The primary node leads the consensus process for a given period, called a view, proposing a new block while the other nodes, or backup nodes, participate in a

three-phase communication protocol (pre-prepare, prepare, and commit) to reach an agreement on the proposed block.

Several blockchain platforms, including Hyperledger Fabric v0.6, Hyperledger Iroha, and BigchainDB, employ PBFT in their consensus mechanisms. Despite boasting higher transaction processing speeds than computationally intensive consensus methods, PBFT requires a central authority to select backup nodes and a leader, rendering it less decentralized. PBFT suits permissioned blockchain networks, where authority nodes are selected. However, it faces scalability issues with increased backup node numbers due to the increased communication messages required [71]. Increased exchanged messages in the PBFT consensus could contribute to an upsurge in computation energy due to communication and network overheads.

Additionally, PBFT may be vulnerable to Sybil's attacks, where an adversary could create faulty nodes without certificate authority or control a network portion to influence the consensus outcome. Despite these vulnerabilities, PBFT's low latency, low computational overhead, and high transaction processing speed render it suitable for the system under consideration. However, the significant communication cost associated with PBFT makes it unsuitable for more extensive networks and better suited to smaller systems.

2.4.2.5. Delegated Byzantine Fault Tolerance (DBFT)

The NEO blockchain platform introduced the DBFT approach in 2014 [72]. This innovative method blends elements from the Practical Byzantine Fault Tolerance (PBFT) algorithm and a node selection voting mechanism, creating a consensus algorithm that effectively addresses the Byzantine Generals Problem in distributed systems. The DBFT offers enhanced efficiency and scalability compared to traditional Byzantine Fault Tolerance (BFT) algorithms such as PBFT. This performance enhancement is primarily due to the fusion of critical elements from BFT and Delegated Proof of Stake (DPoS) consensus mechanisms, leading to a scalable and efficient consensus approach.

In a DBFT system, network participants elect a group of trusted nodes, known as delegates, to partake in the consensus process. The core responsibilities of these delegates include validating transactions and creating new blocks. The delegate election process typically hinges on a voting mechanism, with network participants staking their tokens or other assets to vote for their

preferred delegates. This democratic voting process bolsters decentralization by inviting the broader community to partake in the delegate selection process.

While the DBFT consensus method presents several potential advantages for system implementation, it also faces challenges like PBFT, primarily related to communication overhead and susceptibility to Sybil attacks. Additionally, the DBFT mechanism has a longer average delay in block creation (approximately 15 seconds) than PBFT. This longer delay might make the DBFT consensus algorithm unsuitable for the system under consideration, mainly if the system demands rapid transaction validation and block creation.

2.4.2.6. Steller Consensus Protocol (SCP)

The Stellar network employs the Stellar Consensus Protocol (SCP), designed by David Mazieres. SCP aims to deliver a secure and efficient consensus while preserving decentralization [73]. SCP is built on the Federated Byzantine Agreement (FBA) model, an expansion of the conventional BFT model.

In SCP, every node in the network individually selects a subset of other nodes, termed quorum slices, with which it aims to reach a consensus. A quorum is a collective group of nodes that agree on a particular value. The intersections of quorum slices form a quorum, which assures that the network can agree, even if malicious or faulty nodes are present. This flexible trust model promotes a more decentralized network than traditional BFT mechanisms, which commonly rely on a fixed set of validators.

SCP presents several benefits, such as reduced computational demands and high throughput, thus making it an appealing choice for specific systems. However, as Pahlajani et al. [74] noted, choosing quorum slices within the network can potentially introduce security vulnerabilities. The local consensus achieved amongst nodes within the subset groups is propagated throughout the network via these intersections, which can introduce weaknesses if not carefully chosen.

Furthermore, SCP might demonstrate higher latency than other consensus mechanisms due to its multi-step process that involves achieving local consensus within subset groups and broader consensus among the entire networks. Thus, SCP may not be the most suitable consensus mechanism for systems requiring swift decision-making and low latency.

Considering these limitations, SCP may not be the best choice for the system under examination. A meticulous evaluation of the system's requirements and constraints is crucial to identify the most suitable consensus algorithm for the specific application.

2.4.2.7. Raft

The Raft consensus protocol is a fault-tolerant distributed consensus algorithm developed to manage replicated logs across a cluster of nodes. Designed by Diego Ongaro et al. [75], Raft emerged as an alternative to the complex Paxos algorithm, offering an easier-to-understand and implement solution while maintaining a similar level of performance and robustness.

Raft operates under the assumption that most nodes in the cluster are operational and can communicate with each other. The protocol revolves around a strong leader concept, wherein one node is elected as the leader, and the remaining nodes function as followers. The leader manages and replicates log entries across the cluster while the followers passively accept updates from the leader and respond to their requests.

The Raft consensus protocol can suit systems requiring a straightforward, easy-to-understand consensus mechanism while ensuring fault tolerance and consistency. However, its reliance on a strong leader can potentially create performance bottlenecks and limit scalability in larger systems. Moreover, network latency or communication issues between the leader and followers might affect the protocol's performance, making it potentially unsuitable for applications with strict latency requirements. Some blockchain platforms like Hyperledger Fabric v1.0 and Quorum employ Raft for consensus.

Table 2.2 compares the applicability of the discussed consensus techniques for the proposed framework. The scale ranges from (Good), denoting the least suitable option, to (Excellent), representing the most appropriate choice. The Raft consensus algorithm demonstrates superior scalability to PBFT consensus algorithms, which require numerous communications between nodes to validate a transaction. As the number of nodes increases, communication requirements follow suit, leading to slower synchronization. Raft exhibits lower latency than DPBFT and SCP and does not require specialized hardware like PoET. Consequently, the Raft consensus algorithm emerges as the most suitable mechanism for the proposed framework.

The subsequent section will delve deeper into applying the Raft consensus method on blockchain platforms to determine an appropriate platform for the proposed system.

Table 2.2: The summary of the comparison of consensus algorithms for IoT.

Criteria	**Consensus Algorithms**				
	PoET	PBFT	DBFT	SCP	Raft
Latency	Low	Low	Medium	Medium	Low
Throughout	High	High	High	High	High
Scalability	High	Low	High	High	High
Computing Overhead	Low	Low	Low	Low	Low
Adversary Tolerance	N/A	<33.3% Fault replicas	<33.3% Fault replicas	Variable	<50% Crash fault
Suitability	Good	Very Good	Good	Good	Excellent

2.4.3. Blockchain Platforms

The previous section thoroughly examined a variety of consensus algorithms vital to blockchain implementation, eventually identifying the Raft consensus algorithm as the most suitable for IoT applications. Due to the pluggable nature of many blockchain platforms, an evaluation and analysis of existing consortium blockchain platforms employing the Raft consensus algorithm are subsequently undertaken. This examination will be based on each platform's unique architecture and transaction flow. The most prominent platforms implementing the Raft consensus algorithm include Quorum, Corda, and Hyperledger Fabric.

2.4.3.1. Quorum Platform

Quorum, a permissioned blockchain platform derivative of Ethereum, was initiated by JP Morgan [76]. Quorum was developed by modifying the core features of Ethereum, such as transaction privacy, consensus algorithms, permissioned network access, and the eradication of transaction fees. It has been designed to cater to diverse industries, including finance and supply chain management, by providing a secure and efficient environment for smart contract execution and transparent ledger maintenance. Primarily employed within the banking industry, Quorum is an open-source platform with pluggable capabilities, allowing for the execution of various consensus protocols.

Quorum's architecture is built upon the Ethereum framework but incorporates modifications enhancing privacy and permissioning. The platform uses a unique consensus mechanism, which improves transaction throughput and reduces energy consumption compared to Ethereum's Proof of Work mechanism. Quorum also introduces an advanced permissioning system, giving organizations control over access to their blockchain networks, thereby ensuring data privacy and regulatory compliance.

Quorum supports the development and execution of smart contracts using Solidity, Ethereum's primary programming language. This compatibility eases the transition for developers acquainted with Ethereum, allowing them to take advantage of Quorum's additional features. Furthermore, the platform integrates with enterprise systems and other blockchain networks, facilitating interoperability and collaboration between organizations.

In summary, the Quorum blockchain platform, derived from Ethereum, is a permissioned network offering enhanced privacy, permissioning, and consensus mechanisms. It is designed to cater to various industries by enabling secure and efficient execution of smart contracts and maintaining a transparent, shared ledger.

2.4.3.2. Corda Platform

Developed by R3, the Corda platform [63] is a permissioned, semi-open-source blockchain designed to serve the financial sector [77]. In contrast to fully open-source platforms, Corda's semi-open-source nature indicates that portions of the source code are retained privately to protect sensitive business applications. Corda's architecture is structured to enable secure and

efficient transactions between consortium participants while focusing on preserving privacy and minimizing fraud potential.

Corda's architecture employs a notary pool to achieve consensus, akin to traditional banking systems. This notary pool comprises a group of trusted nodes that verify transactions and prevent double-spending. This approach circumvents the need for global consensus, as only the parties involved and the notary pool need to agree on each transaction. As a result, Corda provides improved scalability and efficiency compared to other blockchain platforms.

Alongside its unique consensus mechanism, Corda supports the development of smart contracts. This capability enables automated agreement execution between participants. Developers can write these smart contracts in familiar programming languages such as Java or Kotlin, simplifying application development on the platform.

In summary, the Corda platform, a permissioned semi-open source blockchain, was developed by R3 with a primary focus on the financial sector. Its architecture prioritizes privacy, efficiency, and security by utilizing a notary pool to achieve consensus and facilitate smart contract development.

2.4.3.3. Hyperledger Fabric Platform

Hyperledger Fabric is a permissioned blockchain network developed by the Linux Foundation, designed to accommodate the needs of business consortium networks [78]. As an open-source blockchain project, Hyperledger Fabric aims to serve a wide range of business and government use cases while preserving the privacy of the blockchain network. A vital feature of the platform is its ability to manage multiple distributed ledgers within a single system. The platform's modularity and pluggable features are significant assets, allowing for flexible application across diverse use cases. For instance, the ledger data in Hyperledger Fabric does not adhere to a specific required format, making it versatile and suitable for a wide range of applications. Due to the fully pluggable nature of the consensus mechanism, different scenarios can employ distinct consensus mechanisms.

The main components of Hyperledger Fabric's architecture include peers, orderers, endorsers, shared ledgers, chaincode, and member service providers [78]. Peers are network nodes that store a copy of the ledger, execute chaincode (smart contracts), and endorse transactions. The

orderer is responsible for arranging transactions, creating new blocks, and appending them to the ledger in the correct sequence. Endorsers, another essential entity in the network, endorse and validate transactions and determine whether parts of the network are authorized to perform specific ledger actions. Shared ledgers offer a distributed, tamper-proof record of all transactions within the network. Chaincode, the equivalent of smart contracts, outlines business logic and transaction execution rules. The member service provider manages all network participant identities, ensures entity authentication and registration, and grants roles for authorizing and managing actions within a specific ledger.

By leveraging its modular architecture and a pluggable consensus mechanism, Hyperledger Fabric allows customization and flexibility to meet various industries' specific requirements and use cases.

In conclusion, Table 2.3 provides a comparative summary of blockchain platforms suitable for the IoT domain:

- **Area of focus:** Quorum and Corda are primarily designed for the banking and financial industries, respectively, while Hyperledger Fabric is suitable for a broader range of applications, including enterprise and business-to-business solutions.
- **Programming languages:** By supporting commonly used and popular programming languages, the platform can attract more developers and become more adaptable. Quorum supports Solidity as it is a fork of Ethereum. Hyperledger Fabric supports Go, Java, and NodeJS, while Corda supports Java and Kotlin. Cai et al. [79] noted that stable and adaptable implementation is crucial when evaluating prospective blockchain platforms.
- **Transaction rate:** Transaction rates per second vary across these platforms, with Quorum handling a few hundred transactions, Corda processing up to 170 transactions, and Hyperledger Fabric capable of managing up to 2000 transactions.
- **Architecture:** The architectural design of these platforms is another essential consideration. Quorum employs an Order-Execute architecture, where transactions are ordered and then sequentially executed on all participating nodes in the same order [78]. This architecture ensures consistency across the network by enforcing the same transaction order for every node. However, it could impact system throughput due to the sequential nature of transaction execution, potentially leading to network delays or

bottlenecks. In contrast, Corda and Hyperledger Fabric utilize the Execute-Order-Validate (EOV) approach [78]. In this model, transactions are initially executed and then ordered before being validated by the network. In this model, transactions are initially executed, ordered, and finally validated by the network. This architecture affords greater flexibility in handling non-deterministic smart contracts and enhances overall system performance. By decoupling the execution and validation stages, EOV enables more efficient processing, leading to higher throughput and reduced latency within the blockchain network.

Table 2.3: Summary of the comparison of blockchain platforms for IoT.

Criteria	Quorum	Corda	Hyperledger Fabric
Founder	JP Morgan	R3	Linux Foundation
Area of focus	Banking sectors	Financial industry	Enterprise and B2B
Programming languages	Solidity	Java, Kotlin	Go, Java and NodJS
Transaction rate per second	Few hundred	Up to 170	Up to 2000
Architecture	Order-Execute	EOV	EOV

In comparing the blockchain platforms, Corda's design is primarily tailored for the financial industry, aiming to streamline complex business operations and ensure privacy in asset transactions. Corda's unique consensus approach and emphasis on privacy are notable; however, its primary concentration on the financial sector makes it less suitable for IoT applications than Hyperledger Fabric.

Conversely, Quorum, an Ethereum-based platform, offers similar features to Hyperledger Fabric. These include support for the Raft consensus algorithm, robust privacy measures, and smart contracts. Nevertheless, Quorum's strong affiliation with Ethereum and its foundation as

an Ethereum fork potentially restricts its flexibility and adaptability, especially when compared to Hyperledger Fabric.

Upon a comprehensive analysis of the three platforms, Hyperledger Fabric is the optimal choice for implementing a Raft-based consensus algorithm in the IoT domain. The platform's modular architecture allows customization to meet specific application needs. The support for intelligent contracts enables automated transaction execution, contributing to efficiency and accuracy. Moreover, Hyperledger Fabric boasts wide adoption and an active community, which can be beneficial in terms of continuous platform development, problem-solving, and resource sharing.

Thus, Hyperledger Fabric is the ideal candidate for constructing a secure, scalable, and efficient IoT system using blockchain technology, taking full advantage of a Raft-based consensus algorithm [78].

2.5. Overview of Hyperledger Projects

The Hyperledger ecosystem consists of many projects specifically designed to address needs within the blockchain domain. This section outlines the most prominent Hyperledger projects, underscoring their objectives, architecture, and use cases.

2.5.1. Hyperledger Sawtooth

Hyperledger Sawtooth, a scalable, secure, enterprise-grade blockchain platform, highlights modularity as its crucial feature [80]. It utilizes a distinct consensus algorithm named PoET (Proof of Elapsed Time), which promotes energy efficiency and scalability compared to traditional consensus mechanisms like Proof of Work. Furthermore, through its Sawtooth Ethereum (Seth) transaction family, Sawtooth offers compatibility with Ethereum smart contracts, thus facilitating the use of pre-existing Ethereum tools and languages such as Solidity. In terms of suitability, Sawtooth finds its niche in projects related to supply chain management, digital asset management, and smart contract execution.

2.5.2. Hyperledger Iroha

Hyperledger Iroha is a simple, permissioned, modular blockchain platform that integrates seamlessly into various infrastructures [81]. Iroha employs a Byzantine Fault-Tolerant consensus mechanism known as YAC (Yet Another Consensus), which ensures high performance and reliability. Its version of smart contracts, called “commands”, is designed to focus on specific use cases, thus expediting application development [82]. Iroha is ideally suited for projects that require digital asset management, identity management, and interbank transactions.

2.5.3. Hyperledger Indy

Hyperledger Indy is a distributed ledger designed particularly for decentralized identity management [83]. Using zero-knowledge proof cryptography, the platform offers tools and libraries for creating, issuing, storing, and verifying digital identities. Such cryptographic methods enhance privacy and uphold system integrity. In the realm of digital identity, Indy is mainly utilized for self-sovereign identity purposes, providing a platform where individuals and organizations can autonomously manage and control their digital identities [84].

2.5.4. Hyperledger Besu

Hyperledger Besu is an Ethereum-compatible, enterprise-grade blockchain client developed using Java [85]. It supports public and permissioned networks and is compatible with the expansive Ethereum ecosystem. Such compatibility encompasses the Ethereum JSON-RPC API, the Ethereum test suite, and smart contracts penned in Solidity. With a modular design, Besu encourages customization, while its dedicated permissioning system ensures secure and regulated access to the network. Professionals often choose Besu for finance, supply chain management, and asset tokenization.

2.5.5. Hyperledger Fabric

Hyperledger Fabric is a permissioned, modular, and extensible blockchain platform suitable for various industries [86]. It allows for establishing private channels between participants,

ensuring the confidentiality and security of transactions. Fabric's modular architecture supports plug-and-play components, including consensus mechanisms and membership services. This flexibility allows customization to cater to specific enterprise needs. Fabric's smart contracts, known as chaincode, can be written in general-purpose programming languages like Go, Java, and JavaScript. Fabric is ideally suited for supply chain management, finance, healthcare, and IoT projects.

The diverse range of projects within the Hyperledger ecosystem significantly contributes to advancing secure and innovative blockchain technologies. These projects foster collaboration and interoperability across industries and use cases, laying the groundwork for the widespread adoption of blockchain solutions across sectors such as supply chain management, finance, healthcare, and IoT.

2.6. Limitations of Classical Cryptography

As previously stated, blockchain functionality as a secure, decentralized ledger of transactions is underpinned by two vital cryptographic protocols:

1. Asymmetric digital signatures, such as ECDSA and RSA, are based on public-key cryptography.
2. Hashing functions, such as SHA-256, are integral to consensus scheme implementation.

While both cryptographic protocols pose substantial computational challenges for contemporary computers employing classical algorithms, they are not quantum secure. Specifically, Shor's algorithm [19] can break the trapdoor functionality of classical asymmetric digital signatures, while Grover's algorithm [87] can expedite the PoW algorithm, thus heightening the probability of a quantum node controlling the consensus scheme.

2.6.1. Vulnerability of Asymmetric Key Cryptography

The advent of quantum computing introduces considerable challenges to conventional cryptographic protocols such as RSA and ECDSA. These are founded on the computational difficulty of the integer factorization (IF) problem and the Elliptic Curve Discrete Logarithm Problem (ECDLP). Quantum computers, however, can solve these problems in a significantly

reduced timeframe [19]. Since traditional digital signature schemes in blockchains utilize ECDSA, they are vulnerable to quantum attacks.

Shor's algorithm resolves integer factorization and ECDLP in polynomial time [19], posing a significant threat to public-key cryptography, primarily relying on RSA or ECDSA. Classical computers would require an exponential amount of time to solve these problems. Additionally, quantum computers can employ Grover's algorithm to hasten the generation of hashes, potentially enabling the entire blockchain to be reconfigured. Moreover, Grover's algorithm can be adapted to identify hash collisions, thus allowing block substitution in a blockchain without compromising the system's integrity [87].

In Bitcoin, for instance, network transactions are susceptible to quantum attacks. Before a transaction, bitcoin addresses are hashed values of the user's public key; consequently, neither the private nor public key is exposed. However, when a transaction is broadcasted to the network, the user's public key is revealed to validate the transaction [88]. Since quantum adversaries can effortlessly break the ECDSA, they can derive the user's private key from the public key. The adversary could forge the user's digital signature using the private key to authenticate fraudulent transactions. They could impersonate users to publish transactions that transfer funds to themselves or others without their knowledge or permission. If the illegitimate transaction is published and added to the blockchain before the legitimate transaction, the former is accepted, and the latter is compromised [89].

In preparation for a future where quantum computers are widespread, actions must be taken to ensure that cryptographic systems and the corresponding digital signature schemes resist quantum attacks.

2.6.2. Performance Gain for PoW

A quantum computer with a sufficiently large memory register can achieve a quadratic speedup in computation time for the PoW consensus mechanism over any classical device using Grover's algorithm [90]. In a setting where miners search for a nonce that generates a block hash with a specified number of leading zero bits, such as Bitcoin, this speedup can provide substantial advantages for early adopters. When integrated into cryptocurrency mining, quantum computers are anticipated to dominate the entire mining process. Owing to the

cascading effect, when a sufficiently powerful quantum computer is added to a PoW-based blockchain, the hash rate of the entire network will surge, thereby reducing the average time required to calculate a block. Furthermore, the PoW difficulty parameter must be adjusted for the block time. This competitive advantage will incentivize miners to invest more in quantum mining technology, leading to a cycle in which the entire mining industry adopts quantum-based technology [91].

Once quantum miners control the majority of mining, the network becomes immune to 51% attacks based on the quadratic advantage of quantum computers [91]. Consequently, it can be inferred that the quadratic advantage of quantum miners over PoW blockchains only poses a temporary network security issue. The critical factor is introducing a quantum computer with sufficient processing power to escalate the mining power to the point where quantum mining dominates the network.

Alternatives to consensus algorithms, such as PoS, Proof of Space, and Lattice-based PoW, have been proposed to mitigate the quantum computing advantage over classical PoW. PoS is a feasible alternative to PoW, albeit with certain security compromises [92]. Numerous publicly accessible blockchains utilizing PoS, such as ALGORAND [93], and many using Proof of Space, such as the proposed cryptocurrency SpaceMint [94], have emerged. Lattice-based PoW, a variant of PoW that enhances resistance to advantage gain using Grover's algorithm, has been proposed [95].

In conclusion, the limitations of classical cryptography, particularly in the context of blockchain technology, necessitate the development of new cryptographic protocols and consensus algorithms that are resistant to quantum attacks. The growing capabilities of quantum computers pose a considerable threat to the security and integrity of blockchain systems reliant on classical cryptographic techniques. To preserve the security and reliability of blockchain networks, researching, developing, and implementing quantum-resistant cryptographic methods and consensus schemes is essential.

2.7. Post-Quantum Cryptography

As previously discussed, classical blockchain technology relies on cryptographic primitives that must be re-evaluated for a blockchain to be quantum-safe. The focus in the following

sections will be on signature methods that propose addressing the above public-key cryptography vulnerabilities. These signature methods are based on several mathematical principles, including lattice-based cryptography, code-based cryptography, multivariate polynomial cryptography, and hash-based signatures.

2.7.1. Lattice-based Cryptography

Lattice cryptography is one of the most investigated post-quantum signature scheme techniques now. The principal issue that makes lattice-based signature methods promising is the Short Vector Problem (SVP). It is believed that even quantum computers will find it computationally challenging to solve the SVP. Additionally, the Short Integer Solution (SIS) problem based its hardness on SVP_γ, where γ is a scaling parameter in a search for a short vector.

2.7.1.1. Lattices

To comprehend the algorithm used to solve the SVP, it is necessary to have a fundamental grasp of lattices and how they are mathematically expressed. In addition, the lattice-Gaussian distribution is introduced, which is necessary for the algorithms described in subsequent chapters [96].

Definition 2.1 (Lattice)**.** A lattice Λ of $\mathbb{R}^n$ is a discrete subgroup of $\mathbb{R}^d$. In this work, only integer lattices are considered, i.e., $\Lambda \subseteq \mathbb{Z}^d$.

Definition 2.2 (Basic of a Lattice)**.** Let $B = (\mathbf{b}_1, \dots, \mathbf{b}_d) \subset \mathbb{R}^d$ consist of d linearly independent vectors such that [96]

$$\Lambda = \Lambda(B) = \left\{ \sum_{i=1}^{d} x_i \mathbf{b}_i : \; x_i \in \mathbb{Z}, 1 \leq i \leq d \right\}. \tag{2.1}$$

Note that by convention, $\mathbf{b}_i$ are column vectors, and $B\mathbf{k} = k_1\mathbf{b}_1 + \cdots + k_d\mathbf{b}_d$, where $\mathbf{k}$ is a column vector.

Figure 2.7 illustrates a two-dimensional lattice for illustrative purposes.

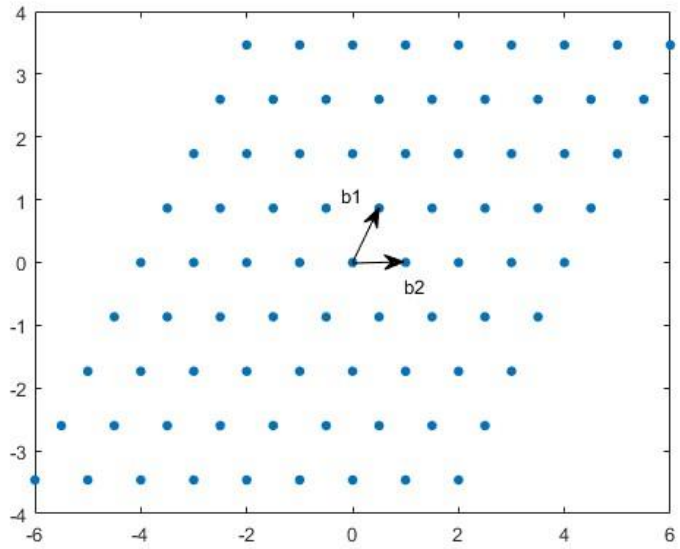

Figure 2.7: Example of a two-dimensional lattice and one such possible basis (b1, b2).

In addition, a distribution over lattices can be defined to arrive at the definition of lattice-Gaussian distributions.

Lattice distribution refers to the probability function defined over a support Λ. Further, lattice distributions are examined in which a density function $\mathbb{R}^d$ induces the probability distribution. The density functions of the form shown below are mainly focused [96].

$$f(\mathbf{x}) = \frac{e^{-\psi(\mathbf{x})}}{Z_\psi} \tag{2.2}$$

For all $\mathbf{x} \in \mathbb{R}^d$. With the potential function $\psi(\mathbf{x})$ and $Z_\psi = \int_{\mathbb{R}^d} e^{-\psi(\mathbf{x})}\, d\mathbf{x}$. Let $P_\Lambda(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^d$ be a lattice distribution induced by the above $f(\mathbf{x})$ [96]:

$$P_\Lambda(\mathbf{x}) = \frac{e^{\psi(\mathbf{x})}}{Z} \tag{2.3}$$

For all $\mathbf{x} \in \mathbb{R}$ with $Z = \sum_{\mathbf{x}\in\Lambda} e^{-\psi(\mathbf{x})}$

To get a sample from a probability distribution specified on a lattice $P_{\Lambda}(\mathbf{Bz})$ with $\mathbf{z} \in \mathbb{Z}$, it is sufficient to randomly sample from $P_{\mathbb{Z}^d}(\mathbf{z}) = \frac{e^{\psi(\mathbf{Bz})}}{Z}$ with $\mathbf{z} \in \mathbb{Z}^d$ and obtain $\mathbf{x}$ as $\mathbf{x} = \mathbf{Bz}$. So, the distribution over the support $\mathbb{Z}^d$ can be sampled rather than having to sample directly from the lattice distribution [96].

$$P_{\mathbb{Z}^d}(\mathbf{z}) = \frac{e^{\psi(\mathbf{Bz})}}{Z} \tag{2.4}$$

Let $\phi(\mathbf{x}) \coloneqq \psi(\mathbf{Bx})$ for all $\mathbf{x} \in \mathbb{R}^d$ such that [96]:

$$P_{\mathbb{Z}^d}(\mathbf{z}) = \frac{e^{\phi(\mathbf{z})}}{Z} \tag{2.5}$$

Additionally, the probability density π can be defined as follows [96]:

$$\pi(\mathbf{x}) \coloneqq \frac{e^{\phi(\mathbf{x})}}{K} \tag{2.6}$$

For all $\mathbf{x} \in \mathbb{R}^d$ with $K = \int_{\mathbb{R}^d} e^{-\phi(\mathbf{x})}\, d\mathbf{x}$

The problem of sampling from a distribution defined on any lattice Λ is therefore reduced to sampling from a probability distribution defined on $\mathbb{Z}^d$.

Definition 2.3 (Gaussian Function)**.** the Gaussian function centred at $\mathbf{c} \in \mathbb{R}^d$ for standard deviation $\sigma > 0$ is defined as [96]

$$\rho_{\sigma,\mathbf{c}}(\mathbf{z}) = e^{-\frac{\|\mathbf{z}-\mathbf{c}\|^2}{2\sigma^2}} \tag{2.7}$$

Combining this Gaussian distribution with a Lattice Λ yields the following definition.

Definition 2.4. The discrete Gaussian distribution over Λ with centre $\mathbf{c} \in \mathbb{R}^n$ and standard deviation $\sigma > 0$ is defined as [96]

$$D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{Bx})}{\rho_{\sigma,\mathbf{c}}(\Lambda)} = \frac{e^{-\frac{\|\mathbf{Bx}-\mathbf{c}\|^2}{2\sigma^2}}}{\sum_{\mathbf{x}\in\mathbb{Z}^d} e^{-\frac{\|\mathbf{Bx}-\mathbf{c}\|^2}{2\sigma^2}}} \tag{2.8}$$

For all $\mathbf{x} \in \mathbb{Z}^d$ where $\rho_{\sigma,\mathbf{c}}(\Lambda)$ is a scaling factor to obtain probability distribution.

The discrete Gaussian resembles the continuous Gaussian but is defined only at the lattice points. Figure 2.8 visually depicts such a distribution so the reader can understand its appearance.

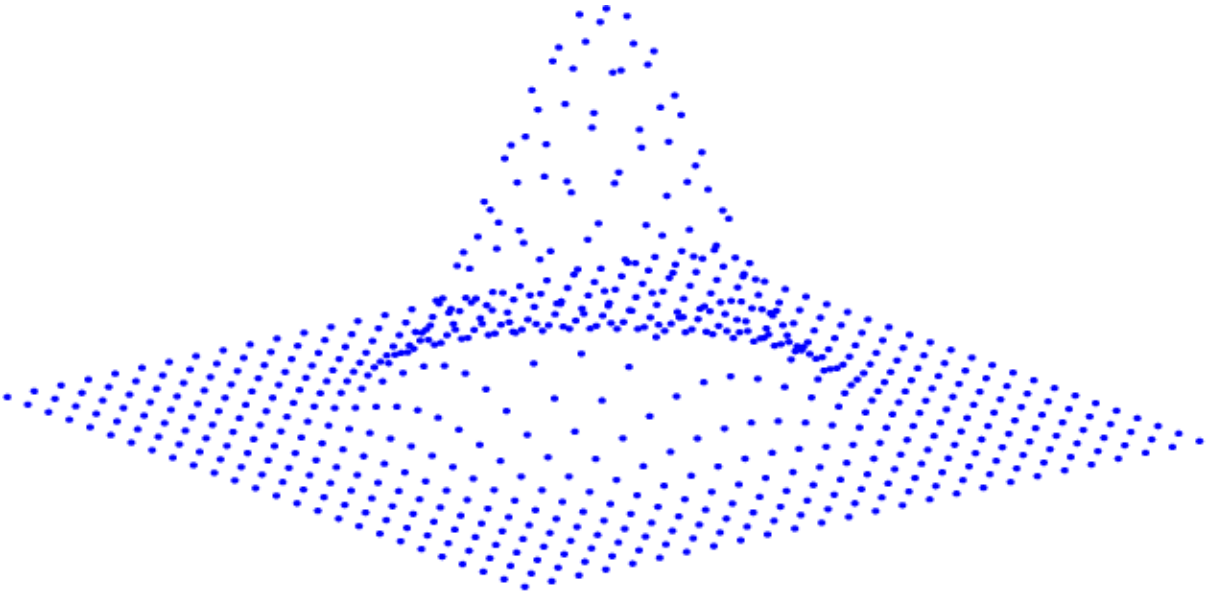

Figure 2.8: An illustration of a two-dimensional lattice Gaussian distribution [97].

2.7.1.2. Short Vector Problem

Three SVP variations are equivalent to one another [98]. The first is to identify the shortest nonzero vector, the second is to calculate the length of the shortest nonzero vector, and the third step is to determine whether the smallest nonzero vector is shorter than a given real integer. $v \in \Lambda(B) \setminus \{0\}\}$ is defined as the shortest nonzero vector in lattice $\Lambda(B)$ such that $\| v \| = \lambda_1(\Lambda(B))$ where B is the lattice basis. The solution to the SVP problem is a, the shortest unique nonzero vector in the lattice.

The SVP_γ is the modified version of the SVP problem that involves finding the shortest vector based on a scaling parameter γ. It can be formalized given a basis B of Λ find nonzero $v \in \Lambda$ such that $\| v \| = \gamma\lambda_1(\Lambda(B))$ [96]. For both classical and quantum computers, the SVP_γ problem remains computationally hard making lattice-based cryptography a candidate for post-quantum secure cryptosystems.

2.7.1.3. Short Integer Solution Problem

Let $a_1, a_2, \dots, a_n \in \mathbb{Z}^{m\times n}$ be n vectors, and q is an integer prime number. The SIS problem is defined as finding a linear combination with non-trivial and small vector $x \in \mathbb{Z}^{m\times n}$ such that $x_1 \cdot a_1 + x_2 \cdot a_2 + \cdots + x_n \cdot a_n = 0 \bmod q$. Considering q-ary lattices, let $A = (a_1, a_2, \dots, a_n)$ be a vector in $\mathbb{Z}^{m\times n}$ and $\Lambda_q^{\perp}(A) = \{z \in \mathbb{Z}^m : Az = 0 \bmod q\}$. The SIS problem is to find the shortest vector problem for the lattice $\Lambda_q^{\perp}$. Which is a computationally hard task, as mentioned above.

2.7.2. Code-based Cryptography

The algorithmic primitive in code-based cryptography uses error correction codes. An asymmetric encryption mechanism, introduced in 1978 by Robert McEliece [99], was the first of these systems whose security is based on the syndrome decoding problem [100]. The public key is a random generating matrix of a randomly permuted private key version that is an arbitrary binary irreducible Goppa code. The ciphertext is a codeword with certain flaws that can only be removed by the private key owner (the Goppa code). Even though certain parameter adjustments have been necessary during the last three decades, no attack has been identified as posing a substantial danger to the system, even on a quantum computer. McEliece's system is very fast because the encryption and decryption procedures are simple, which is beneficial for completing quick blockchain transactions. The McEliece cryptosystem, however, necessitates the storage and handling of large matrices that function as public and private keys. These keys can demand between 100 kilobytes and several megabytes of storage, which may pose a challenge for devices with limited resources. Consequently, implementing the McEliece cryptosystem in such devices can be problematic due to these size constraints.

The McEliece public key cryptosystem is summarized as follows:

1) Let C be length-n binary Goppa code Γ of dimension k with minimum distance $2t + 1$ where $t \approx (n - k)/\log_2(n)$;

2) *Private key:* The McEliece secret key consists of a generator matrix G for Γ. An efficient t-error-correcting decoding algorithm for Γ; an $n \times n$ permutation matrix P and nonsingular $k \times k$ matrix S.

3) *Public key:* The McEliece public key is the $k \times n$ matrix $G' = SGP$.

4) *Encryption:* Compute $\mathbf{m}G'$ and add a random error vector $\mathbf{e}$ of weight $\mathbf{t}$ and length n. Then send $\mathbf{y} = \mathbf{m}G' + \mathbf{e}$.

5) *Decryption:* Compute $\mathbf{y}P^{-1} = \mathbf{m}G'P^{-1} + \mathbf{e}P^{-1} = (\mathbf{m}S)G + \mathbf{e}P^{-1}$. Then, fast decoding is used to find $\mathbf{m}S$ and $\mathbf{m}$.

The security parameters for the McEliece cryptosystem must be chosen for known attacks. The original parameters from [99] are $n = 1024, k = 524, t = 50$

Harald Niederreiter developed a knapsack-type cryptosystem, a dual variant of the McEliece public key cryptosystem, in 1986 [101]. Unlike the McEliece cryptosystem, Niederreiter proposed encoding the message into the error vector instead of representing it as a codeword. The dual variant uses the smaller public key size while slowing down encryption and decryption. The security of both public key cryptosystems is equivalent [102].

2.7.3. Multivariate-based Cryptography

The multivariate public-key cryptosystem is based on multivariate functions over a finite field instead of single-variable NP-hard or NP-complete functions. This family is regarded as one of the key public-key cryptography families capable of withstanding even the most powerful quantum computers in the future. The public is the set of quadratic polynomials [103]:

$$P = (p_1(w_1, \dots, w_n), \dots, p_m(w_1, \dots, w_n)), \tag{2.9}$$

where each p_i is a nonlinear polynomial in $\mathbf{w} = w_1, \dots, w_n$ [103]:

$$z_k = p_k(\mathbf{w}) := \sum_i P_{ik} w_i + \sum_i Q_{ik} w_i^2 + \sum_{i>j} R_{ijk} w_i w_j \tag{2.10}$$

At any given value, the evaluation of these polynomials corresponds to either the encryption or verification procedure.

The main drawback of multivariate schemes is the large public key size. Further research is needed for better decryption speed and reduced key size [104]. Presently, some of the most promising multivariate-based schemes involve the use of square matrices with random quadratic polynomials, cryptosystems derived from Matsumoto and Imai's algorithm, and schemes based on hidden field equations (HFE) [105], [106], [107]. These approaches can produce signature sizes comparable to those generated by RSA and ECC-based signatures, making them attractive alternatives in cryptographic systems.

2.7.4. Hash-based Cryptography

Like any other digital signature technique, hash-based digital signature systems rely on a cryptographic hash function. The security of these methods is determined by the hash function's collision resistance rather than the difficulty of a mathematical problem. Collision-resistant hash functions might be a prerequisite for a digital signature method to sign many documents with a single private key. This method dates back to the late 1970s when Lamport developed a one-way function-based signature scheme [108]. This schema uses a one-way function, and the security parameter n is a positive integer number

$$f: \{0,1\}^n \rightarrow \{0,1\}^n, \tag{2.11}$$

and a cryptographic hash function

$$g: \{0,1\}^* \rightarrow \{0,1\}^n. \tag{2.12}$$

1) *Private key:* 256 pairs of numbers chosen uniformly at random. Each number is 256 bits long, and the total generated numbers are 16KB. The private key $2n$ bit strings of length n.

2) *Public key:* Each number of the private key is hashed, creating 512 different hashes of 256-bit length. It can be said that the public key consists of $2n$ bit strings of length n.

3) *Signature generation:* The message m is signed using the private key, where each bit from the message digest, one number from the private key is chosen. As a result, this signature consists of a series of n-bit strings, each length n.

4) *Signature verification:* To verify the signature, a verifier calculates the digest of message m, then checks whether each bit of the hashed message is the same as the corresponding hash from the public key.

The key and signature generation of Lamport's one-time signature scheme is very efficient, but the signature size is large.

Subsequently, hash-based signature schemes were developed by R. Merkle [109]. Presently, variants of the extended Merkle signature method (XMSS) [110], such as XMSS-T and SPHINCS [111], are viewed as promising hash-based signature schemes for the post-quantum era, originating from the Merkle tree scheme. Owing to their performance, XMSS and SPHINCS might be unsuitable for blockchain applications. Numerous advancements have been made, positioning hash-based signatures as a potential alternative to RSA and elliptic curve signature systems.

2.8. Features and Challenges of Post-Quantum Cryptography for IoT Integration

In contemporary cybersecurity research, integrating post-quantum cryptography into IoT frameworks is critical. This amalgamation of two sophisticated areas not only presents multifaceted challenges but also reveals distinctive features that can significantly fortify the robustness and functionality of IoT systems. It heralds a new era of secure communication systems equipped to operate proficiently amidst quantum computing realities.

A distinctive feature of post-quantum cryptography is its quantum resistance and immunity against quantum computer-based attacks [112]. The emergence of quantum computing technologies, characterized by their capability for processing data on an unprecedented scale

and speed, has raised concerns in the cryptographic community due to their potential to shatter existing cryptographic systems. In this regard, the quantum-resistance characteristic of post-quantum cryptography is vital. Incorporating post-quantum cryptography algorithms into IoT systems effectively protects against potential threats that potent computational attacks from quantum computers might pose.

Further, post-quantum cryptography leverages the concept of quantum ‘hard problems’, such as the factorization of large integers and the resolution of the discrete logarithm problem [113]. These tasks are considered computationally infeasible for quantum computers to solve. Such ‘hard problems’ enhance the security of IoT systems, particularly as these systems often face the challenge of balancing resource constraints against the need to maintain rigorous security measures.

Another noteworthy feature of post-quantum cryptography is its potential to offer unprecedented security assurance. When implemented correctly, post-quantum cryptography can deliver information-theoretic security, surpassing the security level achievable by classical cryptographic systems [114]. Under information-theoretic security, the encryption becomes unbreakable even with unlimited computational resources available to an adversary, thus providing an unmatched level of security.

Despite these compelling features, integrating post-quantum cryptography with IoT is challenging. IoT devices are inherently resource-constrained, making post-quantum implementation complicated, given their larger key sizes and increased computational resource requirements compared to conventional cryptographic algorithms [115]. These constraints affect the efficiency of cryptographic processes, leading to latency increases that may disrupt communication synchronicity between IoT devices and their corresponding computing servers [116].

Balancing the heightened security provided by post-quantum with optimising other vital parameters, such as energy consumption and computational efficiency, is another considerable challenge. This equilibrium is particularly critical within IoT networks where the requirement for enhanced security often conflicts with the demand for energy and computational efficiency.

In conclusion, the fusion of post-quantum and IoT signifies a considerable shift in secure communication technologies, marking the dawn of a new cryptographic era in the quantum

age. Although the inherent challenges of this transition are substantial and require ongoing research and innovation to overcome, the unique features of post-quantum and the potential benefits of its integration with IoT offer a persuasive incentive to confront these challenges directly. Future research must address the existing challenges and explore post-quantum's unique features in the IoT context.

2.9. Summary

This chapter comprehensively reviews the literature on blockchain technology, IoT, and the interplay between these two domains, focusing on cryptography. The aim is to understand the potential of blockchain technology for addressing the unique challenges of IoT systems and explore the limitations of classical cryptography and the potential of post-quantum cryptography in this context.

The fundamentals of blockchain technology are examined, including critical components such as distributed ledgers, cryptographic primitives, consensus mechanisms, and smart contracts. These elements are crucial for the successful integration of blockchain into IoT applications. Concurrently, the core concepts, characteristics, and challenges associated with IoT are discussed, highlighting the potential of blockchain to address issues such as security and scalability.

The limitations of classical cryptography in blockchain technology and IoT are explored, focusing on the vulnerabilities of asymmetric key cryptography, such as RSA and ECDSA, and the potential performance gains in PoW mining. These limitations emphasize the need to develop new cryptographic approaches to ensure the security and integrity of blockchain networks in a quantum computing era.

Post-quantum cryptography is introduced as a potential solution to the limitations of classical cryptography. Various post-quantum signature methods are considered, including lattice-based cryptography, code-based cryptography, multivariate polynomial cryptography, and hash-based signatures. These methods aim to provide robust security against quantum attacks and ensure the continued viability of blockchain technology for IoT applications.

In summary, this literature review thoroughly examines the intersection of blockchain technology, IoT, classical cryptography, and post-quantum cryptography, highlighting the need for further research and development in these areas to ensure the future security, scalability, and interoperability of blockchain and IoT systems.

Chapter 3 Post-Quantum Algorithms for Blockchain

3.1. Introduction

This chapter contributes to the emerging body of scholarly research on post-quantum cryptography, explicitly focusing on post-quantum algorithms. It adds a new dimension to the discourse by thoroughly examining and comparing various post-quantum signature algorithms within blockchain technology. It critically scrutinizes the efficiency of these algorithms and contemplates their potential integration within blockchain infrastructures.

Through a meticulous analysis, the chapter highlights the strengths and weaknesses of each algorithm, thereby providing a foundation for determining the most viable options for real-world applications. In doing so, it assists in bridging the gap between theoretical cryptographic research and practical implementation within blockchain systems.

Moreover, the chapter illuminates a pathway towards developing quantum-resistant blockchain systems. This signifies a significant advancement, as it could play a pivotal role in securing digital transactions and data in an era where quantum computing could potentially disrupt existing cryptographic safeguards.

The insights presented in this chapter have the potential to guide researchers and practitioners working in the fields of cryptography and blockchain technology. Demonstrating how post-quantum algorithms can be leveraged to bolster blockchain security underscores the need for and feasibility of creating quantum-resistant blockchain systems.

Hence, the contribution of this chapter lies in its in-depth exploration of post-quantum algorithms for blockchain, its analytical comparison of various post-quantum signature algorithms, and its role in steering the future direction of quantum-resistant cryptographic solutions.

3.2. Transitioning from Classical to Quantum-Resistant Blockchain

3.2.1. Security in Blockchain's Public Key Infrastructure

The robustness of public-key cryptosystems against conventional computational attacks is typically gauged using the bit security level metric. This measurement quantifies a classical computer's computational workload to initiate a brute-force attack. For instance, a 1024-bit security in an asymmetric cryptosystem signifies that breaching it with a classical computer demands computational efforts comparable to a brute-force approach on a 1024-bit cryptographic key. Table 3.1 details the security levels of several renowned symmetric and asymmetric cryptosystems.

Table 3.1: Reference security levels for popular symmetric and asymmetric cryptosystems.

Security Level	Symmetric Cryptosystem Key Size	RSA Key Size	ECDSA Curve Key Size
80	2TDEA (112 bits)	1024 bits	prime192v1 (192 bits)
112	3TDEA (168 bits)	2048 bits	secp224r1 (224 bits)
128	AES-128 (128 bits)	3072 bits	secp256r1 (256 bits)
192	AES-192 (192 bits)	7680 bits	secp384r1 (384 bits)

The expense associated with breaching contemporary 80-bit security cryptosystems using classical computers is projected to range from tens of thousands to several hundred million dollars. Cryptosystems with 112-bit security are anticipated to remain impervious to conventional computational attacks for three to four decades [117]. However, studies indicate that 160-bit elliptic curves could be compromised using a quantum computer with 1000 qubits, and for 1024-bit RSA, approximately 2,000 qubits would be necessary [118]. Such vulnerabilities extend beyond cryptosystems built on integer factorization principles (like

RSA) or elliptic curves (such as ECDSA and ECDH). They also encompass those rooted in challenges like the discrete logarithm problem [119], which can be swiftly addressed using Shor's algorithm.

Table 3.2 outlines the principal attributes of the most significant public-key cryptosystems susceptible to quantum threats. Additionally, the table incorporates features of other pertinent cryptosystems that either stand to be breached or will experience substantial ramifications from quantum assaults in connection with Shor's and Grover's algorithms.

Table 3.2: Main blockchain and popular cryptosystems impacted by the quantum threat.

Algorithm	**Main Affected Blockchains/DLTs**	**Function**	**Classical Security Level**	**Estimated Post-Quantum Security Level**
SHA-256	Bitcoin, Ethereum, Dash, Litecoin, Zcash, Monero, Ripple, NXT, Byteball	Hash Function	256 bits	128 bits (Grover)
Ethash (Keccak-256, Keccak-512) SHA-256	Ethereum	Hash Function	256/512 bits	128/256 bits (Grover)
Scrypt	Litecoin, NXT	Hash Function	256 bits	128 bits (Grover)
RIPEMD160	Bitcoin, Ethereum, Litecoin, Monero, Ripple, Bytecoin	Hash function	160 bits	80 bits (Grover)

Keccak-256 RIPEMD160	Monero, Bytecoin	Hash function	256 bits	128 bits (Grover)
Keccak-384	IOTA	Hash function	384 bits	192 bits (Grover)
ECDSA	Bitcoin, Ethereum, Dash, Litecoin, Zcash, Ripple, Byteball	Signature	128 bits	Broken (Shor)
RSA-1024	-	Signature, Encryption	80 bits	Broken (Shor)
RSA-2048	-	Signature, Encryption	112 bits	Broken (Shor)

3.2.2. Hash Function Security

In contrast to public-key cryptosystems, traditional hash functions are generally considered more resilient against quantum computing threats. This perspective is rooted in the challenges of creating quantum algorithms for NP-hard problems [120]. While contemporary studies have introduced new hash functions specifically designed to thwart quantum intrusions [121], a prevailing recommendation is to augment the output length of conventional hash functions. This advice stems from potential quantum offensives utilizing Grover's algorithm, which can amplify brute force attack capabilities by a quadratic factor [18]. More specifically, Grover's algorithm presents two primary avenues of threat to a blockchain:

- Initially, the algorithm can be leveraged to identify hash collisions, enabling the substitution of entire blockchain blocks. For instance, the research detailed in [122] suggests using Grover's algorithm to detect collisions in hash functions. They deduced that for a hash function to deliver an n-bit security level, it must output 3*n bits. This insight implies that several existing hash functions may be inadequate for post-quantum scenarios, necessitating functions like SHA-2 or SHA-3 adjustments to enhance output dimensions.

- Moreover, Grover's algorithm can be harnessed to expedite the mining processes in blockchains such as Bitcoin. This acceleration in nonce generation can lead to rapid blockchain recreation, subsequently compromising their integrity.

Furthermore, Shor's algorithm also poses a tangible threat to hash functions. If a quantum adversary breaches a blockchain's hash function, it could employ Shor's algorithm to falsify digital signatures, impersonate blockchain users, and misappropriate digital assets. For context, Table 3.2 enlists the primary attributes of prevalent hash functions adopted by significant blockchains, shedding light on how quantum computing might alter their security parameters.

3.2.3. Post-Quantum Initiatives

In 2016, the NIST initiated a program and competition to identify the future standards for post-quantum cryptography signature schemes. By the 2017 deadline, 59 encryption schemes had been submitted for consideration. Numerous candidates were eliminated throughout multiple rounds, and three digital signature algorithms were selected: Falcon, Dilithium, and SPHINCS+ [123]. These schemes were carefully evaluated based on various factors, including key size, security of the underlying algorithms, potential vulnerabilities, and more.

3.2.4. Efficiency Metrics for Post-Quantum Cryptosystems

To ensure optimal efficiency in a blockchain environment, a post-quantum cryptosystem should exhibit the following key characteristics:

- **Small key sizes.** Devices interfacing with a blockchain benefit from using succinct public and private keys. Using shorter keys reduces storage needs and eases computational challenges linked to key management. Such compactness gains importance, especially in blockchain networks that incorporate IoT end devices. Characteristically limited in computational and storage capacities, these devices are pivotal in the evolving technological landscape. The rise of IoT and its intersection with technologies like deep learning [124] has been noteworthy [125], [126], [127]. However, the challenges, predominantly in security [128], [129], [130], curtail its synergistic integration with blockchains and its broader acceptance.

- **Small signature and hash length.** Blockchains fundamentally archive transactional data, encompassing user signatures and associated data/block hashes. Thus, an expansion in the length of signatures/hashes directly correlates to increased blockchain dimensions.
- **Fast execution.** Postquantum algorithms need to function with high agility to accommodate extensive transactional throughput. Typically, swift processing implies reduced computational demands, which is essential for ensuring the inclusion of devices with limited resources in blockchain interactions.
- **Low computational complexity.** While interlinked with execution speed, it is pivotal to differentiate that swift processing on specific hardware does not innately mean a cryptosystem is computationally efficient. Some algorithms may demonstrate accelerated processing on platforms like Intel's Advanced Vector Extensions 2 (AVX2) but might not fare as well on ARM-based systems. Balancing computational depth, processing time, and hardware compatibility becomes essential.
- **Low energy consumption.** Specific blockchains, exemplified by Bitcoin, have been critiqued for their significant energy footprints, primarily attributed to their consensus mechanisms. However, other variables, including hardware specifications, communication transactions, and especially the intricacies of security protocols, contribute substantially to power utilization [131], [132].

3.3. Overview of Post-Quantum Signature Algorithms

3.3.1. Falcon Signature Algorithm

Falcon is a lattice-based signature scheme over NTRU that NIST selected as a finalist in the NIST post-quantum cryptography contest round 3. Falcon utilizes the GPV framework with NTRU lattices as a post-quantum signature algorithm, and as a trapdoor sampler, it uses a novel technique known as fast Fourier sampling [133].

Gentry, Peikert, and Vaikuntanathan created the GPV framework in 2008 to obtain secure lattice-based signatures. The following is a high-level description of that framework:

- The public key is used to generate q −ary lattice Λ, which contains a full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ where $m > n$;
- The private key is used to generate $\Lambda_q^{\perp}$, which contains $\mathbf{B} \in \mathbb{Z}_q^{m \times m}$, and is the lattice orthogonal to Λ modulo q. At the same time, the rows of **A** and **B** need to be a pairwise orthogonal: $\mathbf{B} \times \mathbf{A}^{\mathrm{t}} = \mathbf{0}$;
- The message m's signature is a short value $\mathbf{s} \in \mathbb{Z}_q^m$ and it should verify $\mathbf{s}\mathbf{A}^{\mathrm{t}} = H(\mathrm{m})$;
- To compute a valid signature, first, compute a preimage $\mathbf{c_0} \in \mathbb{Z}_q^m$, which verifies $\mathbf{c_0}\mathbf{A}^{\mathrm{t}} = H(\mathrm{m})$, where c_0 is not necessarily required to be short and $m \geq n$. Then, a vector $\mathbf{v} \in \Lambda_q^{\perp}$ close to $\mathbf{c_0}$ is computed using matrix **B**. $\mathbf{s} = \mathbf{c_0} - \mathbf{v}$ is a valid signature.

Falcon, like other signature algorithms, has three phases:

1) *Key pair generation:* f and g short polynomials are chosen randomly using an appropriate distribution. The matching F and G polynomials are then found in the solution of the NTRU equation. In this case, the public key is a basis for a $2n$ dimension lattice, where n is typically 512 or 1024.

$$\begin{bmatrix} -h & I_n \\ qI_n & O_n \end{bmatrix} \tag{3.1}$$

The corresponding private key is another basis for the same lattice.

$$\begin{bmatrix} g & -f \\ G & -F \end{bmatrix} \tag{3.2}$$

$g, f, G,$ and F need to fulfil the following equations.

$$h = g/f \bmod w \bmod q \tag{3.3}$$

$$fG - gF = q \bmod w \tag{3.4}$$

2) *Signature generation:* The message and a random nonce are first hashed into polynomial c modulo w. Next, a pair of short polynomials (s_1, s_2) are generated using

the knowledge of the secret lattice basis (f, g, F, G) such that $s_1 = c - s_2 h \bmod w \bmod q$, where the signature is s_2.

3) *Signature verification:* After computing s_1 using the hashed message c and s_2, it should be verified that (s_1, s_2) is a short vector with the process integer computations mod q.

3.3.2. Dilithium Signature Algorithm

The CRYSTALS-Dilithium lattice-based signature, proposed by Ref. [134], is another finalist in the NIST post-quantum cryptography contest. The Dilithium signature algorithm can be summarized in the following steps:

1) *Key pair generation:* Initially, a matrix $\mathbf{A}$ with polynomial entries in the ring $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ is generated, where n is a power of 2. Then, the two private key samples s_1 and s_2 are generated randomly. Finally, the second part of the public key is calculated from $\mathbf{t} = \mathbf{A}\mathbf{s_1} + \mathbf{s_2}$, where the public key is $(\mathbf{A}, \mathbf{t})$ and the private key is $(\mathbf{s_1}, \mathbf{s_2})$.

2) *Signature generation:* The potential signature is calculated as $\mathbf{z} = \mathbf{y} + c\mathbf{s_1}$, where $\mathbf{y}$ is a vector of polynomials and the challenge c is generated using digest and a vector $\mathbf{w}_1$. $\mathbf{y}$ needs to be less than the parameter γ_1. $\mathbf{w}_1$ is then high-order bits of the coefficients of vector $\mathbf{Ay}$, and every coefficient w in $\mathbf{Ay}$ can be written as $w = w_1 \cdot 2\gamma_2 + w_2$, where $|w_2| \leq \gamma_2$. Thus, $\mathbf{w}_1$ is the vector, including w_1. Afterwards, the rejection sampling is used to avoid the dependency of$\mathbf{z}$ on the secret key and prevent the leakage of information about the secret key.

3) *Signature verification:* The verification process computes $\mathbf{w}_1'$ and accepts if all the coefficients of $\mathbf{z}$ are less than $\gamma_1 - \beta$ from $\mathbf{Az} - c\mathbf{t}$ and if c is the hash of the message and $\mathbf{w_1'}$.

3.3.4. SPHINCS+ Signature Algorithm

SPHINCS [111] was designed by Bernstein, Hopwood, Hülsing, Lange, Niederhagen, Papachristodoulou, Schneider, Schwabe, and Wilcox-O'Hearn as a stateless hash-based signature scheme and was the first signature scheme to propose parameters to resist quantum cryptanalysis.

SPHINCS+ [123] is an advanced and optimized version of the SPHINCS signature scheme. Built on a layered approach, SPHINCS+ integrates several intricate constructs to ensure its robustness and efficiency. This section delves into the apparatus underpinning SPHINCS+.

SPHINCS+ employs a hierarchy of trees, usually comprised of d layers of h/d height each. The design paradigm is based on a few-tree (Few-Time) signature scheme. If h denotes the total height and d represents the number of layers, each layer will consist of a hypertree of height h/d.

At the heart of the SPHINCS+ scheme lies the Winternitz One-Time Signature (WOTS+). For a security parameter n, a checksum and message function M is defined. Let $t = \left\lceil \frac{\log (\text{len } (M)+\text{len } (M)\times(w-1))}{\log w} \right\rceil$. Each WOTS+ key encompasses t chains of length w. A signature involves traversing a specific number of steps, $0 \leq \text{steps} < w$, in each chain determined by the message and its checksum.

Hash functions form the essence of SPHINCS+. Here, $F, H,$ and T are PRFs instrumental in node computation, tree traversal, and public key derivation. These are often instantiated with cryptographic primitives like ChaCha or BLAKE.

$$PK = T\left(\text{ Seed }_{pub}\right)$$

The Forest of Random Subsets (FORS) is a distinctive feature, replacing the traditional HORST. For a message digest M, a FORS tree yields k values $M_1, M_2, \ldots, M_k$. Generating a signature necessitates creating authentication paths for all these k values.

Sign and Verification Process:

For a message M :

- Determine the digest $D = H(M)$.

- Using the digest, derive k values and their corresponding FORS tree authentication paths.
- Produce the WOTS+ signature for the FORS tree's root.
- Traverse the hypertree to derive a signature for D. Verification involves using the public key, message M, and the signature to reconstruct the root and verify its authenticity.

Parameters and Efficiency:

SPHINCS+ offers a variety of parameter sets to cater to different security and efficiency needs. Depending on the selection, variables like the height h, Winternitz parameter w, and others are modified to ensure an optimal blend of signature size and computation.

3.4. Performance Evaluation of Post-Quantum Cryptographic Schemes

Traditional cryptographic systems are confronted with profound challenges in the quantum computing landscape. The superior computational abilities of quantum computers can potentially render many existing cryptographic techniques obsolete. Given this backdrop, exploring quantum-resistant cryptographic solutions has emerged as a paramount endeavour. The third round of NIST's Post-Quantum Cryptography Standardization initiative presents a promising set of resilient candidates against quantum threats. This section delves into a rigorous examination and analysis of the performance attributes of these noteworthy digital signature algorithms.

3.4.1. Key Size and Quantum Security Evaluation

Table 3.3, delineated below, catalogues the key dimensions, signature sizes, and the corresponding quantum security claims associated with each algorithm.

Table 3.3: Selected digital signature schemes of the NIST third round.

Algorithm	Type	Claimed Quantum Security	Public Key	Secret Key	Signature
Falcon – 512	Lattice-based	108 bits	0.89 KB	1.28 KB	0.71 KB
Falcon – 1024	Lattice-based	252 bits	1.79 KB	2.30 KB	1.33 KB
Dilithium 2	Lattice-Based	112 bits	1.31 KB	2.52 KB	2.47 KB
Dilithium 3	Lattice-based	169 bits	1.95 KB	4.00 KB	3.35 KB
Dilithium 5	Lattice-based	241 bits	2.59 KB	4.86 KB	4.65 KB
SPHINCS+-SHAKE256-128f-simple	Hash-based	128 bits	32 bytes	64 bytes	17 088 bytes
SPHINCS+-SHAKE256-192f-simple	Hash-based	192 bits	48 bytes	96 bytes	35 664 bytes
SPHINCS+-SHAKE256-256f-simple	Hash-based	256 bits	64 bytes	128 bytes	49 856 bytes
SPHINCS+-SHA-256-128f-simple	Hash-based	128 bits	32 bytes	64 bytes	17 088 bytes
SPHINCS+-SHA-256-192f-simple	Hash-based	192 bits	48 bytes	96 bytes	35 664 bytes

SPHINCS+-SHA-256-256f-simple	Hash-based	256 bits	64 bytes	128 bytes	49 856 bytes
SPHINCS+-Haraka-128f-simple	Hash-based	128 bits	32 bytes	64 bytes	17 088 bytes
SPHINCS+-Haraka-192f-simple	Hash-based	192 bits	48 bytes	96 bytes	35 664 bytes
SPHINCS+-Haraka-256f-simple	Hash-based	256 bits	64 bytes	128 bytes	49 856 bytes

A critical examination of the data in Table 3.3 yields several discerning insights:

- **Lattice-based schemes (Falcon and Dilithium)**: The algorithms from this category offer a varied range of quantum security levels, from 108 to 252 bits. Notably, the Falcon variants, Falcon-512 and Falcon-1024, stand out for their relatively diminutive signature sizes. While the Dilithium variants exhibit slightly larger keys and signatures, they remain competitive, considering the quantum security levels they proffer.
- **Hash-based schemes (SPHINCS+ variants)** demonstrate a relatively uniform key size, spanning 32 to 64 bytes. However, they present a challenge with their considerably extensive signature lengths, especially when juxtaposed with their lattice-based counterparts.

3.4.2. Computational Performance Metrics Analysis

Evaluating cryptographic schemes extends beyond merely scrutinizing key and signature sizes; computational efficiency remains an essential metric. This analysis was measured on an Intel(R) Core(TM) i7-1165G7 @ 2.80GHz processor.

Table 3.4 captures the computational performance metrics associated with key generation, signing, and verification processes for the candidate algorithms.

Table 3.4: Performance comparison of post-quantum digital signature algorithms selected by NIST.

Algorithm	Key Generation (cycles)	Signing (cycles)	Verification (cycles)
Falcon – 512	21,697,480	616,326	84,273
Falcon – 1024	63,624,161	1,148,744	196,434
Dilithium 2	300,751	1,355,434	327,362
Dilithium 3	544,232	2,348,703	522,267
Dilithium 5	819,475	2,856,803	871,609
SPHINCS+-SHAKE256-128f-simple	9,649,130	239,793,806	12,909,924
SPHINCS+-SHAKE256-192f-simple	14,215,518	386,861,992	19,876,926
SPHINCS+-SHAKE256-256f-simple	36,950,136	763,942,250	19,886,032
SPHINCS+-SHA-256-128f-simple	5,590,602	138,610,500	7,757,942
SPHINCS+-SHA-256-192f-simple	8,227,944	232,973,880	11,768,382

SPHINCS+-SHA-256-256f-simple	21,763,590	468,188,036	11,934,164
SPHINCS+-Haraka-128f-simple	9,137,070	232,172,172	13,148,448
SPHINCS+-Haraka-192f-simple	13,399,816	392,561,468	20,424,354
SPHINCS+-Haraka-256f-simple	35,650,224	832,534,808	22,061,746

Upon detailed study of Table 3.4, the following observations can be drawn:

1. **Efficiency in Key Generation:** Falcon showcases remarkable efficiency in key generation, particularly Falcon-512. The SPHINCS+ with SHA-256 variants in hash-based schemes depict a considerable edge over their SHAKE and Haraka counterparts.
2. **Signing Efficiency:** Lattice-based schemes, notably Falcon, exhibit a distinct advantage in signing efficiency. Within the hash-based domain, the SHA-256 variants of SPHINCS+ are more time-efficient.
3. **Verification Efficiency:** Falcon's efficiency is further exemplified in the verification process, with Falcon-512 setting a benchmark. Although lagging behind Falcon, the Dilithium variants remain substantially efficient, especially when juxtaposed against the expansive cycles required by SPHINCS+ variants.

In light of the computational metrics and the operating environment of the Intel(R) Core(TM) i7-1165G7 processor, it becomes evident that lattice-based algorithms, especially Falcon, offer a synergistic blend of quantum security and computational efficiency. This blend is particularly pivotal for blockchain applications, where security and efficiency are prized.

Figure 3.1 visually represents the average execution times (in milliseconds) across the selected digital signature schemes, further elucidating the comparative efficiencies.

Considering the intricate interplay of quantum resilience, computational efficiency, and the specifics of blockchain demands, selecting a post-quantum cryptographic scheme mandates a nuanced understanding and judicious evaluation.

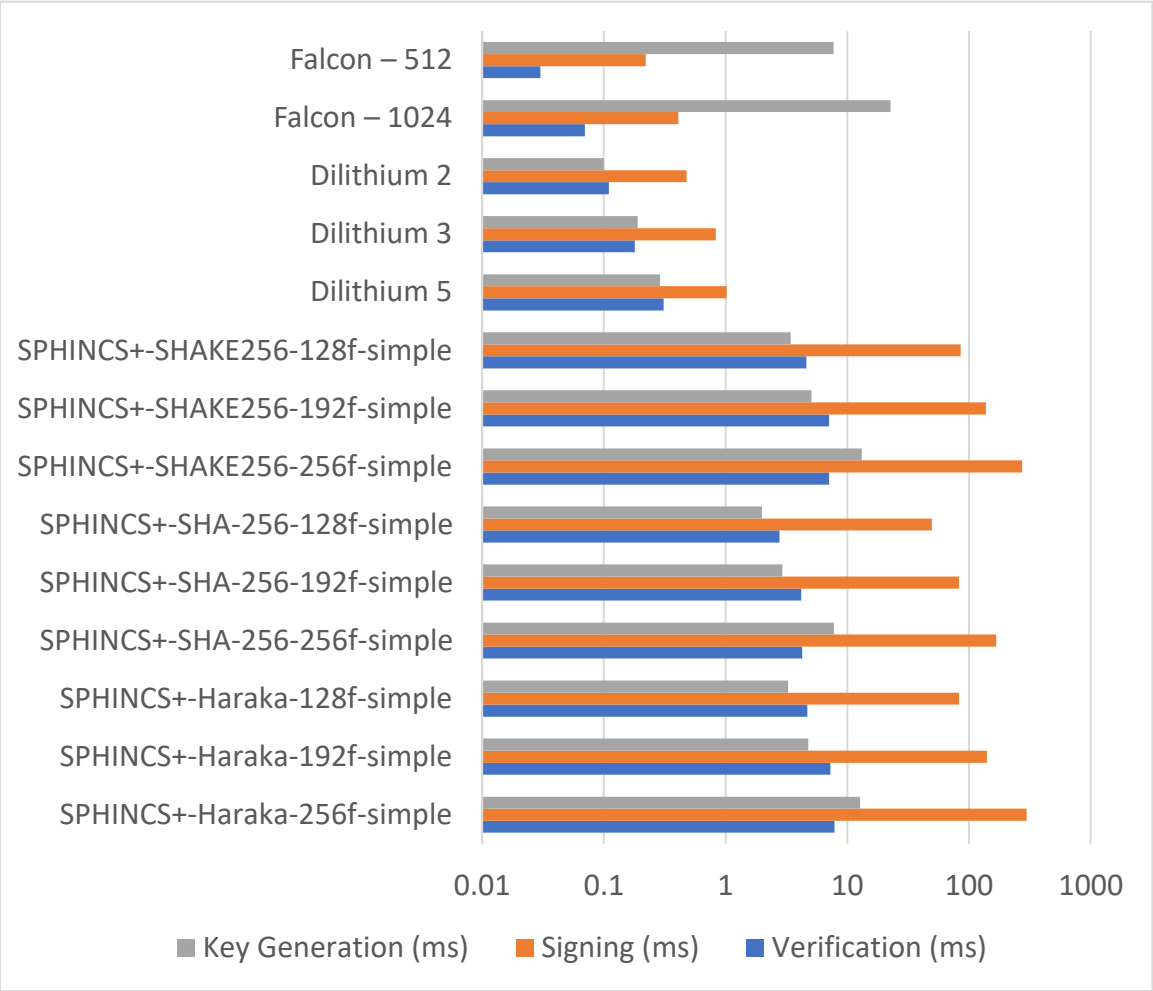

Figure 3.1: Comparison of the average execution times (in milliseconds) of NIST-selected digital signature schemes.

3.4.3. Implications of Signature Schemes for Blockchain Applications.

Blockchain technology, grounded in cryptographic principles, is at an intersection where the imminent surge of quantum computing capabilities challenges its foundational security protocols. The exploration and understanding of post-quantum digital signature schemes thus

become a strategic imperative for blockchain applications' continued viability and evolution. Delving into these implications:

- **Scale, Efficiency, and Scalability:** Blockchain networks, especially those underpinning popular cryptocurrencies, hinge on the capability to process transactions rapidly and efficiently. The computational performance metrics (as evident in Table 3.4) are vital indicators of how signature schemes can impact this aspect. For instance, Falcon's key generation, signing, and verification prowess render it a prime contender. However, the considerable signature lengths of SPHINCS+ variants might present bottlenecks in block sizes and transaction propagation speeds, affecting the scalability of the entire blockchain network.
- **Quantum Security and Resilience:** As highlighted in Table 3.3, the quantum security levels of these signature schemes provide insights into their resilience against quantum adversarial attacks. Lattice-based algorithms, notably Falcon and Dilithium, stand out with their robust security measures, making them potential front-runners in the quest to quantum-proof blockchain ecosystems.
- **Storage Implications and Network Growth:** Given blockchains' decentralized and persistent nature, key and signature sizes have direct storage implications. With the continual growth of the blockchain, accumulating an extensive ledger of transactions, compact signature schemes like Falcon provide a more storage-efficient trajectory, safeguarding the network's ability to scale sustainably.
- **User and Node Dynamics:** Blockchain's multifaceted ecosystem, constituting regular users, miners, and nodes, presents diverse operational requirements. While users may emphasize transaction validation speeds, miners and nodes might be more concerned about computational overheads, block propagation times, and consensus mechanisms. Therefore, selecting a signature scheme should be holistic, accounting for these varied stakeholders.
- **Adaptability and Technological Evolution:** Cryptographic research's dynamic nature necessitates signature schemes adaptable to enhancements or shifts in the cryptographic landscape. Schemes that can be upgraded or complement other cryptographic components ensure the blockchain can evolve without necessitating disruptive overhauls.
- **Scalability Considerations:** As blockchain networks aim for more extensive adoption and integration into mainstream systems, scalability becomes a paramount concern.

Factors such as transaction throughput, block propagation speed, network latency, storage efficiency, and interoperability all interlink with the choice of signature scheme. Schemes that offer compact signatures, swift computations, and modular designs will inevitably support more scalable blockchain architectures.

In essence, the cryptographic decisions made in the present will be foundational in charting the trajectory of blockchain systems in a post-quantum era. With quantum computing on the horizon, integrating quantum-resistant cryptographic tools with blockchain is not just a matter of operational efficiency but of strategic importance, ensuring longevity and security in the face of evolving computational threats.

3.5. Existing Post-Quantum Blockchain Proposals

An increasing number of studies have explored post-quantum blockchains or modifications to existing blockchains to address potential quantum threats [135], [136], [137]. For example, a model for sharing sensitive industrial data via publicly distributed networks is proposed in [138]. This model is compatible with the Inter-Planetary File System (IPFS) and Ethereum, supporting Diffie-Hellman Key Exchange using SIDH. In [139], Ethereum is further adapted with the Rainbow multivariate-based cryptosystem, and its performance is compared to the traditional Ethereum implementation (based on ECDSA).

In [140], researchers suggest enhancing Bitcoin (which employs the Koblitz curve secp256k1 and SHA-256 during the ECDSA signature process) with TESLA# [141], utilizing BLAKE2 [142] and SHA-3 [143]. In [12], a transparent e-voting protocol built on a blockchain using Niederreiter's code-based cryptosystem is introduced to demonstrate resistance to quantum attacks.

Various studies [121] have recommended the development of quantum-resistant blockchains. In [144], a quantum-resistant transaction authentication method based on lattice-based cryptography is presented, applying a standard transaction model to prevent quantum attacks. Similarly, a lattice-based signature approach for establishing a post-quantum blockchain suitable for implementing a cryptocurrency is discussed in [121].

Additionally, commercial blockchains have evaluated and addressed the implications of quantum computing. DLTs such as IOTA's Tangle [145] claim to offer higher resistance to quantum attacks affecting nonce search [146] than Bitcoin. IOTA also benefits from using one-time hash-based signatures (Winternitz signatures) instead of ECC. Moreover, IOTA plans to use ternary hardware (as an alternative to standard binary hardware) and implement a newly developed hash function called CURL-P, which is currently being audited. Several blockchains have been developed to replace Bitcoin in the post-quantum era, like the Quantum-Resistant Ledger [147], which substitutes secp256k1 with XMSS.

While existing research has presented various post-quantum blockchain proposals, there is a noticeable absence of proof-of-concept implementations, particularly in IoT environments, and evaluations of their resistance to quantum attacks, such as those outlined by NIST. These limitations leave critical questions unanswered regarding the practicality, performance, and security of proposed solutions when applied to real-world scenarios, specifically in the context of IoT systems.

To address these gaps, this research aims to provide a comprehensive proof-of-concept that not only demonstrates the practical viability and effectiveness of a post-quantum blockchain solution but also focuses on its application within IoT environments. Furthermore, the proof-of-concept will be subjected to rigorous testing against quantum threats, including those identified by NIST, to evaluate its resilience against potential quantum attacks.

In doing so, this work will contribute valuable insights into the feasibility, performance, and security of post-quantum blockchain solutions for IoT applications and their resistance to quantum attacks. This proof-of-concept will serve as a reference for other researchers and developers seeking to enhance the security of their blockchain systems in the face of potential quantum threats, particularly within IoT environments. The goal is to foster the adoption of post-quantum cryptographic techniques in blockchain technologies and ensure their long-term security and viability in an era of advancing quantum computing capabilities.

3.6. Challenges and Opportunities in Implementing Post-Quantum Cryptography in Blockchain

Implementing post-quantum cryptographic algorithms in existing blockchain systems presents challenges and opportunities. Successfully addressing these challenges and harnessing the opportunities will be crucial in securing blockchain systems against the looming threat of quantum computing. This section discusses the key challenges and opportunities of integrating post-quantum cryptography into blockchain technology.

3.6.1. Integration with Existing Systems

One of the primary challenges in implementing post-quantum cryptography is integrating existing blockchain systems built on classical cryptographic primitives. Suitable post-quantum schemes must be identified to ensure a smooth transition without compromising security or disrupting ongoing operations, and their performance and compatibility with legacy systems must be thoroughly tested. Developing new protocols and software tools that support post-quantum cryptography in blockchain systems is essential for facilitating seamless integration.

3.6.2. Performance Trade-offs

Post-quantum cryptographic algorithms generally exhibit larger key sizes, longer signature lengths, and increased computational requirements than their classical counterparts. These performance trade-offs can affect the efficiency, latency, and scalability of blockchain systems. To address these issues, ongoing research is needed to optimize post-quantum algorithms, explore hybrid cryptographic solutions, and investigate new techniques for efficient implementation in blockchain systems without sacrificing security.

3.6.3. Scalability and Network Efficiency

Blockchain networks must efficiently handle a growing number of transactions and users. However, the larger key and signature sizes associated with post-quantum cryptographic schemes can exacerbate scalability challenges by increasing storage and bandwidth

requirements. To mitigate these concerns, researchers and developers need to investigate methods to compress keys and signatures, reduce storage overhead, and minimize the impact of post-quantum cryptography on the overall performance of blockchain networks.

3.6.4. Standardization Efforts and Regulatory Considerations

Standardizing post-quantum cryptographic algorithms is critical in their adoption and deployment in blockchain systems. Organizations like the NIST are crucial in evaluating and recommending post-quantum schemes for widespread use. As the standardization process progresses, developers and organizations must be prepared to adopt these new standards and adapt their systems accordingly. Additionally, the evolving landscape of data protection, privacy, and cybersecurity regulations must be considered when implementing post-quantum cryptography in blockchain systems, as these regulations can impact the adoption of new cryptographic technologies.

In conclusion, implementing post-quantum cryptography in blockchain systems presents challenges and opportunities that require careful consideration and strategic action. Researchers and developers can work collaboratively to ensure blockchain technology's long-term security and stability in the post-quantum era by focusing on integration, performance optimisation, scalability, and standardisation.

3.7. Strategies for Post-Quantum Blockchain Integration

Integrating post-quantum cryptographic algorithms into existing blockchain systems is a critical and complex task that demands a well-planned and strategic approach. This section discusses several strategies to facilitate the seamless integration of post-quantum cryptography into blockchain technology, ensuring that these systems remain secure and viable in the face of quantum computing threats.

3.7.1. Gradual Transition to Post-Quantum Cryptography

A gradual transition from classical cryptographic schemes to post-quantum alternatives can help minimize potential disruptions and ensure the smooth integration of new algorithms. This

approach progressively updates the underlying cryptographic primitives and protocols while maintaining backward compatibility with existing systems. Combining classical and post-quantum algorithms during the transition phase, hybrid cryptographic schemes can provide additional security and flexibility.

3.7.2. Thorough Testing and Validation

Before implementing post-quantum cryptographic algorithms in blockchain systems, it is essential to rigorously test and validate their security, performance, and compatibility with existing components. This process includes evaluating the resistance of the chosen algorithms to quantum attacks, assessing their performance in computation and communication overheads, and ensuring that they comply with established standards and guidelines. In addition, the potential impact of these new algorithms on the overall system performance and user experience should be carefully analysed and mitigated as necessary.

3.7.3. Interoperability and Standardization

To facilitate the integration of post-quantum cryptography into blockchain systems, it is crucial to ensure interoperability among different implementations and platforms. This can be achieved by adopting standardized post-quantum cryptographic algorithms and protocols, which enable seamless communication and interaction between various blockchain networks and components. Additionally, collaborating with standardization organizations, such as the NIST, can help ensure compliance with established guidelines and promote the widespread adoption of secure and reliable post-quantum solutions.

3.7.4. Education and Awareness

The successful integration of post-quantum cryptography into blockchain systems requires raising awareness among developers, users, and stakeholders about the potential threats posed by quantum computing and the importance of adopting quantum-resistant solutions. This can be accomplished through educational initiatives, training programs, and dissemination of research findings that highlight the benefits and challenges of implementing post-quantum cryptographic algorithms in blockchain systems. By fostering a better understanding of post-

quantum cryptography, stakeholders will be better equipped to make informed decisions regarding adopting and integrating these advanced technologies.

3.7.5. Continuous Research and Development

The field of post-quantum cryptography is constantly evolving, with new algorithms and techniques being developed to address emerging challenges and improve upon existing solutions. Therefore, continuous research and development are vital to keeping pace with advancements in both quantum computing and post-quantum cryptography. By staying abreast of the latest developments and actively engaging in research, developers and practitioners can ensure that their blockchain systems remain secure and up to date in the face of ever-changing quantum threats.

In summary, successfully integrating post-quantum cryptographic algorithms into blockchain systems demands a strategic approach encompassing gradual transition, thorough testing, interoperability, education, and continuous research. By adopting these strategies, the blockchain community can work collaboratively to ensure the long-term security and viability of distributed ledger technology in the era of quantum computing.

3.8. Summary

This chapter has comprehensively analysed the challenges and opportunities of quantum computing to blockchain technology. The rapid advancements in quantum computing can potentially undermine the security of classical cryptographic schemes, which underpin most existing blockchain systems. Consequently, there is an urgent need to explore and implement post-quantum cryptographic algorithms to ensure blockchain networks' ongoing security and reliability.

In this chapter, the fundamentals of post-quantum cryptography have been discussed, and an overview of various post-quantum cryptographic schemes has been provided, focusing on lattice-based signature algorithms such as Falcon, Dilithium and SPHINCS+. These algorithms have demonstrated promising performance characteristics, making them strong contenders for replacing classical cryptographic schemes in blockchain systems.

Moreover, a performance comparison of pre-quantum and post-quantum cryptographic algorithms has been presented, highlighting the trade-offs between security, key size, and computational efficiency. The analysis shows that while post-quantum schemes generally exhibit larger key sizes and signatures, they provide higher security against quantum computing attacks.

The chapter has also explored existing post-quantum blockchain proposals, which have aimed to adapt current blockchain systems or create new ones resilient to quantum attacks. These proposals include the integration of the Rainbow multivariate-based cryptosystem into Ethereum, enhancing Bitcoin with TESLA# and utilizing BLAKE2 and SHA-3, and the development of Quantum-Resistant Ledger, which substitutes secp256k1 with XMSS. Despite these efforts, there is still a need for more proof-of-concept implementations and evaluations of their resistance to quantum attacks.

The challenges and opportunities associated with implementing post-quantum cryptography in blockchain technology have been discussed, including integration with existing systems, performance trade-offs, scalability, network efficiency, standardization efforts, and regulatory considerations. Addressing these challenges and harnessing the opportunities will be crucial in securing blockchain systems against the emerging threat of quantum computing.

In conclusion, developing and deploying post-quantum cryptographic algorithms in blockchain systems represent a critical and timely research area that will shape the future of secure and resilient distributed ledger technology. By addressing the challenges and opportunities presented by quantum computing, researchers and practitioners can help ensure that blockchain technology remains a robust foundation for a wide range of applications in the face of emerging quantum threats.

Chapter 4 Post-Quantum Hyperledger Fabric Blockchain in IoT domain

4.1. Introduction

Integrating blockchain, IoT, and post-quantum cryptography heralds unprecedented innovations and complexities in a realm where technology is incessantly evolving. This chapter meticulously navigates the multifaceted domain of crafting a Post-Quantum Hyperledger Fabric Blockchain suited explicitly for IoT applications, focusing mainly on the real-time acquisition and processing of temperature and humidity data.

4.1.1. Comprehensive Contribution

This chapter manifests a substantial and multifaceted contribution to the field of quantum-resistant blockchain solutions tailored for IoT scenarios. It unveils a groundbreaking and nuanced framework which enables the development and integration of a quantum-secure Hyperledger Fabric solution, characterized by its crypto-agility and capability for a seamless transition to quantum-resistant states.

- **Development and Proposal of a Novel Framework:** The elucidation of this innovative architecture adds an unexplored dimension to existing paradigms, pioneering a pathway for subsequent research and enhancements in integrating blockchain, post-quantum cryptography, and IoT. This exploration lends itself to many applications, enriching the discourse on the practical implications and innovations in blockchain technologies amid the advancements in quantum computing.
- **In-depth Comparative Analysis and Integration:** The profound comparative analysis between the evolved Post-Quantum Hyperledger Fabric and each NIST candidate unfolds nuanced understandings and strategic insights, catalysing the optimization and implementation of highly adaptable quantum-resistant solutions in varied real-world contexts.
- **End-to-End Process Detailing of IoT Data Management:** The chapter renders intricate details and comprehensive insights into the holistic management of IoT data

within the developed quantum-secure blockchain, encompassing every phase from data acquisition to secure transmission and final ledger appending. This portrayal fortifies the development of secure and efficient systems and advances practical knowledge in the domain, enhancing the reliability and security of IoT applications against quantum threats.

4.1.2. Implications and Innovations

The synergistic integration of innovations and explorations discussed in this chapter accentuates the pivotal role of blockchain technologies in fortifying IoT applications against looming quantum threats. It furthers the conversation on the feasibility and necessity of quantum-resistant blockchain technologies, exemplifying the possibilities and practical advancements this integration can bring forth.

By juxtaposing theoretical profundity with practical implications, this chapter serves as a seminal reference, illuminating how researchers, technologists, and industry practitioners endeavour to sculpt the future landscape of secure, innovative, and quantum-resistant blockchain and IoT ecosystems.

This comprehensive contribution distinctly clarifies the innovative framework, detailed analyses, and practical advancements this chapter offers to the existing knowledge base, providing a coherent and holistic understanding of the contribution to the readers.

4.2. In-Depth Analysis of Hyperledger Fabric

Hyperledger Fabric, a part of the Linux Foundation’s open-source Hyperledger project, is among the most actively developed permissioned blockchain systems [148]. The modular design of Hyperledger Fabric enables easy interchange of consensus, endorsement, and storage protocols. Being open-source and widely adopted across various industries [149], Hyperledger Fabric offers a high level of flexibility, availability of source material, and robust community support.

4.2.1. Hyperledger Fabric Architecture

Traditional blockchain technologies like Bitcoin and Ethereum utilize an order-execute architecture [78]. This architectural design involves organising and executing transactions sequentially across all participating peers in the network, as depicted in Figure 4.1. The key characteristics of this approach include:

- **Transaction ordering before execution:** Transactions are consistently gathered and sorted within the architecture before being executed on each peer to maintain coherence across the network.
- **Global state synchronization:** All peers in the network maintain the same state, execute identical transactions, and produce consistent results. This ensures that the blockchain remains secure and tamper-proof.
- **Limited smart contract support:** The order-execute architecture is primarily tailored for domain-specific languages like Solidity, which is used to create smart contracts on the Ethereum platform. It does not support general-purpose programming languages, such as Go and Java.
- **Scalability limitations:** The sequential nature of transaction execution and the requirement for all peers to execute transactions in the same order present challenges in terms of scalability, affecting the network's throughput and performance.

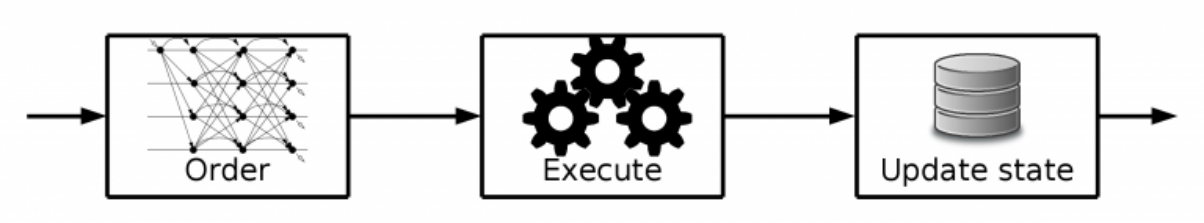

Figure 4.1: Order-execute architecture.

While the order-execute architecture is suitable for certain blockchain use cases, it may face scalability and smart contract support limitations. Alternative architectures, such as the EOV model used by Hyperledger Fabric [150], have been proposed to address these limitations. As illustrated in Figure 4.2, the EOV architecture consists of three distinct phases:

1. *Execution Phase:* During this phase, a smart contract, called chaincode in Hyperledger Fabric, is executed on one or more endorsing peers to process transactions. Endorsers are responsible for simulating transactions and validating them against defined endorsement policies. This allows for non-deterministic smart contracts since endorsers execute transactions independently before being added to the blockchain.
2. *Ordering Phase:* Endorsed transactions are then sent to the ordering service, which organizes them into a block. The ordering service employs a consensus algorithm to ensure all participating peers receive the same transaction sequence, maintaining consistency across the network and preventing double-spending attacks.
3. *Validation Phase:* Network peers validate the transactions within the block upon receiving a block from the ordering service. They verify that endorsement policies have been satisfied and that the transactions' read and write sets have not been invalidated by other concurrent transactions. If the transactions pass these checks, peers update their ledgers with the new block.

The EOV architecture offers several advantages over the order-execute model:

- **Enhanced Scalability:** By separating execution and ordering phases, the EOV architecture allows for greater parallelism, which can improve the network's throughput and performance.
- **General-Purpose Language Support:** Hyperledger Fabric enables chaincode to be written in general-purpose programming languages like Go, Java, and JavaScript, allowing developers to leverage existing skills and tools.
- **Greater Flexibility:** The EOV architecture supports private data collections, enabling confidential transactions and data storage. It also allows for customizable endorsement policies, providing increased control over transaction validation requirements.
- **Improved Security:** By differentiating the roles of endorsing peers and orderers, the EOV architecture minimizes collision risk and enhances overall network security.

Compared to the traditional order-execute model used by platforms like Bitcoin and Ethereum, Hyperledger Fabric's EOV architecture offers increased scalability, flexibility, and security. Subsequent sections will delve deeper into Hyperledger Fabric's architecture.

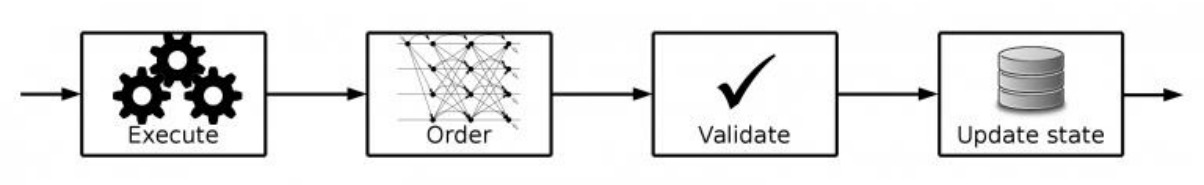

Figure 4.2: Execute-order-validate architecture.

4.2.2. Components of Hyperledger Fabric Architecture

This section presents an overview of key terms and concepts frequently used within the Hyperledger Fabric blockchain framework. Understanding these terms is essential for comprehending the framework's functioning and architecture.

In Hyperledger Fabric, nodes have distinct functions and responsibilities. These nodes can be classified as clients, peers, orderers, endorsers, and committers:

- **Client Node:** A client node is a crucial network element responsible for initiating transactions and interacting with the underlying blockchain infrastructure. It serves as an interface between end-users and the blockchain system, creating and submitting transaction proposals to the endorsing peers on the network. The client node ensures secure and efficient interaction between end-users and the Hyperledger Fabric network, facilitating communication and transaction processing within the system.
- **Peer Node:** A peer node plays a central role in the network, maintaining the shared ledger and executing smart contracts (also referred to as chaincode). It is responsible for preserving a copy of the ledger, endorsing transactions, validating blocks, and updating it with new ones. The peer node's transaction endorsement and block validation functions ensure the network's security and consistency.
- **Orderer Node:** The orderer node is crucial for managing and maintaining the order of transactions. It groups endorsed transactions into blocks and distributes the newly created blocks to the peer nodes consistently and sequentially. The orderer node contributes to the network's overall scalability and performance, providing an efficient and reliable mechanism for handling large volumes of transactions.

Membership Service Provider (MSP): The MSP manages users' identities transacting on the Hyperledger Fabric network using digital certificates. Users sign transactions with these

certificates and submit them to the Fabric blockchain. The digital certificate provided to the user includes authentication to enter the system and a specific set of access rights.

Certificate Authority (CA): The CA is a critical component responsible for managing the network's digital certificates, which are used to verify the identity of network entities. The CA issues, manages, and revokes digital certificates, ensuring only authorized entities can participate in the network.

Chaincode: Chaincode, developed in Go, Node.js, or Java, is a program that implements a specific interface. It manages the state of the distributed ledger using transactions submitted by applications. Chaincodes encapsulate the mutually agreed-upon business logic among various network partners and are often called "smart contracts".

Channel: A channel is a private "subnet" communication between two or more specified network participants for performing private and confidential transactions. Members, anchor peers, the shared ledger, chaincode application(s), and the ordering service node(s) comprise a channel. Each transaction on the network is completed on a channel, for which each participant must be authenticated and granted permission to transact.

4.2.3. Creating the Hyperledger Fabric Network

Figure 4.3 illustrates the initial stage in establishing a Hyperledger Fabric network, which involves configuring an orderer peer. The orderer, part of the ordering service, adheres to the network configuration policy embedded in the blockchain network. Org1-CA, the Fabric certificate authority for Org1, generates certificates that enable Org1 to interact and participate within the network. These certificates fulfil two functions: they allow the organization's client applications to validate transaction proposals and enable endorsers to provide digital signatures corresponding to transaction outcomes. Organizations may either employ Fabric-CA, the default certificate authority supplied by Hyperledger Fabric or opt for an alternative certificate authority provider.

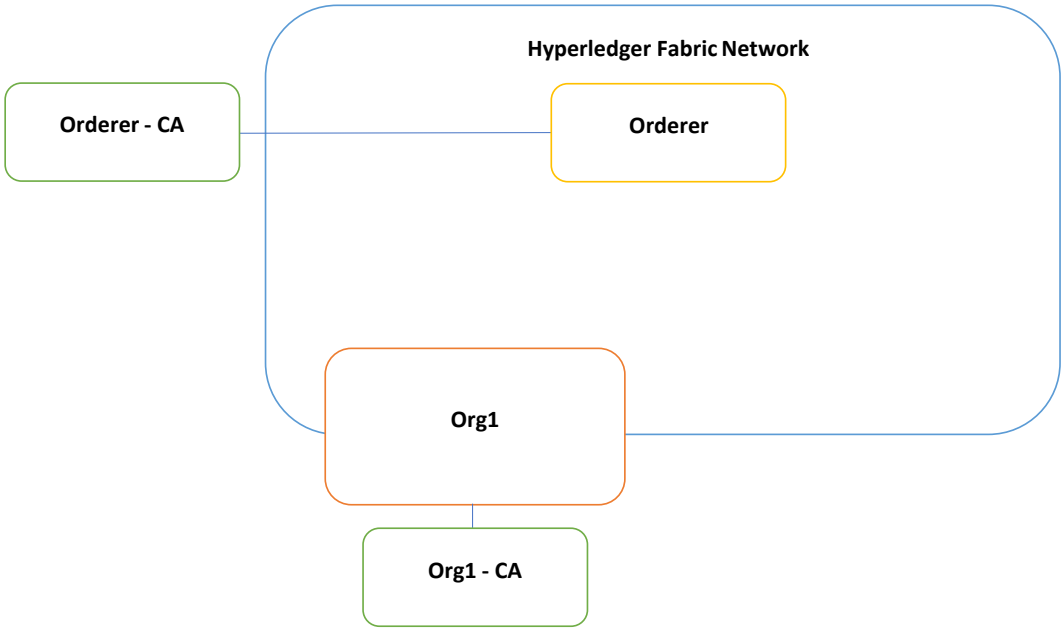

Figure 4.3: Creating the Hyperledger Fabric network.

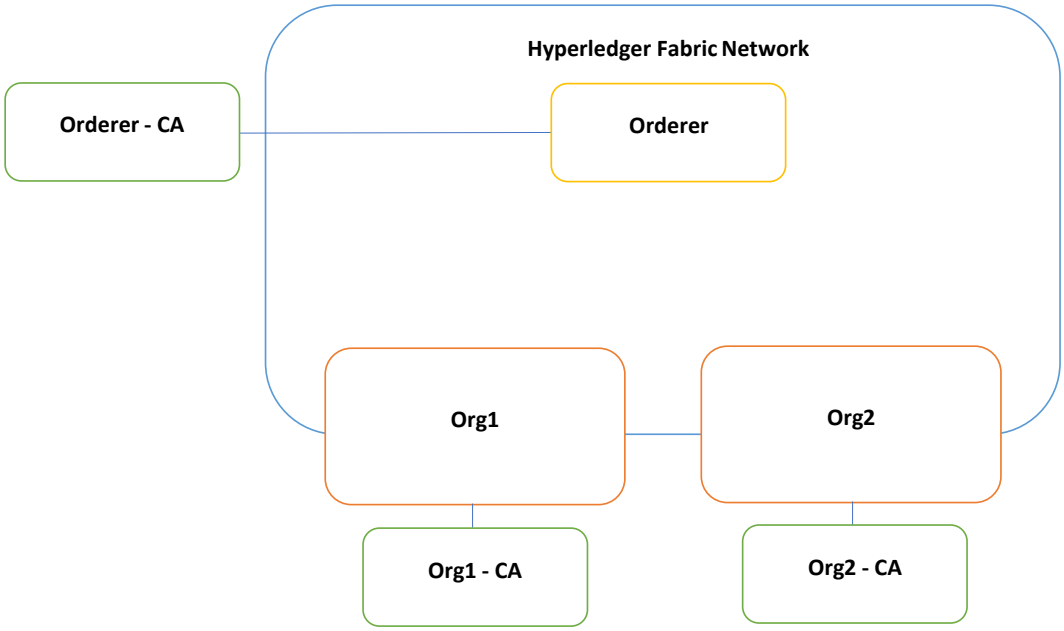

Figure 4.4: Adding organizations as administrators.

Designating an administrator is necessary for the Hyperledger Fabric blockchain framework, as it is a permissioned blockchain. As depicted in Figure 4.4, Org1 is established as an organization with administrative rights, authorized to add Org2 as an administrator, with both organizations complying with the same network configuration policies. To grant users from Org2 access to the network, Org2-CA must be incorporated into Org2. Both organizations can be created at the onset of the network architecture.

Before participating in a channel, organizations must first join a consortium. In this scenario, Org1 and Org2 form a new consortium and reach a consensus on the network's governing policies. Each transaction necessitates the creation of a consortium involving multiple parties. Subsequently, a channel must be created and configured to facilitate consortium organizations in sharing network infrastructure and engaging in private communication, as demonstrated in Figure 4.5.

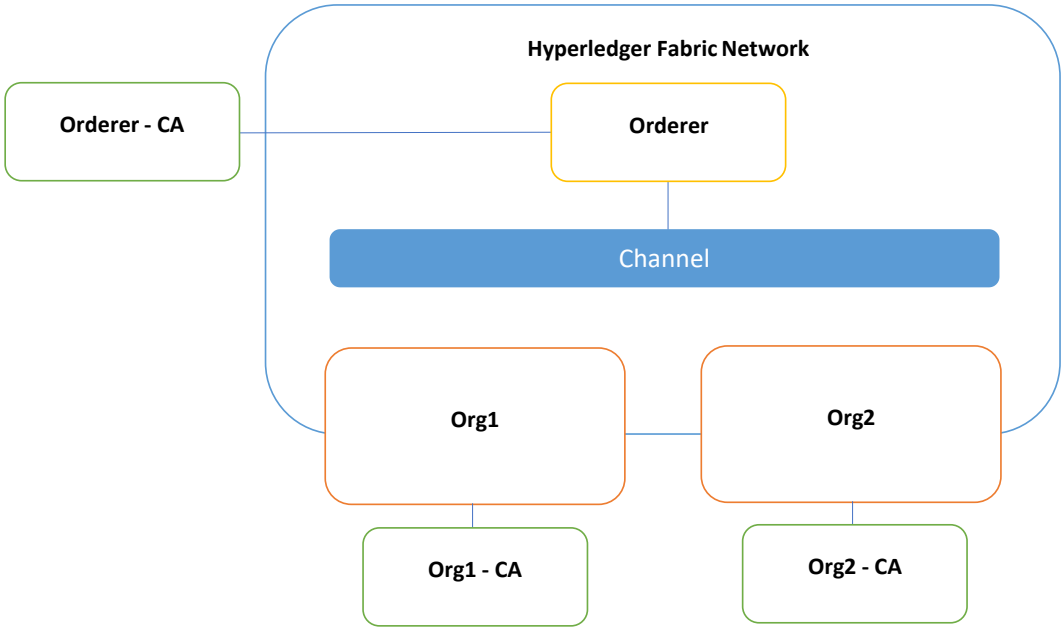

Figure 4.5: Creating a channel.

Figure 4.6 shows that when peers connect to a channel, multiple peers from different organizations can participate. Peer1, belonging to Org1, partakes in the channel and maintains a ledger, with a copy of the ledger also present on the channel. Similarly, Peer2, affiliated with

Org2, joins the channel and accesses the identical ledger copy. While the ledger is logically stored on the channel, it is physically situated on the respective peer nodes.

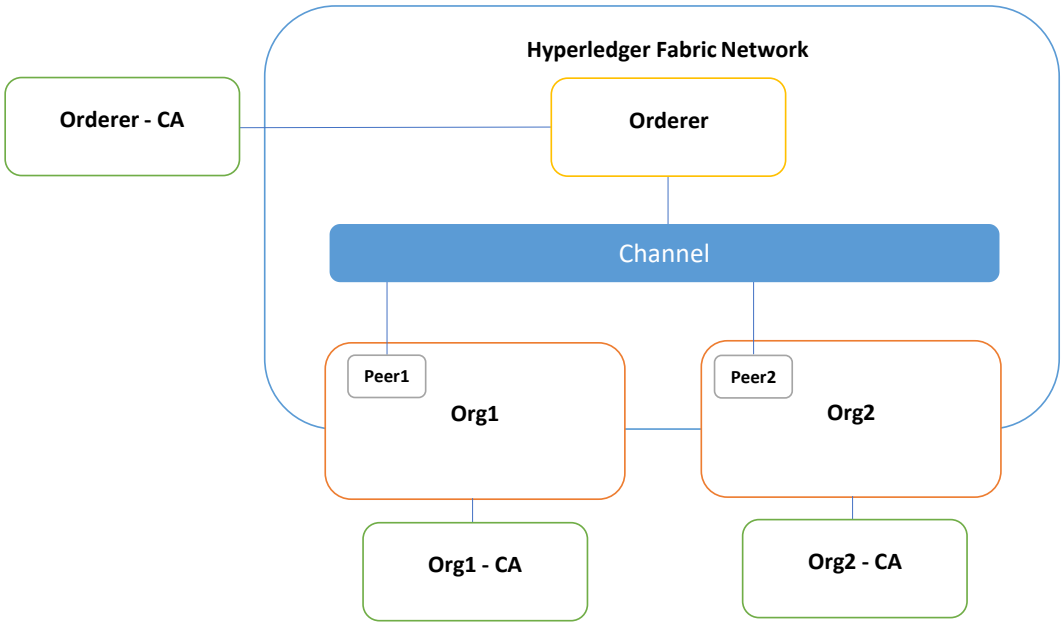

Figure 4.6: Defining peers and joining them to the channel.

Figure 4.7 depicts the procedure for deploying client applications and smart contracts (chaincodes) within the Hyperledger Fabric network. An external client application generates transaction proposals with the peer's ledger. The smart contract on Peer1 is an intermediary between the client application and the ledger. The client application cannot directly access the ledger and must first invoke its smart contract. The smart contract must be installed and initiated on all network nodes and added to the channel, ensuring other components know its existence.

Figure 4.8 presents a simplified network comprising two organizations connected to the fabric network via client applications. Each organization possesses a peer node linked to a single ordering service on a channel, resulting in a single logical ledger within the network. The network can be expanded by incorporating additional organizations and peers into new consortiums and configuring further channels. Each channel has unique rules, and multiple ordering services can manage various channels within the same network. Consequently, the number of smart contracts and distributed ledgers increases.

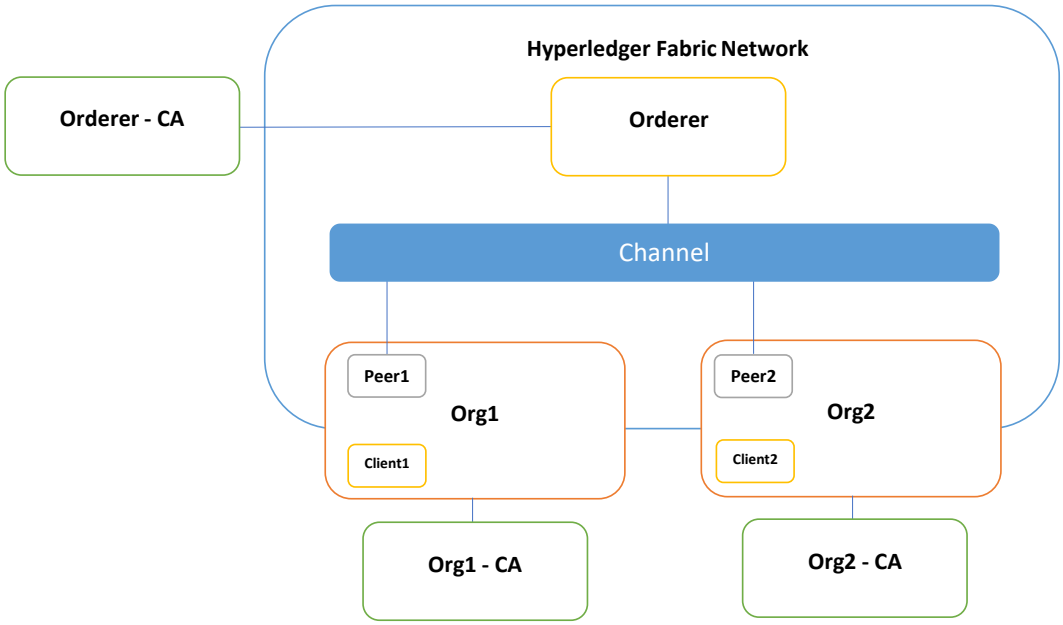

Figure 4.7: Adding client applications and chaincode.

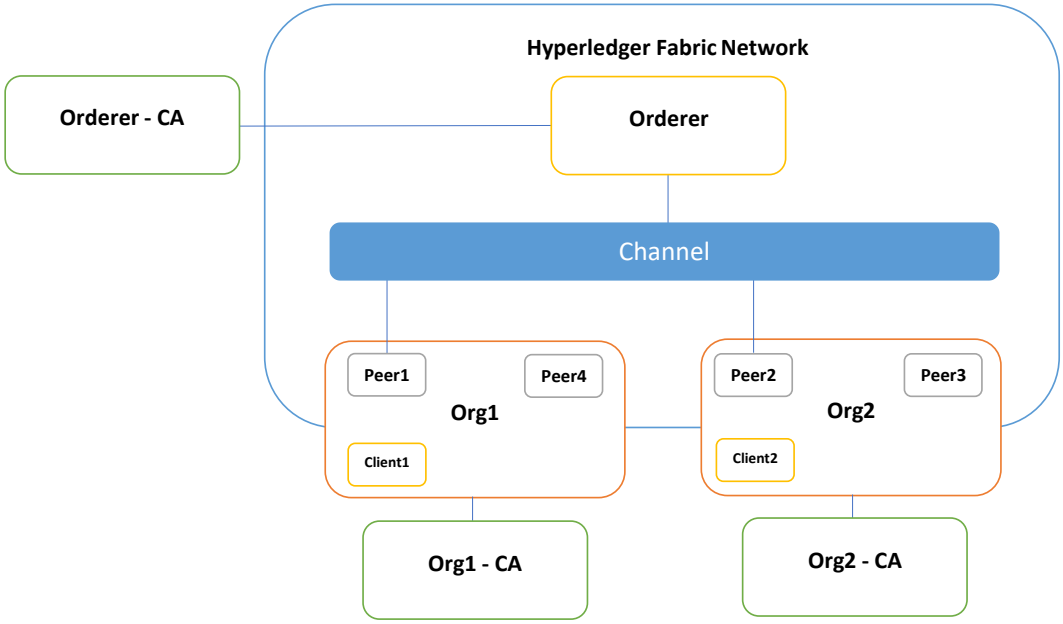

Figure 4.8: Complete Hyperledger Fabric architecture with two organizations.

4.2.4. The Transaction Flow in Hyperledger Fabric

This section explores the transaction flow in Hyperledger Fabric, illustrating the steps involved in the execution, ordering, and validation phases.

- **Execution phase:** The execution phase is crucial for processing transactions on one or more endorsing peers within the network. During this phase, chaincodes are created and executed. The endorsing peers simulate transactions, generate results that include proposed changes to the ledger state, and produce read-write sets. This phase ensures the accuracy and consistency of transactions proposed by client applications. By processing transactions on the endorsing peers before updating the ledger, Hyperledger Fabric guarantees that transactions are valid and follow the rules specified within the smart contract. Any discrepancies, inconsistencies, or violations of the smart contract logic are detected and flagged, allowing only valid transactions to proceed to the ordering and validation phases. This method significantly enhances the security and reliability of the blockchain network, ensuring data integrity and consistency across all participating nodes.
- **Ordering phase:** In the ordering phase, transactions are grouped and sequenced before being submitted to the ordering service, a vital part of the consensus process. The ordering service maintains the integrity and consistency of the distributed ledger by ensuring all transactions are ordered correctly and preventing double-spending. Endorsed transactions are sent by the client application to the ordering service, which groups them into blocks and orders them chronologically. The consensus algorithm, such as the Raft or Kafka algorithm, is used by the ordering service to agree on the order of transactions among the participating orderer nodes. After transactions are ordered and grouped into blocks, the blocks are disseminated to all peers within the channel, ensuring every peer in the network receives an identical copy of the block and preserving the consistency and synchronicity of the shared ledger.
- **Validation phase:** The validation phase is the final stage in the consensus process, during which peers in the network verify the correctness and integrity of transactions within the received blocks. Each peer within the channel performs a series of checks to validate transactions, including verifying endorsers' digital signatures, ensuring transactions comply with the endorsement policy, and confirming transactions do not conflict with the current state of the ledger. After completing validation checks, the

peer updates its local copy of the ledger with the new block if all transactions are valid. If any transaction is invalid, the peer marks it as such and does not update the corresponding state in the ledger. This approach allows the network to tolerate potentially invalid or malicious transactions without compromising the integrity of the overall ledger.

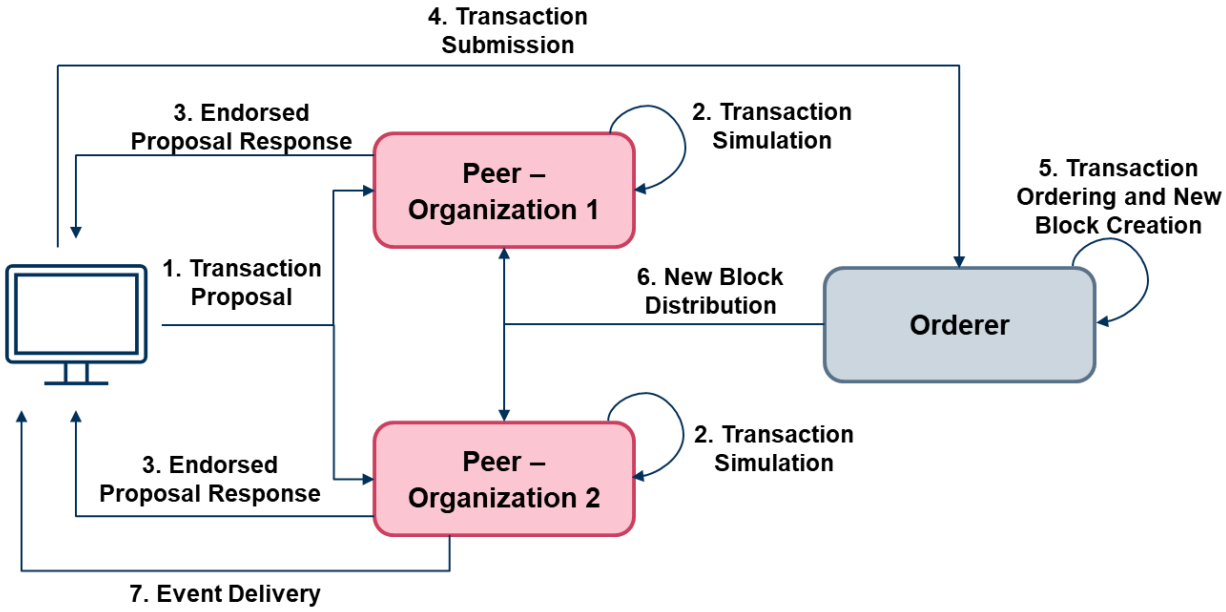

Figure 4.9: Hyperledger Fabric transaction workflow.

Figure 4.9 demonstrates the Hyperledger Fabric transaction invocation process:

1. The client submits a transaction proposal, signed with the user's certificate, to a preselected group of endorsing peers on a designated channel.
2. The endorsing peers authenticate the user's identity and permission using the proposal payload. Upon successful verification, the endorsing peer simulates the transaction, produces a response containing the read-write set, and endorses it using its certificate.
3. The client collects and examines the proposal responses from the endorsing peers.
4. The client forwards the transaction and the endorsed proposal responses to the orderer.
5. The orderer arranges the incoming transactions, generates a new block containing the organized transactions, and signs it using its certificate.
6. The orderer distributes the block to all peers (endorsing and committing) on the designated channel. Each peer verifies that the relevant endorsing peers have signed every transaction in the received block and that sufficient endorsements exist. The peer

then conducts a multi-version concurrency control (MVCC) check on each transaction in the received block and compares each transaction's readset to the world state in its ledger. If a transaction passes the verification process, the world state of each peer is updated. Invalid transactions do not affect the world state. Despite incorrect transactions, every peer appends the received block to its local blockchain.

7. EventHub delivers any subscribed events to the client as necessary.

In conclusion, the Hyperledger Fabric transaction flow is a carefully designed process that ensures secure, consistent, and efficient management of transactions within the network. By implementing a three-phase approach—execution, ordering, and validation—Hyperledger Fabric addresses potential risks and vulnerabilities, promoting high data integrity and reliability. The execution phase allows for detecting and flagging any inconsistencies or violations of the smart contract logic, while the ordering phase ensures the correct sequencing and distribution of transactions. Finally, the validation phase verifies the correctness and integrity of transactions, updating the ledger only when transactions are valid. This comprehensive process establishes a robust and trustworthy blockchain network, supporting the development of secure and efficient decentralized applications.

4.3. Proposed Solution for Post-Quantum Hyperledger Fabric Integration with IoT

4.3.1. System Requirements

Hyperledger Fabric, a large-scale distributed system comprising numerous nodes and clients, necessitates a post-quantum cryptography solution that ensures backward compatibility while allowing for a gradual and seamless transition. Organizations using Fabric should be able to continue leveraging their existing blockchains without restarting from scratch when adopting post-quantum cryptographic techniques. This approach enables the coexistence of classically encrypted client applications with post-quantum Hyperledger Fabric, facilitating a progressive shift from classical to post-quantum cryptography without synchronized downtime.

Furthermore, as NIST finalizes the list of post-quantum signature schemes, the proposed solution must exhibit adaptability and flexibility. It should be designed to be compatible with all candidate algorithms and remain algorithm-agnostic, allowing organizations to select the most appropriate approach without significant configuration challenges. By ensuring compatibility, adaptability, and ease of configuration, the proposed solution will help organizations transition to post-quantum cryptographic techniques efficiently and securely while maintaining the overall integrity and functionality of their existing Hyperledger Fabric systems.

4.3.2. Identity Proposal

Hyperledger Fabric transmits public key identities in the form of X.509 certificates. X.509 specifies public key certificates, which authenticate entities using signatures from publicly trustworthy authorities. To implement new post-quantum signatures in X.509, the X.509 algorithms would need to be modified. The recommendation in [151] is followed to generate post-quantum X.509 certificates, incorporating three non-essential Extensions:

- Alt − Signature − Algorithm: This term refers to the post-quantum algorithm for signing the certificate's key material. By employing this algorithm, the certificate offers enhanced security against potential quantum attacks.
- Subject − Alt − Public − Key − Info: This extension contains the post-quantum public key associated with the certificate. Depending on the specific use case, this field may be left empty or null if not required.
- Alt − Signature − Value: This field represents the post-quantum public key of the certificate, which has been signed using the post-quantum key of the issuing certificate authority (CA). This additional layer of security helps ensure the certificate's robustness in the face of potential threats posed by quantum computing.

Incorporating these elements into the certificate structure makes it possible to support both classical and post-quantum cryptographic schemes within the same certificate. This flexibility allows for a gradual transition to post-quantum cryptography as the technology evolves, ensuring the continued security and integrity of digital communication in the era of quantum computing.

4.3.3. Overall System Architecture

The overall design of the system comprises two distinct components.

- Post-Quantum Hyperledger Fabric
- IoT

4.3.4. Post-Quantum Hyperledger Fabric System Setup

The Hyperledger Fabric Blockchain infrastructure in this implementation consists of a client, an orderer, and two organizations, org1 and org2, each with one peer. Additionally, there are CA1 and CA2 Certificate Authorities for each organization. The peers are designated as Peer 0 and Peer 1. The domains org1 and org2 differentiate the peers belonging to each organization (refer to Figure 4.10).

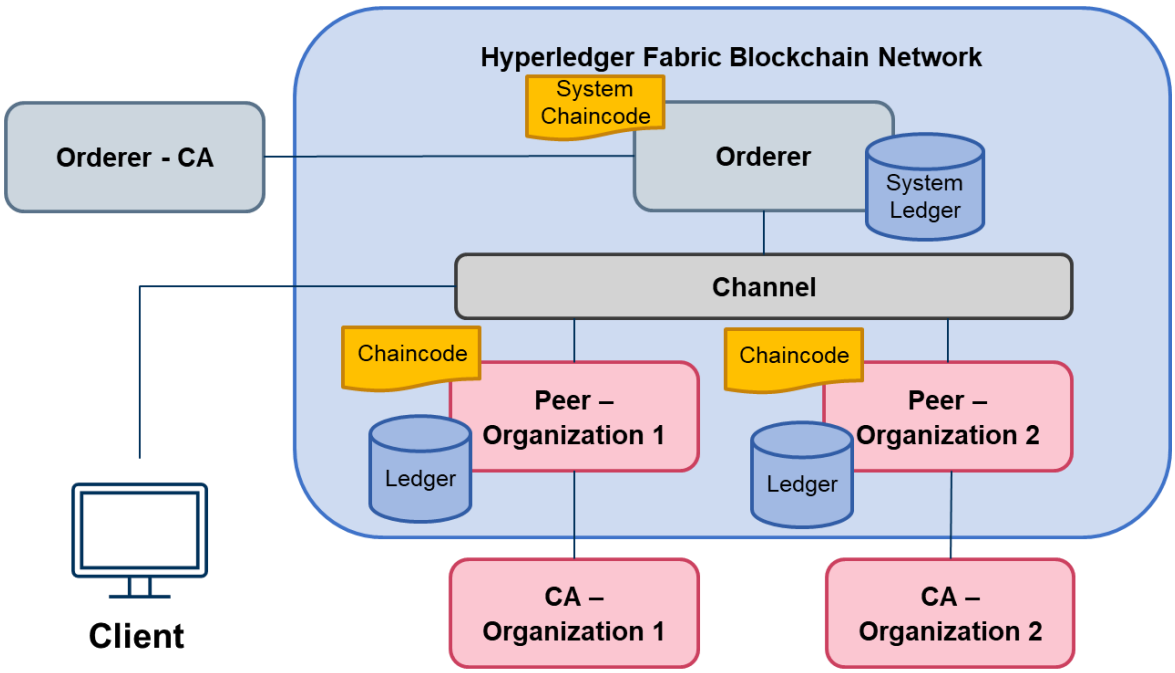

Figure 4.10: Hyperledger Fabric architecture with two organizations.

4.3.5. IoT System Setup

4.3.5.1. Hardware

The hardware components include the following:

- Raspberry Pi 4
- DHT22 temperature and humidity sensor
- Jump wires
- System running Linux (Ubuntu)

4.3.5.2. Data Collection and Communication

A Python library is employed to interpret the data gathered from the DHT22 sensor. It is essential to read the data from the sensor at a minimum interval of 2 seconds.

MQTT is an OASIS-standard Internet of Things communications protocol (IoT). It is designed as an extremely lightweight publish/subscribe message transport suited for linking remote devices with minimal network traffic and a small amount of code. MQTT is employed in numerous industries, including automotive, manufacturing, telecommunications, oil, and gas.

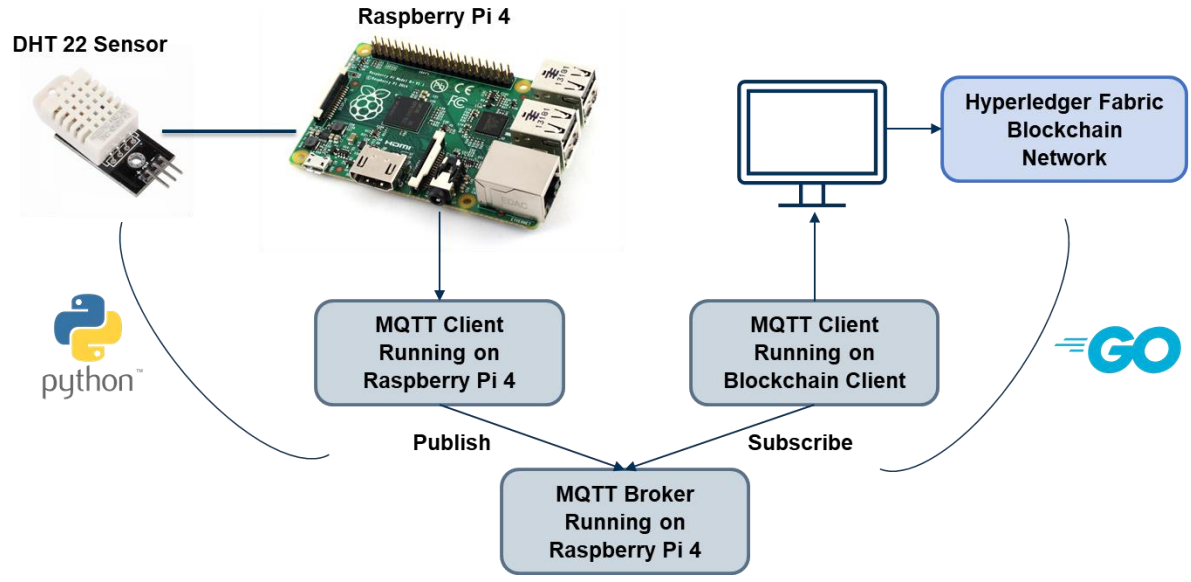

Figure 4.11: Hyperledger Fabric Blockchain on IoT System Setup.

The DHT 22 sensor is the core component of the IoT system. It is connected to the Raspberry Pi 4 through jump wires. MQTT broker software is installed and operating on the Raspberry Pi 4. In addition, a Linux system is utilized as the host for both the MQTT Client and the Hyperledger Fabric Blockchain Network.

4.4. Implementation and Configuration of Post-Quantum Hyperledger Fabric in IoT Environments

Hyperledger Fabric features a modular cryptographic provider, which replaces existing digital signature algorithms with alternative ones. IBM researchers highlighted this adaptability in a 2018 guide discussing quantum cryptography in the context of Hyperledger Fabric [152]. However, implementing these changes is not as straightforward as it may seem. This section explores the core structure of the implementation process, offering insights into how the cryptographic provider functions and how it can be customized to meet the evolving requirements of quantum-secure systems. By understanding this framework, it becomes possible to adapt and enhance the cryptographic capabilities of Hyperledger Fabric to maintain robust security in a world where quantum computing poses new challenges.

4.4.1. Core Structure

This implementation is founded on Hyperledger Fabric 2.4, the most recent version available during the developmental phase. LibOQS 0.7.2 [153] was utilized to incorporate post-quantum cryptographic signature algorithms. As liboqs is written in C and Hyperledger Fabric in Go, a CGO wrapper for liboqs was developed to bridge the gap between the two languages.

The methodology employs three struct types: SecretKey, PublicKey, and OQSSignInfo, each linked to the OQS algorithm name. The Go representation of the liboqs library and Sig objects are concurrently loaded, maintaining pointers to the liboqs C functions. With a focus on signing and verifying messages using post-quantum cryptography, the wrapper includes KeyPair, Sign, and Verify functions.

Enabling post-quantum signatures requires updating three essential parts of the Hyperledger Fabric source.

1. **Cryptogen:** This command-line utility within Hyperledger Fabric facilitates the generation of the necessary cryptographic material for running a Fabric network. While not officially supported by Hyperledger Fabric, Cryptogen offers users a helpful starting point, particularly during network setup and testing. Based on provided configuration files, this utility produces cryptographic artifacts, such as public and private keys, X.509 certificates, and other vital components for each organization and peer within the network. These artifacts are essential for secure communication, identity management, and trust-building among various entities in the blockchain network. Cryptogen simplifies the creation of cryptographic material, aiding developers and organizations in establishing and configuring their Fabric network. Organizations using Fabric can generate this material using alternative methods tailored to their specific security and operational requirements.
2. **Blockchain Cryptographic Service Provider (BCCSP)**: BCCSP is a critical component within Hyperledger Fabric, offering a standardized and modular interface for cryptographic operations. BCCSP guarantees that Fabric's core functionality remains separate from underlying cryptographic algorithms and implementations. By providing a consistent interface, BCCSP enables developers to utilize various cryptographic primitives while maintaining the blockchain network's security and integrity. The BCCSP module is designed for high flexibility and extensibility, allowing users to integrate cryptographic libraries, algorithms, and key management systems according to their needs. This adaptability improves a Fabric network's security by incorporating advanced cryptographic techniques or replacing outdated algorithms with more secure alternatives. The primary modification involved adding a new key type and interface, encompassing KeyImport, KeyPair, Sign, Verify, and other functions. As a result, the Signer module shared by all BCCSP keys is also updated.
3. **MSP:** MSP is a vital component in Hyperledger Fabric that manages network participants' identities, authentication, and authorization. It is responsible for validating users' and nodes' credentials and permissions within the blockchain network, ensuring only authorized entities can access and interact with the system. MSP plays a critical role in maintaining the security and integrity of the Hyperledger Fabric network by enforcing policies and rules governing participants' actions. By doing so, MSP helps prevent unauthorized access, tampering, and malicious activities within the network. Although the signature functionality is fully incorporated into a shared Signer within

the BCCSP module, the corresponding verification methods have not been integrated similarly. Therefore, MSP modifications are necessary to accommodate these verification methods.

4.4.2. Building Network

The network configuration necessitated installing Hyperledger Fabric prerequisites, such as Git, cURL, NPM v8.17.0, Node.js v16.16.0, and the Go1.18 programming language. Subsequently, the /config and /bin directories were employed to install the platform-specific configuration and binary files for Hyperledger Fabric. Docker was utilized to create containers for each component, including peers, orderers, CAs, and the command-line interface (CLI). Instead of using entire physical or virtual machines, this container technology offers more lightweight implementations and enables faster testing of configuration files. All configuration information is stored in a docker-compose file, and the CLI container is employed to access the network's peers and orderers. The configuration encompasses mountable properties, volumes, and the locations of keys, certificates, and MSP settings.

Table 4.1: Environment configuration.

Component	**Description**
CPU	Inter Core i7
Memory	16 GB
Operating System	Windows 11
Linux Distribution	WSL2 Debian
Hyperledger Fabric	v2.4.0
NPM	v8.17.0
Node	v16.16.0

Docker Desktop	v4.16.3

4.4.2.1. Network Artifact Creation

Building a Hyperledger Fabric network entails generating and configuring several critical artifacts. These artifacts are instrumental in establishing the network's foundation and facilitating efficient communication and secure transactions among participating organizations. The steps involved in creating network artifacts include the following:

1. **Cryptographic material generation:** Generating cryptographic materials for each participating organization is foundational. Utilize tools like cryptogen or a Certificate Authority to create the necessary cryptographic components for secure communication and identification within the network.
2. **Genesis block creation:** The genesis block is the first block for the ordering service's system channel. Generate the genesis block using the configtxgen tool and a predefined configuration file (configtx.yaml) containing essential network configuration information.
3. **Channel configuration transaction generation:** A channel configuration transaction file is required to create and configure a new channel within the network. Generate this file using the configtxgen tool based on the configtx.yaml file, ensuring the channel's settings align with the network's requirements.
4. **Chaincode packaging and installation:** Develop chaincodes that encapsulate the agreed-upon business logic for the network's application. Package the chaincode files and install them on the appropriate peer nodes, allowing the execution of smart contracts and ledger updates.
5. **Connection profile creation:** A connection profile configuration file is necessary to facilitate interaction between client applications and the Hyperledger Fabric network. This file should include details about channels, organizations, orderers, peers, and certificate authorities, enabling seamless connectivity and secure communication within the network.
6. **Anchor peer configuration:** Assigning anchor peers for each organization helps maintain efficient and reliable communication between network organizations. Update the configtx.yaml file to include anchor peer configurations for each organization, ensuring robust inter-organizational communication.

Upon completing the creation and configuration of network artifacts, the next step involves deploying and initiating the Hyperledger Fabric network components, such as orderers, peers, and certificate authorities. With these components in place, instantiate chaincodes on the channel and begin developing and testing client applications. Creating network artifacts is essential for establishing a secure and functional Hyperledger Fabric network, laying the groundwork for successful implementation and real-world applications.

4.4.3. Docker Container Configuration

When setting up a Hyperledger Fabric network, various network entities, such as peers, orderers, and CLIs, must be configured. This necessitates establishing the appropriate configuration for multiple Docker containers, each corresponding to a distinct network entity. The configurations for these Docker containers are stored in a Docker Compose file, outlining essential settings for each container, including network connections, volume mounts, and environment variables. Utilizing Docker Compose streamlines the deployment and management of diverse network entities in a Hyperledger Fabric environment, ensuring a more efficient and organized approach to constructing and maintaining blockchain networks.

4.4.4. Incorporating External IoT Data into Hyperledger Fabric

Monitoring environmental conditions, precisely temperature and humidity, is achieved using a DHT22 sensor connected to a Raspberry Pi 4 device. The Adafruit Python library facilitates the collection of sensor data and its transmission to the Hyperledger Fabric node via the MQTT Protocol. The data payload, which consists of timestamp, temperature, and humidity readings, is published to an MQTT broker hosted on a Raspberry Pi under the topic “IoTData”.

Subsequently, the Hyperledger Fabric node subscribes to the topic and obtains the data, which is processed by a shell script to create a transaction proposal. For a transaction to be considered legitimate, it must receive signatures from at least one peer representing each participating organization. The payload data is appended to the ledger once signatures are verified, and the transaction is deemed valid.

4.4.5. Smart Contract (Chaincode)

The primary function of the chaincode, createIoTData, is to incorporate incoming IoT data into the distributed ledger. This function accepts a single string parameter representing IoT data.

Within the createIoTData function, the stub.PutState method generates a state entry in the ledger. This method requires two parameters: a key and a value. In this context, the key corresponds to IoT data, while the value is a JSON string encapsulating both temperature and humidity data points.

Using the createIoTData function, IoT-generated data can be effectively stored and managed within the Hyperledger Fabric blockchain. This enables the seamless integration of real-world IoT devices into the ledger, fostering secure and efficient data storage and retrieval for various applications within the IoT domain.

4.4.6. Channel Setup

In Hyperledger Fabric, creating a new channel necessitates using the client SDK to interact with the configuration system chaincode. The client refers to properties such as anchor peers and member organizations to initiate channel creation during this process. As a result, a genesis block for the channel ledger is generated, encompassing crucial details about channel policies, member organizations, and anchor peers.

When adding a new member to an existing channel, the genesis block or the latest reconfiguration block is disseminated to the new member. This ensures that all members possess the most current information concerning the channel and its associated policies. By maintaining an up-to-date ledger and synchronizing information across all members, the Hyperledger Fabric network ensures consistency, transparency, and security within the channel, fostering efficient and reliable communication among the participating organizations.

4.4.7. Deploying Chaincode to Channel

The chaincode lifecycle in Hyperledger Fabric is a systematic process governing the deployment, management, and upgrade of smart contracts, known as chaincodes, within the network. The chaincode lifecycle consists of several stages, including:

- **Chaincode Development:** Chaincode is created using a supported programming language (such as Go, JavaScript, or Java). The chaincode contains the business logic and rules for executing transactions on the network.
- **Packaging:** The chaincode is then packaged into a deployable format, typically a tarball containing the chaincode source code, metadata, and dependencies. This packaging ensures the chaincode can be easily distributed and installed on the required peer nodes.
- **Installation:** The packaged chaincode is installed on the endorsing peer nodes of the participating organizations. Each organization installs the chaincode on its respective peer nodes independently. This step makes the chaincode available for instantiation.
- **Approval:** After chaincode installation, the organizations that are part of the channel must approve the chaincode definition, which includes the chaincode's version, endorsement policy, and other configuration parameters. The approval process is subject to the channel's governance policies, which may require a majority or a specific set of organizations to approve the chaincode definition.
- **Instantiation (or Commitment)**: Once the necessary approvals are obtained, the channel's chaincode is instantiated (or committed). This step initializes the chaincode on the channel ledger, setting the initial state and allowing the chaincode to be invoked for transactions.
- **Invocation:** With the chaincode instantiated, client applications can now invoke it to execute transactions. The endorsing peers process these transactions according to the chaincode's endorsement policy, and the results are returned to the client application for submission to the ordering service.
- **Validation and Commitment:** After the ordering service orders the transactions, they are delivered to all peer nodes in the channel, where they are validated and, if valid, committed to the channel ledger. This updates the ledger's state based on the executed transactions.
- **Upgrades:** If the chaincode requires modifications or improvements, it can be upgraded by following a similar process to the initial deployment. The new version of the

chaincode must be packaged, installed, approved, and instantiated on the channel, replacing the previous version.

By adhering to this lifecycle, Hyperledger Fabric ensures a controlled and transparent management process for chaincodes, allowing organizations to collaborate effectively and securely within a permissioned blockchain network.

4.5. Evaluation, Results, and Performance Analysis

This section presents the evaluation, results, and performance analysis of the Post-Quantum Hyperledger Fabric implementation in IoT scenarios. The assessment aims to understand the performance characteristics and limitations of the proposed solution when deployed in real-world IoT environments. The evaluation focuses on two main aspects: the performance of post-quantum cryptographic algorithms and the overall performance of Hyperledger Fabric in IoT scenarios.

4.5.1. Post-Quantum Hyperledger Fabric Assessment

Evaluating the proposed post-quantum Hyperledger Fabric implementation is critical in understanding its performance and viability in real-world scenarios. This section aims to provide an in-depth assessment of the post-quantum Hyperledger Fabric implementation by comparing its performance metrics with those of the classical cryptographic configuration.

The implementation is evaluated on a network comprising a client, an orderer, and two peers, with each component operating on a separate Docker container. The assessment is conducted on a system with an Intel(R) Core(TM) i7-1165G7 @ 2.80GHz processor, 16GB RAM, and an SSD.

Table 4.2: Tested algorithms, sorted by certificate size in bytes.

Algorithm	Key Size (bytes)	Cert Size (bytes)
ECDSA	241	810

Falcon – 512	3,052	2,988
Falcon – 1024	5,652	5,035
Dilithium 2	5,498	5,254
Dilithium 3	6,843	6,530
Dilithium 5	10,198	10,617
SPHINCS+-SHA256-128s-simple	225	11,539
SPHINCS+-SHA256-128f-simple	225	24,044

In the standard cryptographic configuration, 128-bit classical security is achieved, and transactions are exclusively signed using ECDSA, as defined over the NIST curve P-256 (according to FIPS 186-3). This setup is compared to a test run where nodes are configured to sign and verify transactions using post-quantum schemes from the remaining NIST round three finalists and several alternates, as implemented in liboqs 0.7.2.

Table 4.2 presents the tested algorithms, sorted by certificate size in bytes. It includes information about the algorithm, key, and certificate size. Table 4.3 shows the execution times of post-quantum signature cryptosystems, including each algorithm's time taken for key generation, signing, and verification.

An analysis of the data in Table 4.2 reveals that the key and certificate sizes vary considerably among the algorithms. ECDSA has the smallest key and certificate sizes, while SPHINCS+-SHA256-128f-simple has the largest certificate size. Smaller key and certificate sizes can lead to reduced communication overhead and storage requirements, making algorithms with smaller sizes potentially more efficient in a practical implementation.

Table 4.3: Execution times of post-quantum signature cryptosystems.

Algorithm	Keygen (ms)	Sign (ms)	Verify (ms)
ECDSA	0,028	0.095	0.234
Falcon - 512	8.68	0.286	0.082
Falcon - 1024	23.82	0.544	0.231
Dilithium 2	0.177	0.276	0.140
Dilithium 3	0.255	0.401	0.203
Dilithium 5	0.406	0.515	0.241
SPHINCS+-SHA256-128s-simple	32.61	265.11	0.136
SPHINCS+-SHA256-128f-simple	0.898	16.18	0.173

Regarding execution times displayed in Table 4.3, ECDSA has the shortest key generation, signing, and verification times. However, among the post-quantum algorithms, Falcon-512 shows the fastest signing time, while Dilithium 2 exhibits the quickest verification time. It is essential to consider these execution times when selecting a post-quantum cryptographic algorithm, as they can impact the overall performance of the Hyperledger Fabric network.

Figure 4.12 compares the average latency of different signature algorithms, including ECDSA, Falcon-512, Falcon-1024, Dilithium 2, Dilithium 3, and Dilithium 5, within the context of the Hyperledger Fabric platform. The latency is measured in milliseconds (ms) for write and read operations.

The graph shows that ECDSA has the lowest latency, with 2.5 ms for write operations and 0.26 ms for read operations. In contrast, Dilithium 5, one of the post-quantum signature algorithms, exhibits the highest latency, with 4.33 ms for write operations and 0.45 ms for read operations.

The Falcon signature algorithms have intermediate latency values. Falcon-512 has 3.06 ms for write operations and 0.32 ms for read operations, while Falcon-1024 shows 3.26 ms for write

operations and 0.34 ms for read operations. Among the Dilithium algorithms, Dilithium 3 has a latency of 3.87 ms for write and 0.4 ms for read operations, and Dilithium 2 has a latency of 3.46 ms for write and 0.36 ms for read operations.

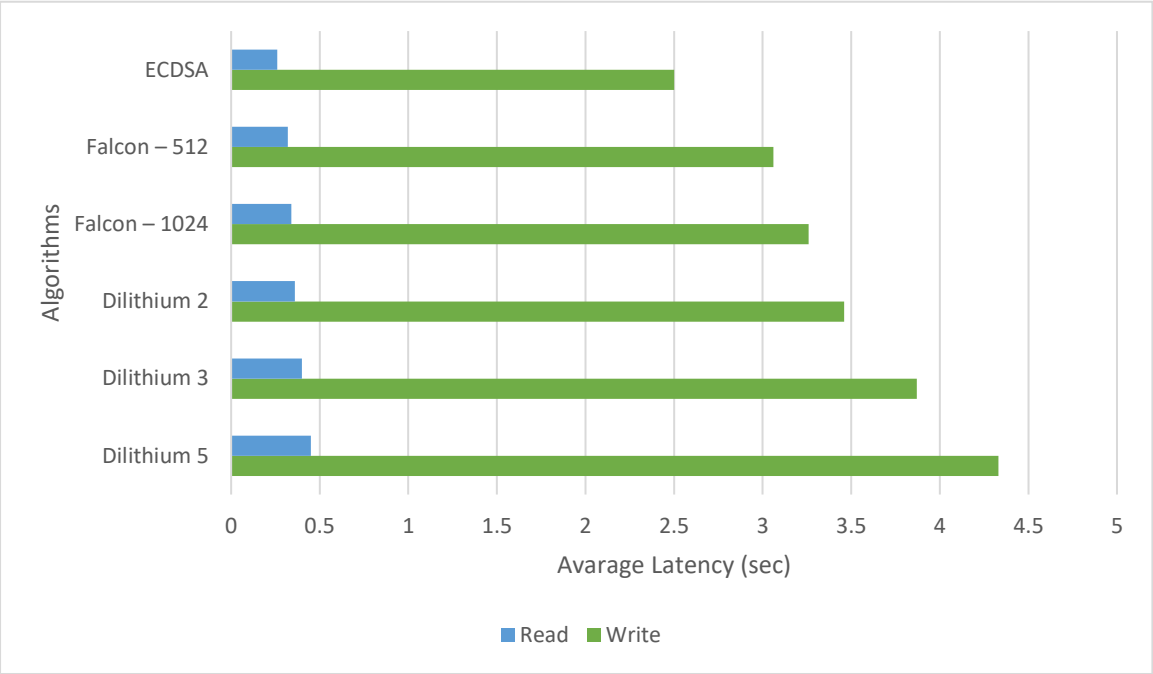

Figure 4.12: Comparison of average latency performance for post-quantum Hyperledger Fabric with different signature algorithms.

Figure 4.13 presents a comparative analysis of the throughput performance of Hyperledger Fabric, utilizing various signature algorithms.

The graph shows that ECDSA has the highest throughput for write and read operations, with 58 writes and 350 reads per second. This can be attributed to its widespread use and optimized implementations in current blockchain systems. Falcon-512, a post-quantum signature algorithm, demonstrates competitive performance with 49 writes and 300 reads per second. This indicates that post-quantum signature algorithms can effectively integrate into Hyperledger Fabric without severely compromising performance.

The graph also highlights the performance differences among the Dilithium variants, with Dilithium 2 achieving better throughput than Dilithium 3 and Dilithium 5. This is likely due to the reduced complexity of Dilithium 2, which results in faster operations.

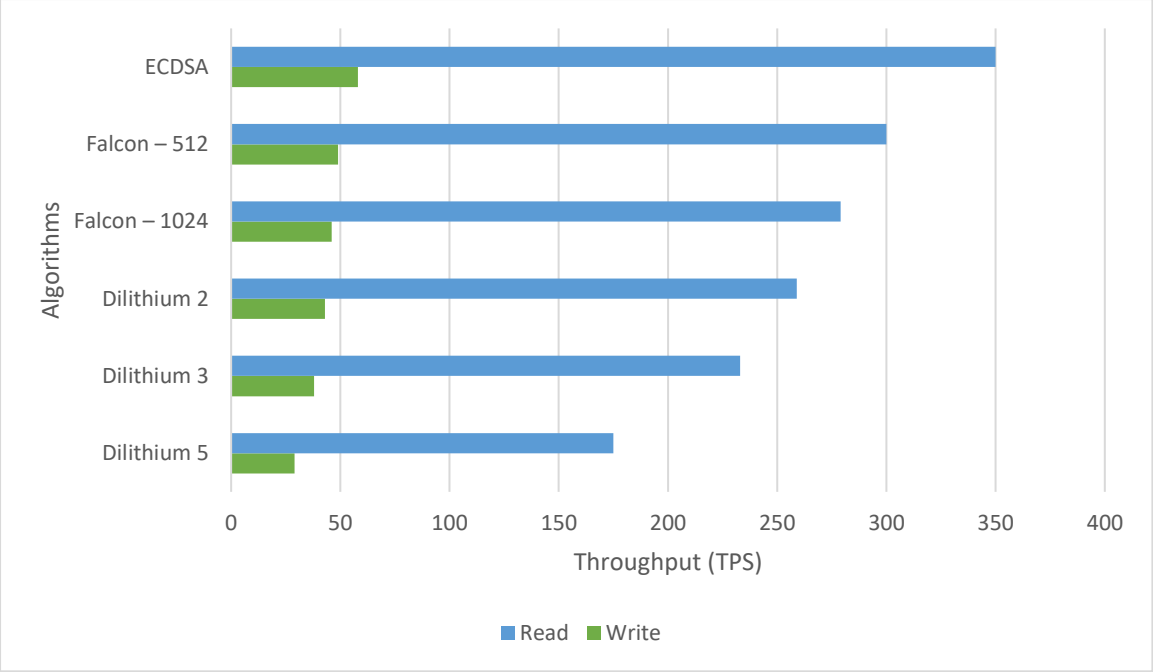

Figure 4.13: Comparison of throughput performance for post-quantum Hyperledger Fabric with different signature algorithms.

In summary, the graphs provide valuable insights into the average latency and throughput capabilities of Hyperledger Fabric when employing different signature algorithms, demonstrating that post-quantum signature algorithms, such as Falcon, can be incorporated into blockchain systems without significantly affecting performance. These results can help guide the choice of appropriate signature algorithms for Hyperledger Fabric implementations, particularly as the need for post-quantum security increases.

4.5.2. Hyperledger Fabric Performance in IoT Scenarios

Experimental tests were conducted using various software tools to evaluate the proposed system to obtain comprehensive results and demonstrate the framework's suitability. Several use case scenarios with distinct configurations were executed to highlight diverse performance indicators.

The performance metrics for the network in this analysis include throughput, defined as the number of successful transactions completed per second, and transaction latency, referring to the time interval between submitting a transaction and receiving a response. Pongnumkul et al.

[154] emphasize that throughput and latency are crucial metrics for understanding a blockchain's performance and limitations.

Hyperledger Caliper [155], a benchmarking tool developed by the Linux Foundation, was employed to test the blockchain network. Caliper provides a standardized framework for evaluating the performance of various blockchain platforms, including Hyperledger Fabric, Ethereum, and others. It enables users to define customized benchmarks, test scenarios, and load patterns to assess specific use cases and performance metrics of interest. Caliper consists of three primary layers: the benchmark, core, and adapters (see Figure 4.14).

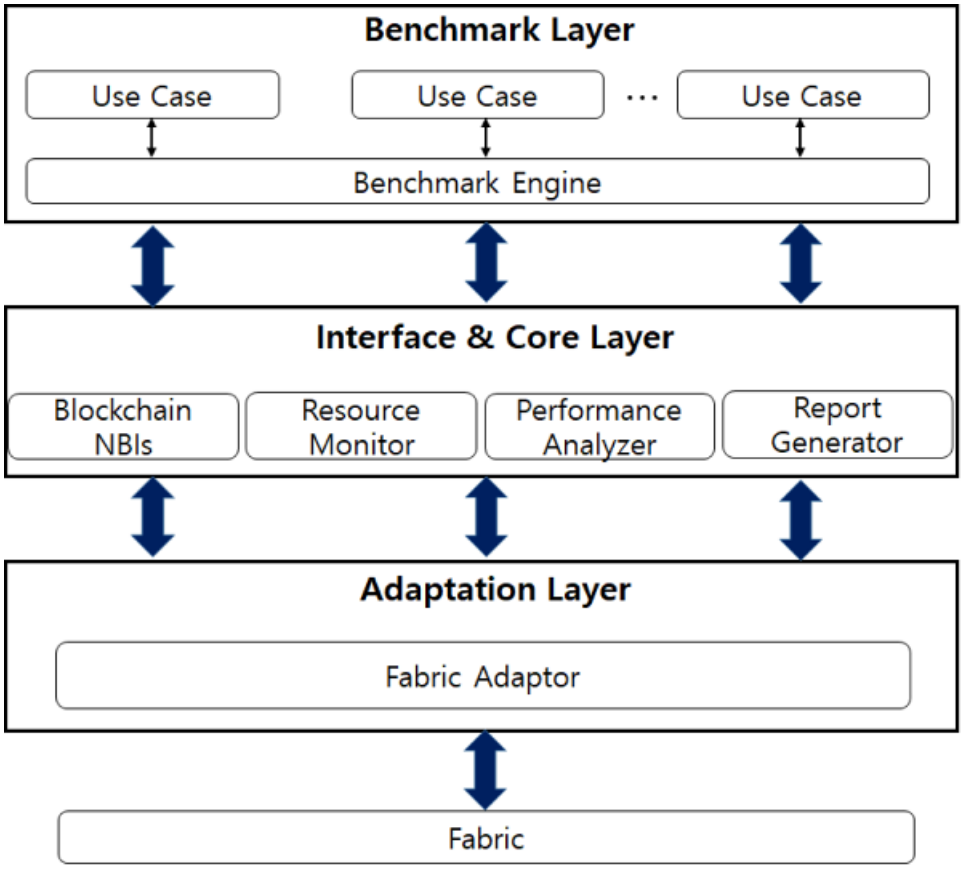

Figure 4.14: Hyperledger Caliper architecture [156].

1. **Benchmark Layer:** This layer defines the workload and benchmarking scenarios executed during performance testing. It comprises configuration files and user-defined smart contracts, which specify the test parameters, the targeted blockchain platform,

and the desired transaction rate. The Benchmark Layer supplies this information to the Core Layer to initiate benchmarking.

2. **Core Layer:** As the central coordinating component of Caliper, the Core Layer manages the benchmarking process and orchestrates the interactions between the other layers. It initializes the Adapters Layer, deploys the smart contracts, controls the rate at which transactions are generated, and collects performance metrics from the Workers Layer. The Core Layer processes these metrics and provides them to the Reporting Layer for further analysis and visualization.
3. **Adapters Layer:** This layer translates platform-specific API calls into a generalized format that the Caliper Core can understand. Providing platform-specific adapters enables Caliper to interact with various blockchain platforms, such as Hyperledger Fabric, Ethereum, or other supported platforms. During benchmarking, the adapters ensure seamless communication between Caliper and the targeted blockchain platform.

These layers work in conjunction to effectively assess the performance of the targeted blockchain platform, enabling users to compare different blockchain implementations, identify potential bottlenecks, and pinpoint areas for optimization.

Various transaction rates were employed during each testing cycle to evaluate the network's performance. The test consisted of six rounds, with transaction rates of 50, 100, 150, 200, 250, and 300 transactions per second (TPS) for each round. Each round contained a total of 1,000 transactions. The average transaction latency and throughput were computed and illustrated in graphical representations throughout the iterations. The testing procedure encompassed write and read transaction modes, comprehensively assessing the network's performance under different transaction rates and operation types. This approach helped identify potential bottlenecks and assess the network's capability to handle varying loads, ultimately contributing to a better understanding of the system's efficiency and scalability.

The first test case, as shown in Table 4.4, evaluates writing transactions in which a ledger must be updated with temperature and humidity values by calculating transaction latency and throughput.

Table 4.4: Testing Writing Transaction Mode.

Parameter	Configuration
Number of Rounds	6
Total transactions	1000
Transaction Rates	50, 100, 150, 200, 250, 300 TPS
Transaction Mode	Write Temperature

Figure 4.15 demonstrates the correlation between average latency, measured in seconds, and throughput in relation to transaction rates, explicitly focusing on the write transaction mode. This visual representation helps to analyse the performance impact of different transaction rates within the context of the writing operation mode.

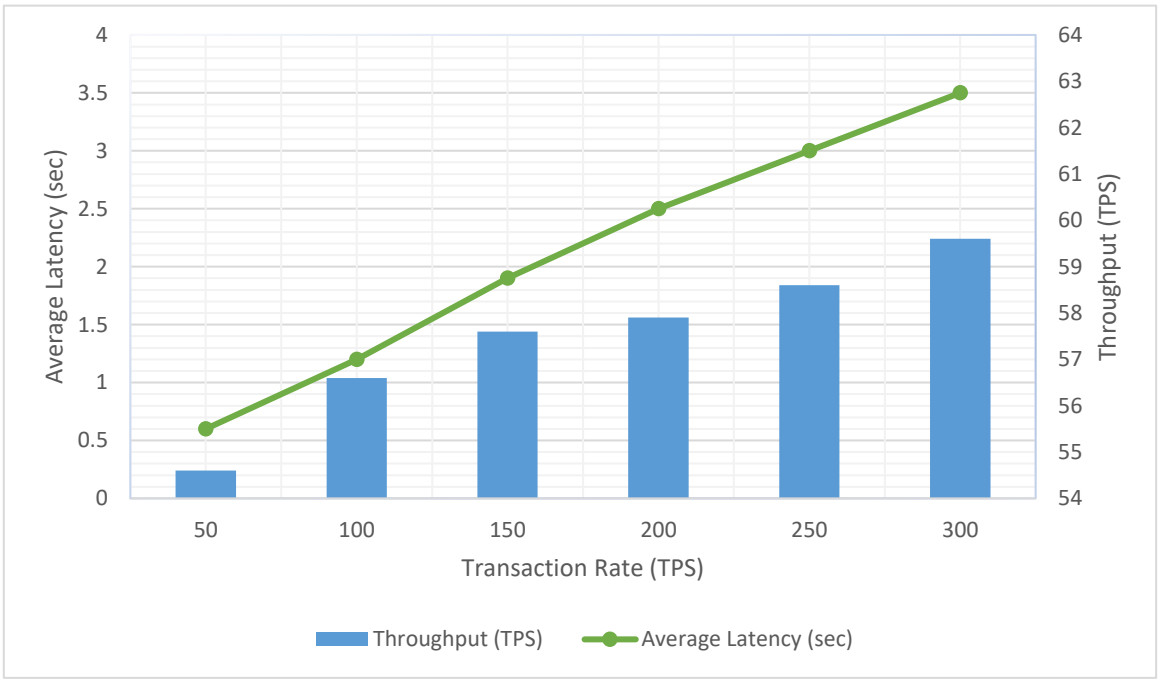

Figure 4.15: Evaluating the first scenario: writing transactions mode on the network latency and throughput.

The average latency gradually increases as the writing transaction rate increases each round. Notably, after reaching 200 TPS, the average throughput follows a nearly linear pattern. In

general, lower latency contributes to higher throughput. This observation indicates that the system can maintain high throughput even when subjected to increasing transaction rates.

The second test scenario assessed transaction latency and throughput to evaluate reading or querying transactions. In this situation, the requisite read workload was generated by all clients simultaneously sending their queries to a single node.

Table 4.5: Testing Reading Transaction Mode.

Parameter	Configuration
Number of Rounds	6
Total transactions	1000
Transaction Rates	50, 100, 150, 200, 250, 300 TPS
Transaction Mode	Read Temperature

Figure 4.16 displays the relationship between average latency, measured in seconds, and throughput concerning transaction rates for the read transaction mode. Based on the assessment results illustrated in the graph, the average latency experiences a minimal increase as transaction rates grow with each iteration. The average latency remains near zero, and the system can process up to 300 TPS smoothly. This suggests that the system's maximum capacity has not been reached and can handle even larger transaction rates. Furthermore, the rising transaction rates demonstrated minimal influence on the average throughput. Throughout all the rounds, the throughput remained relatively stable, consistently surpassing 100 TPS.

The evaluation of write and read transactions in the proposed system reveals that the average delay for write transactions is higher than for read transactions. This can be attributed to the additional processing steps required for write transactions, which involve verifying the transaction, executing the smart contract, and updating the ledger using the consensus

algorithm. In contrast, read transactions simply require the execution of the smart contract to access the data stored in the blockchain.

The system's performance results showcase its lightweight nature, as it minimizes network latency while maintaining high throughput. These findings highlight the efficiency and effectiveness of the proposed system in managing a diverse range of transactions within a blockchain network. The system's design and implementation successfully balance transaction processing demands, demonstrating its suitability for various use cases in the blockchain domain.

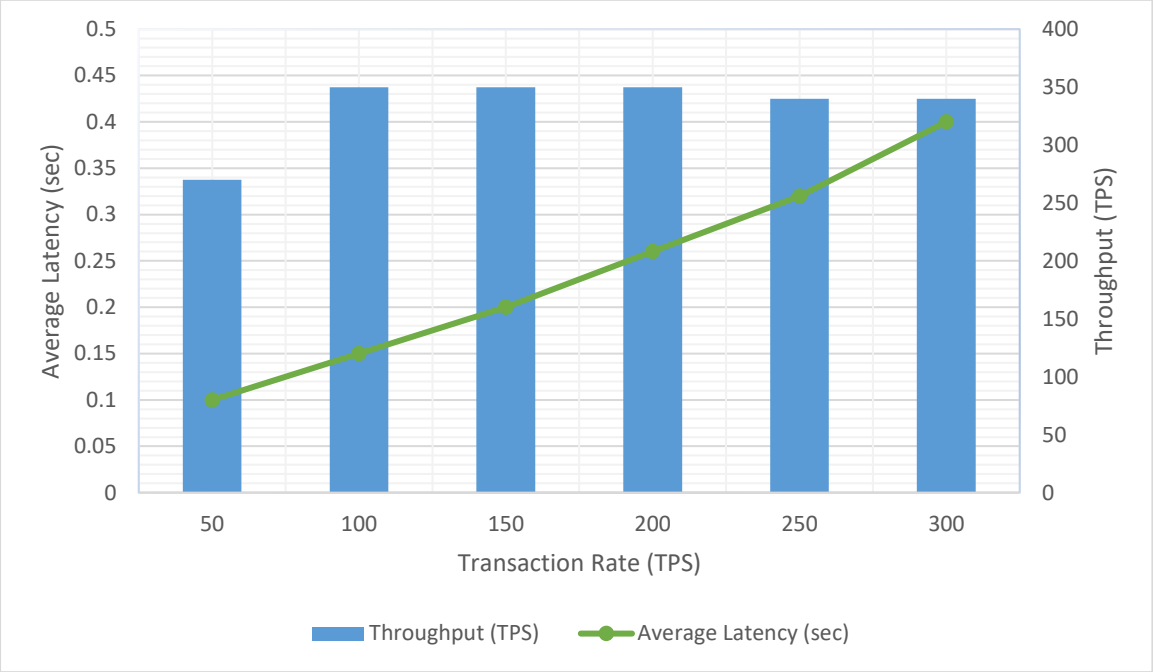

Figure 4.16: Evaluating the second scenario: reading transactions mode on the network latency and throughput.

4.5.3. Resource Consumption

This section elucidates the results of an in-depth study aiming to investigate the consumption of distinct system resources such as CPU, Memory, and Disk by a singular blockchain peer. These insights are crucial for blockchain users and administrators, offering a pivotal understanding of blockchain networks' optimal functionality and resource efficiency.

Figure 4.17: Peer's average CPU usage.

A significant proportion of CPU consumption is witnessed during the execution of Chaincodes, with the level of consumption heavily depending on the complexity of the business logic embedded within the contract. Contracts, especially those integrating complex elements like encryption and loops, demand more CPU resources. Moreover, computing the world state's hash and committing blocks contribute to CPU consumption.

In memory utilization, substantial consumption occurs when the virtual machine or Docker retrieves account data from the world state during the execution of contracts and initiates arrays. Grasping the nuances of memory consumption is pivotal for enhancing operational efficiency, especially during periods of elevated demand.

The blockchain application earmarks a specific segment of disk space for storing critical data, including the world state. The dedicated allocation of disk space is essential during complex blockchain operations, such as the commitment of blocks and execution of contracts, enabling the efficient management of resources.

Utilizing unique consensus protocols orchestrates synchronising a peer's state across varying blockchain systems. These protocols are integral for appending transactions to the network and facilitating the seamless transfer of block data.

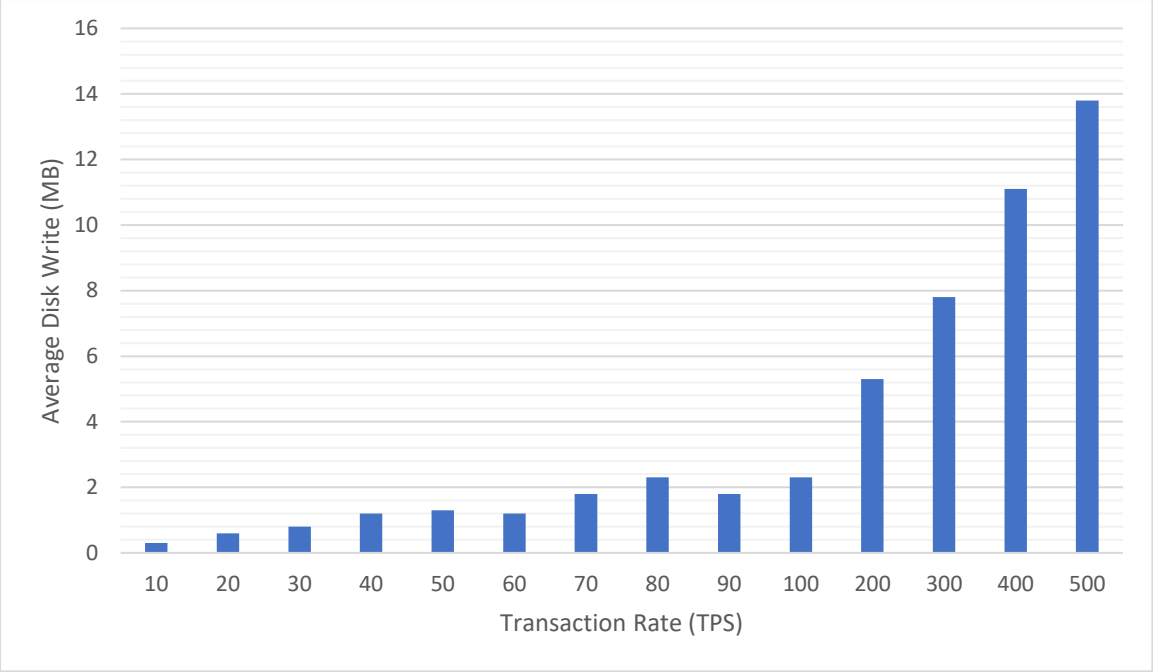

Figure 4.18: Peer's average disk write usage.

Several transaction batches were analysed to delve deeper into these aspects, and the derived results are visually represented in Figures 4.17 to 4.19. Figure 4.17 outlines the average CPU utilization of a single peer, depicting a decrease in average CPU utilization with the inclusion of more components in the network. The representation in Figure 4.18 details the average disk write usage, illustrating a proportional increase with the enhancement in components and batch sizes. Figure 4.19, focusing on average memory consumption, reflects similar trends, indicating optimal system operation at around 100 transactions per second (tps).

The intricate insights and visual representations contained in this section provide a nuanced understanding of the complex resource consumption patterns observed within a singular blockchain peer. This knowledge is invaluable for refining and advancing blockchain technology, enabling the development of optimized and efficient systems.

It is paramount to emphasize the implications of these findings, as they contribute to the enhancement of resource allocation and utilization and ensure the smooth operation of the blockchain network. This focused examination of a single peer lays down foundational insights, paving the way for the evolution and refinement of broader and more intricate blockchain systems in future explorations.

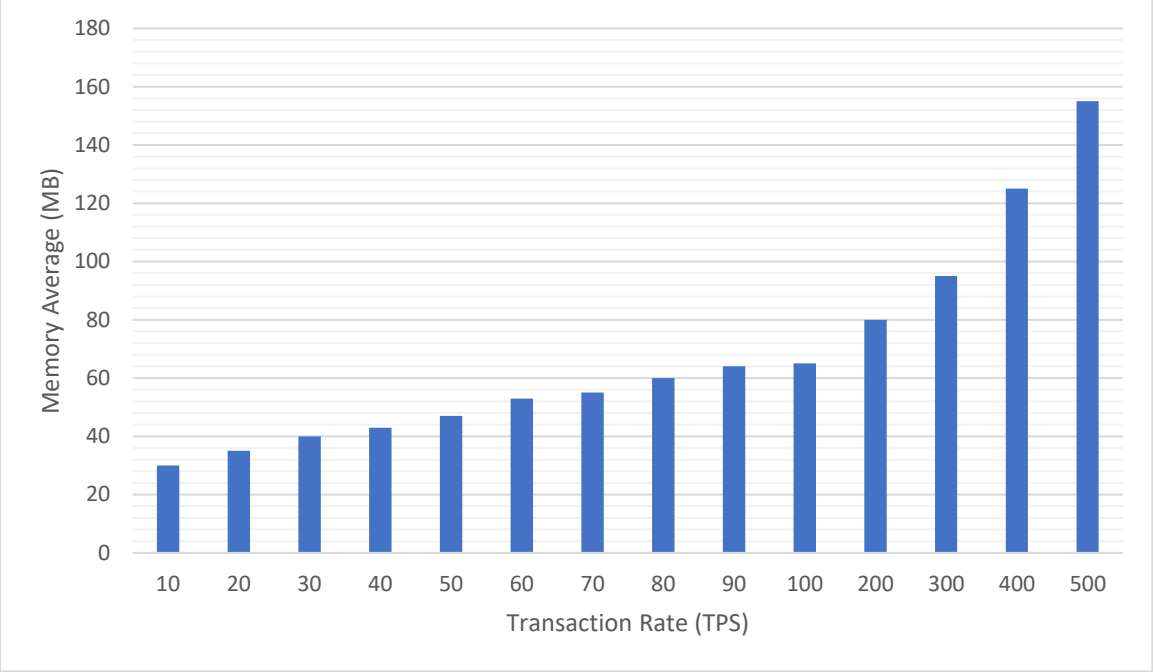

Figure 4.19: Peer's average memory consumption.

The comprehensive analysis and subsequent insights presented in this section are instrumental in fostering innovation and enabling the optimization of blockchain technology by providing a more profound understanding of resource utilization, operational mechanics, and the behaviour of blockchain systems under varying operational demands.

4.6. Summary

This chapter has presented the research and development of a Post-Quantum Hyperledger Fabric Blockchain solution tailored for IoT domain. The primary objective of this study was to design and implement a secure and scalable blockchain-based framework capable of addressing the unique challenges and requirements of IoT systems, particularly considering the potential threat posed by quantum computing advancements.

An overview of Hyperledger projects was provided, focusing on the Hyperledger Fabric platform, highlighting its key components, architecture, and consensus mechanism. A comprehensive solution proposal for integrating post-quantum cryptography into Hyperledger Fabric was then presented, detailing the selection and integration of suitable post-quantum cryptographic algorithms.

The proposed Post-Quantum Hyperledger Fabric implementation in the IoT context illustrated creating a custom blockchain network with IoT-specific smart contracts, configuring post-quantum cryptographic algorithms, and setting up the network topology to accommodate IoT devices.

An in-depth evaluation and performance analysis was conducted to assess the efficiency and effectiveness of the proposed solution. The performance of post-quantum cryptographic algorithms was compared to traditional ECDSA, and the overall performance of Hyperledger Fabric was analysed in various IoT scenarios. The results demonstrated the system's ability to maintain high throughput and low latency, even under increasing transaction rates, showcasing its suitability for diverse IoT use cases.

In conclusion, this chapter has demonstrated the successful development and evaluation of a Post-Quantum Hyperledger Fabric Blockchain solution for IoT domain. The proposed system effectively addresses IoT networks' security and scalability challenges while maintaining satisfactory performance. This research contributes to the ongoing effort of securing and managing IoT systems using blockchain technology and post-quantum cryptography, paving the way for more robust, resilient, and efficient IoT infrastructures in a future threatened by quantum computing advancements.

Chapter 5 Markov Chain Monte Carlo Falcon

5.1. Introduction

This chapter introduces a groundbreaking exploration into the amalgamation of Monte Carlo Markov Chain (MCMC) algorithms with the Falcon [133] signature scheme, a pivotal element in post-quantum cryptography known for its robust security and efficiency. The principal objective of this venture is to scrutinize the implications of embedding MCMC algorithms as trapdoor samplers in Falcon, focusing on the IMHK and SMK algorithms [97] specifically designed and implemented for this purpose.

The essence of this integration is to scrutinize the balance between enhanced security and signature speed. By meticulously modifying the signature generation process of Falcon to incorporate the MCMC algorithms, a significant reduction in the standard deviation of the trapdoor sampler is achieved. This reduction is critical as it potentially refines the security while maintaining acceptable signature speeds, hence offering a harmonious compromise between the two crucial components.

Through in-depth analysis and evaluation, this work elucidates the performance and security ramifications of the modified Falcon signature schemes and sheds light on their resilience against prevailing attacks, including key recovery and signature forgery. The variations in security parameters and their repercussions on the security stance of the system are also meticulously examined.

5.2. Overview of Falcon Signature Scheme

Although various signature schemes claim to offer post-quantum security, widespread adoption remains elusive. Recent international research on post-quantum schemes has focused on the National Institute of Standards and Technology (NIST) Post-Quantum Cryptography projects [157], which aim to identify a select few viable schemes for eventual standardization. In its third round, the process has narrowed to seven encryption schemes. Falcon is one of these finalists.

Several factors contributed to the selection of Falcon as the signature scheme for this study.

- Falcon is among the three finalists in the NIST-hosted competition, indicating its viability across various metrics.
- The primary design principle of Falcon is compactness, focusing on reducing the bitsize of public keys and digital signatures, which is a crucial factor in the viability of post-quantum schemes. It was the smallest of the remaining third-round digital signature scheme finalist proposals [158].
- Falcon is capable of efficient signature generation and verification on modern computers, which is a vital factor in the viability of post-quantum schemes.
- The modular design of Falcon allows for modifying a part of its implementation without significantly impacting the underlying lattice-based system.
- Falcon is based on a lattice-based scheme, offering the advantages described in Section 2.4.1.

Falcon's top-level architecture is straightforward: it utilizes Gentry, Peikert, and Vaikuntanathan's [159] framework for constructing hash-and-sign lattice-based signature schemes. This framework requires two elements:

1. NTRU lattices, a class of cryptographic lattices, are employed due to their compactness and computational speed-ups.
2. The trapdoor sampler in Falcon uses Fast Fourier sampling to achieve rapid and secure computations. Within the scope of this project, this trapdoor sampler is modified.

In summary, the Falcon signature system comprises:

Falcon = GPV framework + NTRU lattices + Fast Fourier sampling

5.2.1. The Gentry-Peikert-Vaikuntanathan Framework

A high-level description of the GPV framework can be outlined as follows:

- $q-$ ary lattice Λ is generated using the public key, which contains a full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ where $m > n$
- $\Lambda_q^{\perp}$ is generated using the private key, which contains $\mathbf{B} \in \mathbb{Z}_q^{m \times m}$, is the lattice orthogonal to Λ modulo q. At the same time, the rows of $\mathbf{A}$ and $\mathbf{B}$ should be a pairwise orthogonal: $\mathbf{B} \times \mathbf{A}^{\mathrm{t}} = \mathbf{0}$.

- A signature of message m is a short value $\mathbf{s} \in \mathbb{Z}_q^m$ and it needs to verify $\mathbf{s}\mathbf{A}^{\mathrm{t}} = H(\mathrm{m})$ where H is a hash function.
- To compute a valid signature, a preimage $\mathbf{c_0} \in \mathbb{Z}_q^m$ is first computed, which verifies $\mathbf{c_0}\mathbf{A}^{\mathrm{t}} = H(\mathrm{m})$ where c_0 is not necessarily short and $m \geq n$. Then, a vector $\mathbf{v} \in \Lambda_q^{\perp}$ close to $\mathbf{c_0}$ is computed using matrix $\mathbf{B}$. A valid signature is $\mathbf{s} = \mathbf{c_0} - \mathbf{v}$.

Klein's algorithm [160], based on randomized rounding, was used to sample the vector v. Without a correct trapdoor sampler, the private basis $\mathbf{B}$ would be exposed, threatening the scheme's security. As part of this project, the goal is to leverage the modularity of Falcon to suggest a minor adjustment to the trapdoor sampler being used. Specifically, the attempt is to replace Klein's algorithm with MCMC sampling.

5.2.2. NTRU Lattices

When instantiating the GPV framework, the choice of lattices is the initial step. Falcon utilizes the class of NTRU lattices developed by Hoffstein, Pipher, and Silverman [161]. These lattices include an additional ring structure that not only reduces the number of public keys by a factor of $O(n)$, but also accelerates several calculations by a factor of at least $O(n/\log n)$.

Let $\phi = x^n + 1$ for $n = 2^k$ and $q \in \mathbb{N}^{\star}$. The NTRU private key sets consist of four polynomials $f, g, F, G \in \mathbb{Z}[x]/(\phi)$, which satisfy the equation:

$$fG - gF = q \bmod \phi \tag{5.1}$$

The public key h can be defined $h \leftarrow g \cdot f^{-1} \bmod q$, given that f is invertible modulo q. It is possible to verify that the matrices $\begin{bmatrix} 1 & h \\ 0 & q \end{bmatrix}$ and $\begin{bmatrix} f & g \\ F & G \end{bmatrix}$ generate the same lattice, while the first matrix contains two large polynomials compared to the second one. Although f and g are quite small, it remains challenging to find small polynomials f', g' that solves the equation $h = g' \cdot (f')^{-1} \bmod q$. The hardness of this problem constitutes the NTRU assumption.

5.2.3. Instantiation with the GPV Framework

GPV framework instantiation over NTRU lattices can be explained as follows:

- The public basis is $\mathbf{A} = [\ 1 \mid h^*\ ]$ (equivalent to knowing h)
- The secret basis is

$$\mathbf{B} = \begin{bmatrix} g & -f \\ G & -F \end{bmatrix} \tag{5.2}$$

- Matrices **A** and **B** are orthogonal, where $\mathbf{B} \times \mathbf{A}^* = 0 \bmod q$.
- The message m's signature consists of a salt r and two polynomials (s_1, s_2) such that $s_1 + s_2 h = H(\mathrm{r}\|\mathrm{m})$. Since s_1 can be calculated from m, r, and s_2, the signature to be sent can be (r, s_2).

5.2.4. Fast Fourier Trapdoor Sampler

The trapdoor sampler is the second choice when instantiating the GPV framework. A trapdoor sampler accepts a matrix **A**, a trapdoor **T** and a target **c** as inputs and generates a short vector **s** such that $\mathbf{s}^{\mathrm{t}}\mathbf{A} = \mathbf{c} \bmod q$. Using notations in [159], this is equivalent to finding $\mathbf{v} \in \boldsymbol{\Lambda}_q^{\perp}$ that is close to $\mathbf{c}_0$. The closer the **v** is to $\mathbf{c}_0$ the more secure the trapdoor sampler will be [133]. Additionally, the trapdoor sampler needs to be both effective and safe. In Falcon, a randomized variant of the fast Fourier nearest plane sampler by Ducas and Prest [159] is used.

5.2.5. Significance of σ

The security of the GPV frameworks depends on the length of the signature $\mathbf{s} = \mathbf{c_0} - \mathbf{v}$ where $\mathbf{c_0} \in \mathbb{Z}_q^m$ is the distance between the preimage and $\mathbf{v} \in \Lambda_q^{\perp}$ which is computed from the private basis **B.** The shorter the signature **s**, the greater the security of the signature scheme [133]. Falcon uses a bound to check system security. To be accepted, the signature (s_1, s_2) must satisfy the inequality:

$$\|(s_1, s_2)\|^2 \leq \lfloor \beta^2 \rfloor \tag{5.3}$$

with

$$\beta = 1.1 \cdot \sigma\sqrt{2n} \tag{5.4}$$

where σ is the standard deviation of the trapdoor sampler. Therefore, the smaller σ, the shorter the signature generated, and the more secure the signature scheme. This is the motivation

behind adopting MCMC sampling, which allows for a smaller trapdoor sampler, enhancing the security of the signature system.

A scheme with a smaller sigma is more secure but has a lower limit. A too-small sigma could potentially leak the secret basis **B**. $\sigma = 0$ makes the signature scheme open to learning attacks for all known samplers [133]. As a result, Falcon uses the following lower limit for the standard deviation:

$$\sigma = \frac{1}{\pi} \cdot \sqrt{\frac{\log\left(4n(1 + 1/\epsilon)\right)}{2}} \cdot 1.17 \cdot \sqrt{q} \geq \eta_{\epsilon}(\mathbb{Z}^{2n}) \cdot \|\mathbf{B}\|_{GS} \tag{5.5}$$

This lower bound is known as the smoothing parameter $\eta_{\epsilon}(\Lambda)$ of a lattice and is defined as the smallest s such that $\rho_{1/s}(\Lambda^{*}\backslash\{0\}) \leq \epsilon$. In Falcon, this is approximately equivalent to $\frac{\sqrt{q}}{\sqrt{2\pi}} \approx 44.22$, since $q = 12289$ [133].

5.3. Incorporating MCMC Sampling as a Trapdoor Sampler

The Markov Chain Monte Carlo (MCMC) sampling technique has been widely adopted for solving complex problems in various scientific domains. This section explores the incorporation of MCMC sampling as a trapdoor sampler in the Falcon signature scheme.

MCMC sampling is a powerful method for obtaining samples from complex probability distributions, especially in high-dimensional spaces. The primary idea behind MCMC is to construct a Markov chain with the desired distribution as its stationary distribution. The resulting samples will approximate the target distribution by simulating this Markov chain for sufficient steps.

The trapdoor sampler in the Falcon signature scheme is used to sample from a discrete Gaussian distribution over a lattice, which is a crucial step for generating lattice-based signatures. The original Falcon trapdoor sampler utilizes the fast Fourier sampling (ffsampling) method, which relies on the Gaussian distribution's Fourier transform properties [160].

To incorporate MCMC sampling as a trapdoor sampler in the Falcon signature scheme, the ffsampling method is replaced with an MCMC-based sampling algorithm [162]. This algorithm involves constructing a Markov chain that converges to the desired distribution and then running the chain for sufficient iterations to obtain samples that approximate the target distribution. The MCMC algorithm can be designed to adaptively update the proposal distribution based on the sampled values, leading to improved efficiency and convergence properties.

There are several benefits to incorporating MCMC sampling as a trapdoor sampler in the Falcon signature scheme. First, it allows for more flexible and efficient sampling from the target distribution, as MCMC methods can be tailored to the specific problem. Second, MCMC sampling can provide improved security guarantees compared to the original ffsampling method, as it can better explore the space of possible lattice points and avoid specific attacks that exploit the sampler's structure.

5.3.1. Overview of MCMC Sampling

MCMC is composed of two components - Markov Chains and Monte Carlo methods.

Markov chains are stochastic processes that change with time following the Markov property. This feature indicates that the current state X_i of the system is only dependent on its immediate predecessor X_{i+1}. The initial state of the system X_0 has no predecessor. Thus, these processes result in a chain of events.

Monte Carlo methods use random sampling instead of analytical methods to determine the parameters of a probability distribution. In MCMC, each random sample generates the following random sample [163]. Many random samples are taken to acquire approximate solutions to analytically unsolvable or computationally expensive issues, and the desired distribution can be approximated.

MCMC sampling is useful for sampling from distribution without computing an analytical solution. One example is Bayesian inference (see equation 5.6), where the goal is to sample from a posterior distribution $P(\Theta|Data)$, based on the prior distribution $P(\Theta)$ and the

likelihood $P(Data|\Theta)$, without having to derive an analytical solution for the evidence $P(Data)$.

$$P(\Theta|Data) = \frac{P(Data|\Theta)P(\Theta)}{P(Data)} \tag{5.6}$$

To accomplish this, the steady state of the Markov chain must be equivalent to the intended probability distribution. Once the Markov chain reaches equilibrium, any additional MCMC samples are equivalent to sampling from the desired probability distribution. The Metropolis-Hastings [164] algorithm is one of these MCMC sampling methods.

5.3.2. Metropolis-Hastings Algorithm

As long as a target distribution f, proportional to the desired distribution P, can be approximated, sampling from P is possible once the Markov chain reaches equilibrium with the Metropolis-Hastings algorithm. The Metropolis-Hastings algorithm can be characterized as a random walk-in search of relatively better approximations of the desired distribution P. The algorithm can be described as follows:

1. Choose an initial state x_t, where $t = 0$ with an arbitrary point.
2. Generate a random state proposal y from $p(y|x)$.
3. Calculate the acceptance ratio $\alpha(x_t, y) = min\left\{1, \frac{\pi(y)p(y,x_t)}{\pi(x_t)p(x_t,y)}\right\}$.
4. Generate a uniform random number $u \in [0,1]$:
   - Accept the new state if $u \leq \alpha(x_t, y)$ and set $x_{t+1} = y$.
   - Reject the new state if $u \geq \alpha(x_t, y)$ and set $x_{t+1} = x_t$.
   - $t = t + 1$.
5. Repeat steps 2 – 4 until convergence.

Thus, the new state will always be accepted if the new model parameter y is more probable than the existing model parameter x_t. However, if the new state is less probable than x_t, y may still be accepted with a probability directly proportional to the ratio $\frac{\pi(y)p(y,x_t)}{\pi(x_t)p(x_t,y)}$. Consequently, the Monte-Carlo Markov chain will eventually converge on the probability density of the desired probability distribution P, allowing future random samples to be drawn from it.

5.3.3. Security Advantage of MCMC Sampling over Klein's Algorithm

The independent Metropolis-Hastings-Klein (MHK) and the symmetric Metropolis-Klein (MK) algorithms were created as an application of MCMC sampling to lattice Gaussian distributions [97].

Klein's algorithm [160] is used in the IMHK algorithm to independently sample the next state y in the Markov chain; the previous state x is only used to calculate the acceptance probability. This Markov chain is uniformly ergodic and converges exponentially quickly to its steady state [97], allowing random samples from the lattice Gaussian distribution.

Klein's technique is used to sample the next state y in the SMK algorithm, and the prior state x is utilized in this process, unlike the IMHK. This Markov chain is geometrically ergodic, with steady-state convergence highly sensitive to the initial state. The SMK is easier to implement.

To ensure the security of the trapdoor sampler, the size of the sampler's standard deviation is crucial: if it is too large, the generated vectors will not be short enough, and the signature scheme will be less secure; if it is too small, the secret basis may be revealed [133]. MCMC sampling with the IMHK method increases security for slightly below GPV's parameter at the expanse of slower signature schemes. In conjunction with deploying the fast Fourier sampler in Falcon, it was conceivable that a fast Fourier variation of MCMC sampling may replace the fast Fourier variant of Klein's algorithm in Falcon, achieving a security benefit at the expense of slightly slower signature times.

5.4. Design and Analysis

In this section, the design and analysis of MCMC-based algorithms for use as trapdoor samplers in the Falcon signature scheme are discussed. The goal is to develop efficient and secure MCMC algorithms that can replace the original ffsampling method in Falcon while maintaining the desirable properties of the signature scheme.

First, it is essential to identify the target distribution from which samples need to be drawn. In the case of Falcon, this distribution is a discrete Gaussian distribution over a lattice. Once the target distribution is specified, an appropriate MCMC method must be chosen. As mentioned in the previous section, MCMC sampling, based on either the IMHK algorithm or the SMK method, could be employed in the Falcon signature scheme to obtain higher security assurance via a lower σ bound.

After selecting an MCMC method, the design of the Markov chain must be carefully considered. This involves determining the proposal distribution, the transition probabilities, and the acceptance criteria for the chain. The proposal distribution should be designed to efficiently explore the space of lattice points, while the transition probabilities and acceptance criteria must ensure that the Markov chain converges to the target distribution.

Next, the convergence properties of the MCMC algorithm must be analysed. This includes examining the mixing time and the number of iterations required for the Markov chain to converge to the stationary distribution. Various factors, such as the choice of the MCMC method, the proposal distribution, and the structure of the target distribution, influence the mixing time. It is crucial to ensure that the mixing time is reasonably small, as this directly affects the efficiency and practicality of the MCMC-based trapdoor sampler.

In addition to the convergence properties, the security of the MCMC algorithm must be carefully evaluated. This involves analysing the resistance of the algorithm to various known attacks, such as lattice reduction and forgery attacks. The security of the MCMC-based trapdoor sampler can be enhanced by adopting suitable countermeasures, such as using a lower standard deviation for the Gaussian distribution or employing additional post-processing techniques to obscure the sampled lattice points.

Finally, the performance of the MCMC-based algorithms should be compared to the original ffsampling method used in Falcon. This comparison should consider signature generation speed, signature size, and bit security. The goal is to develop MCMC-based algorithms that offer improved performance and security compared to the original ffsampling method while maintaining the overall efficiency and practicality of the Falcon signature scheme.

In conclusion, designing and analysing MCMC-based algorithms for trapdoor samplers in the Falcon signature scheme is a critical task involving several interconnected steps. By carefully

considering the target distribution, the choice of the MCMC method, the convergence properties, and the security aspects of the algorithm, it is possible to develop efficient and secure MCMC-based trapdoor samplers that can enhance the performance of lattice-based cryptographic schemes.

5.5. Implementing MCMC Algorithms in Falcon

Implementing MCMC algorithms in the Falcon signature scheme involves integrating the chosen MCMC method with the existing trapdoor sampler for generating signatures. This section details the process of incorporating the MCMC algorithms into Falcon while maintaining the integrity of the original signature scheme.

As discussed above, both the SMK algorithm and the IMHK algorithm were designed for trapdoor sampling in lattice Gaussian distributions to provide a security benefit by reducing σ. Therefore, both were implemented in Falcon to examine their effects on the signature system. First, the implementation of the IMHK algorithm is explained due to its existing theoretical upper bound [162], followed by the implementation of the SMK method.

5.5.1. Original Falcon

The first step of implementing MCMC sampling in Falcon is to modify the signature generation algorithm from the original Falcon. In Algorithm 1, the original Falcon signature generation algorithm is referenced from the document [133]. Given a secret key sk and a message m, the signer uses sk to sign m in the following ways:

1. Uniformly generate the random salt r in $\{0,1\}^{320}$ and obtain a string concatenating the salt and message m (r ‖ m) that is then hashed to the polynomial $c \in \mathbb{Z}_q[x]/\phi$
2. Using the secret key to compute two short values s_1, s_2 such that $s_1 + s_2 h = c \bmod q$, where $q = 12289$
3. s_2 is compressed to a bitstring s.
4. The signature is the pair of (r, s).

Algorithm 1: sign(m, sk, $\lfloor \beta^2 \rfloor$) Original Falcon

Input: A message (m), a secret key (sk), a bound $\lfloor \beta^2 \rfloor$

Output: A signature (sig) of a message (m)

1 r $\leftarrow \{0,1\}^{320}$ uniformly

2 c $\leftarrow \text{HashToPoint}(\text{r}||\text{m}, q, n)$

3 $\mathbf{t} \leftarrow \left(-\frac{1}{q}\text{FFT}(c) \odot \text{FFT}(F), \frac{1}{q}\text{FFT}(c) \odot \text{FFT}(f)\right)$ // $\mathbf{t} = \left(\text{FFT}(c), \text{FFT}(0) \cdot \widehat{\mathbf{B}}^{-1}\right)$

4 **do**

5 | **do**

6 | | $\mathbf{z} \leftarrow \text{ffsampling}_n(\mathbf{t}, \text{T})$

7 | | $\text{s} = (\mathbf{t} - \mathbf{z})\widehat{\mathbf{B}}$ // at this point, s follows a Gaussian distribution: $\text{s} \sim \text{D}_{(\text{c},0)+\Lambda(\mathbf{B}),\sigma,0}$

8 | **while** $\|\text{s}\|^2 > \lfloor \beta^2 \rfloor$

9 | $(s_1, s_2) \leftarrow \text{invFFT}(\text{s})$

10 | $\text{str} \leftarrow \text{Compress}(s_2, 8 \cdot \text{sbytlen} - 328)$ // Remove 1 byte for the header, and 40 bytes for r

11 **while** $\text{s} = \perp$

12 **return** sig = (r, s)

5.5.2. IMHK Algorithm

The following approach is used to implement the IMHK algorithm in Falcon:

1. First, ffsampling is used to generate the initial Markov state z_0, η_0 and i_{mix}. Note that the σ is lower than the original σ in Falcon to achieve a greater security assurance.
    - As a part of MCMC sampling, η_0 is used later to calculate the acceptance ratio α.
    - i_{mix} is the theoretical upper bound derived for the mixing time of the IMHK [162], given by $\Pi_{i=1}^{2n}\theta_3(2\pi\sigma_i^2)$, where the Jacobi theta function $\theta_3(\tau) = \sum_{n=-\infty}^{+\infty} e^{-\pi\tau n^2}$ with $\tau > 0$, and $\sigma_i = \sigma/\|\hat{\mathbf{b}}_i\|$ with $\hat{\mathbf{b}}_i$ being the i^{th} Gram-Schmidt (GS) vector of the private basis **B.**
    - Thus, to calculate i_{mix}, Theta3 was implemented (see Algorithm 3) with $\tau = 2\pi\sigma_i^2$ as input. In the Falcon implementation, σ_i corresponds to the leaf values stored in the Falcon tree (T), extracted during ffsampling (see Line 2 in Algorithm 5).
2. Then, it loops for i_{mix} number of iterations generating the next state z, η with ffsampling, and decides whether to remain current or transit to the new state.

- Calculates the acceptance ratio α using equation 5.7 [97]. with $x \in \mathbb{Z}^n$ corresponding to the previous state and $y \in \mathbb{Z}^n$ corresponding to the next state. To avoid overflow with products, the α in equation 5.7 is described as $\alpha = \min\{0, \eta - \eta_0\}$, where $\eta = \sum_{i=1}^{n} \log\left(\rho_{\sigma_i,\bar{y}_1}(\mathbb{Z})\right)$;
- The SumProb function (see Algorithm 6) is used to calculate $\sum_{\bar{x}_i \in \mathbb{Z}} e^{-\|x_i - \bar{x}_4\|^2/2\sigma_i^2}$ and $\sum_{\bar{y}_1 \in \mathbb{Z}} e^{-\|y_i - \bar{y}_i\|^2/2\sigma_i^2}$ within the ffsampling algorithm.
- Next, a uniform random number $u \in [0,1]$ is generated.
- If $\log(u) \leq \alpha$, the algorithm transits to the new state; otherwise, it remains in the current state.

$$\begin{aligned} \alpha(\mathbf{x}, \mathbf{y}) &= min\left\{1, \frac{\prod_{i=1}^{n} \rho_{\sigma_i,\bar{y}_i}(\mathbb{Z})}{\prod_{i=1}^{n} \rho_{\sigma_i,\bar{x}_1}(\mathbb{Z})}\right\} \\ &= min\left\{1, \frac{\prod_{i=1}^{n} \sum_{\bar{y}_1 \in \mathbb{Z}} e^{-\|y_i - \bar{y}_i\|^2/2\sigma_i^2}}{\prod_{i=1}^{n} \sum_{\bar{x}_i \in \mathbb{Z}} e^{-\|x_i - \bar{x}_4\|^2/2\sigma_i^2}}\right\} \end{aligned} \tag{5.7}$$

Algorithm 2: sign(m, sk, $\lfloor \beta^2 \rfloor$) with IMHK Algorithm

Input: A message (m), a secret key (sk), a bound $\lfloor \beta^2 \rfloor$

Output: A signature (sig) of message (m)

$r \leftarrow \{0,1\}^{320}$ uniformly

$c \leftarrow \text{HashToPoint}(r||m, q, n)$

$\mathbf{t} \leftarrow \left(-\frac{1}{q}\text{FFT}(c) \odot \text{FFT}(F), \frac{1}{q}\text{FFT}(c) \odot \text{FFT}(f)\right)$ // $\mathbf{t} = \left(\text{FFT}(c), \text{FFT}(0) \cdot \widehat{\mathbf{B}}^{-1}\right)$

$\mathbf{z_0}, \eta_0, i_{mix} \leftarrow \text{ffsampling}_n(\mathbf{t}, T, 0, 1)$ // Generate initial state $\mathbf{z_0}$

do

  do

    for $i = 1, \ldots, i_{mix}$ **do**

      $\mathbf{z}, \eta, \_ \leftarrow \text{ffsampling}_n(\mathbf{t}, T, 0, 1)$

      $\alpha \leftarrow min\{0, \eta - \eta_0\}$

      $u \leftarrow [0,1]$ uniformly

      if $\log(u) \leq \alpha$ **then**

        $\mathbf{z_0} \leftarrow \mathbf{z}$

        $\eta_0 \leftarrow \eta$

      end if

end for

$s = (\mathbf{t} - \mathbf{z})\widehat{\mathbf{B}}$ // at this point, s follows a Gaussian distribution: $s \sim D_{(c,0)+\Lambda(\mathbf{B}),\sigma,0}$

while $\|s\|^2 > \lfloor \beta^2 \rfloor$

$(s_1, s_2) \leftarrow \text{invFFT}(s)$

$str \leftarrow \text{Compress}(s_2, 8 \cdot sbytelen - 328)$ // Remove 1 byte for the header, and 40 bytes for r

while $s = \perp$

return $sig = (r, s)$

Algorithm 3: $\text{Theta3}(\tau)$

Input: $\tau > 0$

Output: $\theta_3(\tau) = \sum_{n=-\infty}^{+\infty} e^{-\pi\tau n^2}$

$l \leftarrow \tau - 10$

$h \leftarrow \tau + 10$

$\theta \leftarrow 0$

for $n = l \ldots h$ **do**

$\theta \leftarrow \theta + e^{-\pi\tau n^2}$

end for

return θ

Algorithm 4: $\text{Sampler\_smallZ}(\mu, \sigma')$ for $\sigma' < \sigma_{min}$

Input: $\mu, \sigma' \in \mathbb{R}$ with $\sigma' < \sigma_{min}$

Output: An integer $z \in \mathbb{Z}$ sampled from a distribution very close to $D_{\mathbb{Z},\sigma',\mu}$, $\gamma = \sum_{z\in\mathbb{Z}} \rho_{\mu,\sigma'}(z)$

$l \leftarrow \lceil \mu \rceil - 5$

$h \leftarrow \lceil \mu \rceil + 5$

$size \leftarrow h - l + 1$

$\text{cdt} \leftarrow []$

$\text{int} \leftarrow []$

$\text{pdt} \leftarrow []$

$\gamma \leftarrow \text{SumProb}(\mu, \sigma', 5)$

for $x = 0 \ldots size$ **do**

$\text{int}[x] \leftarrow x + l$

$\text{pdt}[x] \leftarrow \dfrac{e^{-\frac{|\text{int}[x]-\mu|^2}{2\sigma^2}}}{\gamma}$

end for

$\text{cdt} \leftarrow \text{CumSum}(\text{pdt}, size - 1)$

$u \leftarrow [0,1]$ uniformly

$i \leftarrow 0$
for $x = 0 \ldots size$ **do**
  if $\mathrm{cdt}[x] < u$ **then**
    $i \leftarrow i + 1$
  end if
end for
$z \leftarrow \mathrm{int}[i]$
return $z,\ \gamma$

Algorithm 5: Modified $\mathrm{ffsampling}_n(\mathbf{t}, \mathrm{T}, \eta, i_{mix})$ for IMHK Algorithm

if $n = 1$ **then**
  $\sigma' \leftarrow$ T.value
  if $\sigma' > \sigma_{min}$ **then**
    $z_0 \leftarrow \mathrm{SamplerZ}(t_0, \sigma')$
    $z_1 \leftarrow \mathrm{SamplerZ}(t_1, \sigma')$
    $\gamma_0 \leftarrow \mathrm{SumProb}(t_0, \sigma', 18)$
    $\gamma_1 \leftarrow \mathrm{SumProb}(t_1, \sigma', 18)$
  end if
  else
    $z_0, \gamma_0 \leftarrow \mathrm{Sampler\_smallZ}(t_0, \sigma')$
    $z_1, \gamma_1 \leftarrow \mathrm{Sampler\_smallZ}(t_1, \sigma')$
  end
  $\eta \leftarrow \eta + \log(\gamma_0) + \log(\gamma_1)$
  $i_{mix} \leftarrow i_{mix} \times \left(Theta3(2\pi\sigma'^2)\right)^2$
  return $\boldsymbol{z} = (z_0, z_1),\ \eta,\ i_{mix}$
end if
$(l, \mathrm{T}_0, \mathrm{T}_1) \leftarrow$ (T.value, T.leftchild, T.rightchild)
$\mathbf{t_1} \leftarrow \mathrm{splitfft}(t_1)$
$\mathbf{z_1}, \eta,\ i_{mix} \leftarrow \mathrm{ffsampling}_{n/2}(\mathbf{t_1}, \mathrm{T}_1, \eta, i_{mix})$
$z_1 \leftarrow \mathrm{mergefft}(\boldsymbol{z_1})$
$t'_0 \leftarrow t_0 + (t_1 - z_1 \odot l)$
$\mathbf{t_0} \leftarrow \mathrm{splitfft}(t'_0)$
$\mathbf{z_0}, \eta,\ i_{mix} \leftarrow \mathrm{ffsampling}_{n/2}(\mathbf{t_0}, \mathrm{T}_0, \eta, i_{mix})$
$z_0 \leftarrow \mathrm{mergefft}(\boldsymbol{z_0})$
return $\boldsymbol{z} = (z_0, z_1),\ \eta,\ i_{mix}$

Algorithm 6: $\mathrm{SumProb}(\mu, \sigma', r)$

Input: $\mu,\ \sigma' \in \mathbb{R}$, a range r
Output: $\gamma = \sum_{z \in \mathbb{Z}} \rho_{\mu,\sigma'}(z)$
$l \leftarrow \lceil \mu \rceil - r$

2 $h \leftarrow \lceil \mu \rceil + r$
3 $\gamma \leftarrow 0$
4 **for** $x = l \ldots h$ **do**
5 $\quad \gamma \leftarrow \gamma + e^{-\frac{|x-\mu|^2}{2\sigma^2}}$
6 **end for**
7 **return** γ

To achieve greater security, in the IMHK, a lower σ than the original σ in Falcon is used. As a result, the maximum norm of the generated signatures, $\lfloor \beta^2 \rfloor$, is also reduced due to the lower σ. More specifically, $\beta = 1.1 \cdot \sigma\sqrt{2n}$.

In Falcon, the SamplerZ algorithm is used within ffsampling to securely sample Gaussian samples $z \sim D_{\mathbb{Z},\sigma',\mu}$ for any $\sigma' \in [\sigma_{\min}, \sigma_{\max}]$ (σ' varies from degree n). Since σ is reduced in the IMHK, a slight variation of this algorithm, i.e., Sampler_smallZ is utilized to allow sampling with a $\sigma' < \sigma_{min}$ (see Algorithm 4).

5.5.3. SMK Algorithm

The signature generation algorithm has been modified to implement SMK in Falcon, as shown in Algorithm 7. The following modifications have been made:

1. The initial state z_0 generated in the same way as in the IMHK algorithm.
2. It loops for i_{mix} number of iterations generating the next state z:
   - Calculate the acceptance ratio α in line 9.
   - Next, a uniform random number $u \in [0,1]$ is generated.
   - If $u \leq \alpha$ the algorithm transits to the new state; otherwise, it remains at the current state.

Like the IMHK algorithm, a lower σ than the original σ in Falcon is used so that the maximum norm of the generated signatures, $\lfloor \beta^2 \rfloor$, is also reduced due to the lower σ.

Algorithm 7: Falcon sign(m) with SMK Algorithm

Input: A message (m), a secret key (sk), a bound $\lfloor \beta^2 \rfloor$

Output: A signature (sig) of message (m)

1 r $\leftarrow \{0,1\}^{320}$ uniformly

2 $c \leftarrow \text{HashToPoint}(r||m, q, n)$

3 $\mathbf{t} \leftarrow \left(-\frac{1}{q}\text{FFT}(c) \odot \text{FFT}(F), \frac{1}{q}\text{FFT}(c) \odot \text{FFT}(f)\right)$ // $\mathbf{t} = \left(\text{FFT}(c), \text{FFT}(0) \cdot \widehat{\mathbf{B}}^{-1}\right)$

4 $\mathbf{z_0} \leftarrow \text{ffsampling}_n(\mathbf{t}, T)$ // Generate initial state $\mathbf{z_0}$

5 **do**

6 **do**

7 **for** $i = 1, \ldots, i_{mix}$ **do**

8 $\mathbf{z} \leftarrow \text{ffsampling}_n(\mathbf{t}, T)$

9 $\alpha \leftarrow min\left\{1, e^{\frac{1}{2\sigma^2}\left(\|(\boldsymbol{t}-\mathbf{z_0})\widehat{\mathbf{B}}\|^2 - \|(\boldsymbol{t}-\mathbf{z})\widehat{\mathbf{B}}\|^2\right)}\right\}$

10 $u \leftarrow [0,1]$ uniformly

11 **if** $u \leq \alpha$ **then**

12 $\mathbf{z_0} \leftarrow \mathbf{z}$

13 **end if**

14 **end for**

15 $s = (\mathbf{t} - \mathbf{z})\widehat{\mathbf{B}}$ // at this point, s follows a Gaussian distribution: $s \sim D_{(c,0)+\Lambda(\mathbf{B}),\sigma,0}$

16 **while** $\|s\|^2 > \lfloor \beta^2 \rfloor$

17 $(s_1, s_2) \leftarrow \text{invFFT}(s)$

18 $str \leftarrow \text{Compress}(s_2, 8 \cdot sbytelen - 328)$ // Remove 1 byte for the header, and 40 bytes for r

19 **while** $s = \perp$

20 **return** $sig = (r, s)$

5.6. Performance Evaluation and Analysis

Evaluating the performance and analysing the modified Falcon signature scheme with the incorporated MCMC algorithms is crucial to ensure the desired security and efficiency levels are met. This section outlines the various aspects to consider while conducting a comprehensive assessment of the MCMC-based Falcon scheme.

5.6.1. IMHK Algorithm

Initial measurements of the mixing time of the IMHK method were based on the following theoretical upper bound [162]:

$$i_{mix} = \Pi_{i=1}^{2n} \theta_3(2\pi\sigma_i^2) \tag{5.8}$$

where the Jacobi theta function $\theta_3(\tau) = \sum_{n=-\infty}^{+\infty} e^{-\pi\tau^2}$ with $\tau > 0$, and $\sigma_i = \sigma/\|\hat{\mathbf{b}}_i\|$ with $\hat{\mathbf{b}}_i$ being the i^{th} Gram-Schmidt (GS) vector of the private basis **B**.

$\Pi_{i=1}^{2n}\theta_3(2\pi\sigma_i^2)$ for each σ_i was calculated using the Theta3 algorithm (see Algorithm 3) as part of the initial call to the modified ffsampling algorithm.

As part of the initial call to the modified ffsampling algorithm, the Theta3 algorithm was used to calculate $\Pi_{i=1}^{2n}\theta_3(2\pi\sigma_i^2)$ for each σ_i.

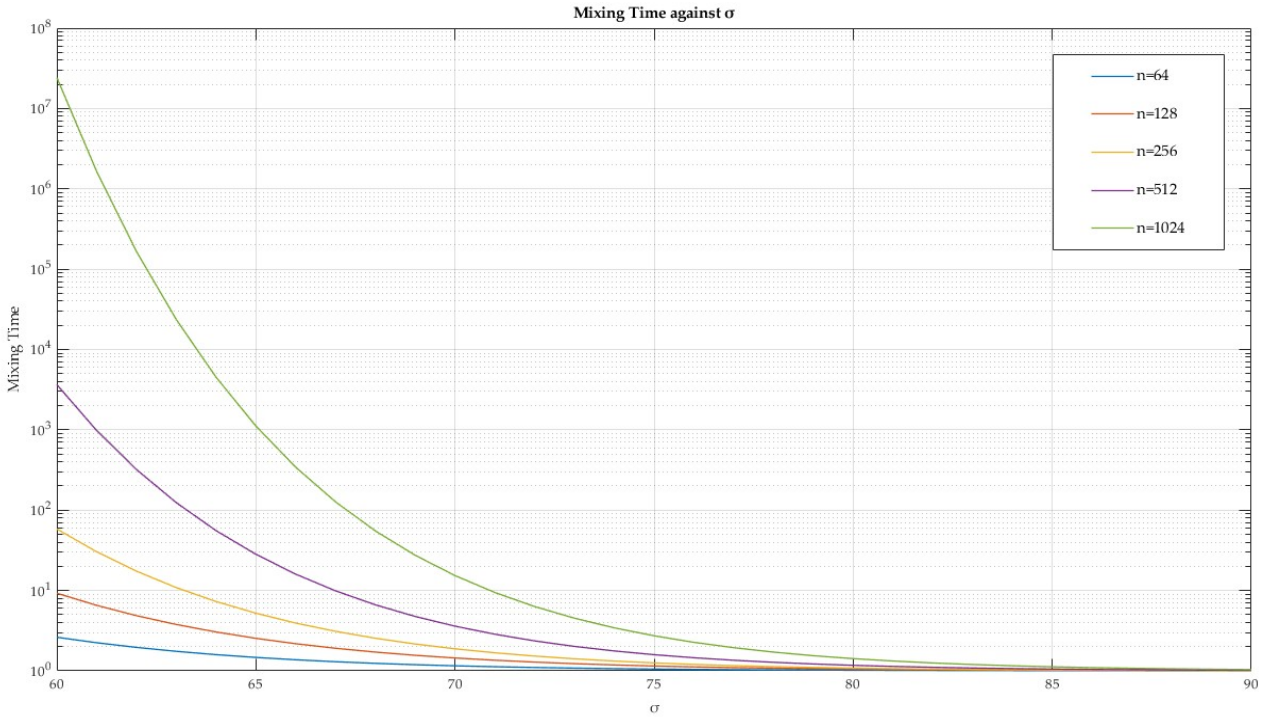

Figure 5.1: The theoretical upper bound of mixing time i_{mix} against $\sigma \in [60; 90]$.

Table 5.1: Comparison of σ on different parameters n of Falcon with IMHK.

Falcon Signature Scheme n	***σ*=60**	***σ*=65**	***σ*=68**	***σ*=70**	***σ*=75**	***σ*=80**
64	3	2	2	2	2	2
128	10	3	2	2	2	2
256	59	8	3	2	2	2

512	3650	29	7	4	2	2
1024	24250129	1114	56	16	3	2

As seen in Figure 5.1 and Table 5.1, i_{mix} decreases exponentially as the value of σ increases, consistent with the results given in [162]. For practical use, the most suitable range is $65 \leq \sigma \leq 75$ for n=512, where i_{mix} would range from 2 to 29, and $68 \leq \sigma \leq 80$ for n=1024, i_{mix} would range from 2 to 56 iterations.

Table 5.2 compares the signature speed of vanilla Falcon and Falcon utilising the IMHK algorithm in C programming language. From Table 5.2, it is possible to approximately halve the σ of the trapdoor sampler using the IMHK algorithm – from 165 to 75 for Falcon-512 and from 168 to 75 for Falcon-1024 with just two iterations of MCMC sampling. This results in a decrease in signature speed (by approximately four times for two iterations of MCMC sampling), signature generation required approximately 0.91ms for Falcon-512 and 1.86ms for Falcon-1024, thus offering an acceptable trade-off between signature speed and security.

Table 5.2: Comparison of the speed of signature generation between Falcon and IMHK Falcon in C implementation.

Signature Scheme	σ	Sign Time (ms)
Vanilla Falcon-512	165.74	0.20
Vanilla Falcon-1024	168.39	0.41
Falcon-512 with IMHK	75.00	0.91
	70.00	1.65
	65.00	14.74
Falcon-1024 with IMHK	80.00	1.86
	73.00	2.97
	68.00	22.23

5.6.2. SMK Algorithm

As illustrated in Table 5.3, for $\sigma = 60$, the SMK algorithm required approximately 437 ms to generate a signature, whereas the independent IMHK required around 2267 ms in C

implementation. Compared to vanilla Falcon-512 with σ = 165.74, as specified in Falcon [133], which required 0.2 ms, the SMK algorithm required a significantly longer signature time.

Table 5.3: Comparison of the speed of signature generation between vanilla Falcon, IMHK Falcon, and SMK Falcon in C implementation.

Signature Schema	σ	Sign Time (ms)
Vanilla Falcon – 512	165.74	0.20
Falcon – 512 with SMK	60	436.91
Falcon – 512 with IMHK	60	2266.96

5.7. Security Assessment and Known Attacks

The security assessment of the MCMC-based Falcon signature scheme is vital to ensure its robustness against known attacks and potential vulnerabilities. This section discusses the main attacks targeting the Falcon signature scheme and the implications of incorporating MCMC algorithms into the system.

5.7.1. Key Recovery

Lattice reduction is where attacks are most effective. The lattice generated by the columns of $\begin{bmatrix} q & h \\ 0 & 1 \end{bmatrix}$ serves as a starting point. After lattice redaction on this basis, all lattice points centred on the origin in a ball of radius $\sqrt{2n} \cdot \sigma_{\{f,g\}}$ are enumerated. Therefore, $[g \mid f]$ can be found with significant probability.

Let Λ represent the $(2n - B)$th Gram-Shmidt norm, which is approximately the norm of the shortest vector of the lattice generated by the last B vectors projected orthogonally to the first $2n - B - 1$ vectors. A sieve method applied to this projected lattice will retrieve any vectors with a norm less than $\sqrt{4/3}\Lambda$ (see [165] for instance). If the projection of the key is among them, that is when

$$\sqrt{B}\sigma_{\{f,g\}} \leq \sqrt{4/3}\Lambda,$$

a secret key vector can be retrieved with high probability from its projection using Babai's Nearest Plane algorithm on all sieved vectors. This is because all remaining Gram-Shmidt norms are larger than Λ, which is much larger than $\sigma_{\{f,g\}}$.

Using DBKZ [166] lattice reduction algorithm, the following is obtained:

$$\Lambda = \left(\frac{B}{2\pi e}\right)^{1-n/B} \sqrt{q},$$

and

$$(B/2\pi e)^{1-n/B}\sqrt{q} = \sqrt{3/4B}\sigma_{\{f,g\}} \tag{5.9}$$

5.7.2. Forging a signature

To forge a signature, one can find a lattice point in the same lattice as in the previous section at a distance restricted by β from a random point. Lattice reduction can also be used to accomplish this task. Using Kannan's embedding, which involves adding $(H(r||m), 0, K)$ to the lattice basis and extending it by a row of zeroes is one option. This matrix is as follows:

$$\begin{bmatrix} q & h & H(r||m) \\ 0 & 1 & 0 \\ 0 & 0 & K \end{bmatrix}.$$

As sieve methods generate several short vectors, it is possible to identify a vector of the form $(c,*,K)$ among them; hence, $H(r||m) - c$ is a lattice point.

Using $K \approx \sqrt{q}$, the DBKZ algorithm [166] determines a success condition for the forgery as follows:

$$\left(\frac{B}{2\pi e}\right)^{n/B} \sqrt{q} \leq \beta \tag{5.10}$$

Interestingly, since the factor $\sqrt{q}$ is also present in β, the modulus q has almost no impact on the most effective forgery attack. This is the most effective attack against the instances. The

blocksize B is converted into concrete bit-security following the methodology of New Hope [167], often known as “core-SVP methodology”. This provides the bit-security according to [168], [169]:

$$\text{Classical: } \lfloor 0.292 \cdot B \rfloor \tag{5.11}$$

$$\text{Quantum: } \lfloor 0.292 \cdot B \rfloor \tag{5.12}$$

Table 5.4 shows bit-security for the original Falcon and MCMC Falcon for n = 512 and n = 1024. As observed in the table, minimizing the standard deviation σ by integrating MCMC in Falcon results in higher bit-security, making the system more secure.

Table 5.4: Comparison of bit-security of vanilla Falcon, IMHK Falcon and SMK Falcon.

Signature Schema	σ	**Forgery**		
		Blocksize B	**Bit-Security**	
			Classical	**Quantum**
Falcon-512	165.74	411	120	108
Falcon-1024	168.39	952	277	249
Falcon-512 with IMHK	75	566	165	148
	70	583	170	152
	65	603	176	157
Falcon-1024 with IMHK	80	1221	356	319
	73	1263	368	330
	68	1298	379	340
Falcon-512 with SMK	60	626	183	164

Falcon-1024 with SMK	60	1362	398	357

As demonstrated in Table 5.4, integrating MCMC algorithms, such as IMHK and SMK, into Falcon results in a more secure system with higher bit-security levels for both classical and quantum environments. This improvement in security is achieved by minimizing the standard deviation σ in the signature generation process, which leads to a more robust resistance against known attacks, including key recovery and signature forgery.

5.8. Key Findings

This exploration into integrating MCMC algorithms within the Falcon signature scheme has unearthed pivotal insights into enhanced security in lattice-based post-quantum cryptography. The incorporation of MCMC algorithms, such as IMHK and SMK, has proven to elevate the security resilience of the Falcon signature scheme, as delineated in Table 5.4.

A closer analysis of Table 5.4 demonstrates that the modified Falcon-512 with IMHK, having σ=75, yielded a classical bit-security of 165 and quantum bit-security of 148, revealing a notable enhancement from the 120 and 108, respectively, found in vanilla Falcon-512. This enhancement is not confined to Falcon-512 but also pervades Falcon-1024 with IMHK and SMK, suggesting a comprehensive improvement across varied configurations.

The refined examination of the bit-security figures, depicted in the same table, reveals a discernible escalation in security levels due to the reduction in standard deviation σ through the integration of MCMC in Falcon. For instance, the elevated classical and quantum bit-security levels in Falcon-1024 with IMHK at σ=80 indicate the development of a more fortified and secure system compared to its original configurations.

Beyond mere numbers, this increased bit-security asserts Falcon's fortified stance against potential attacks, making it more impervious to threats like key recovery and signature forgery. The meticulous recalibration of signature generation processes through the assimilation of MCMC algorithms amplifies the security quotient against known threats and minimizes the susceptibilities to unforeseen vulnerabilities by reducing the standard deviation.

Moreover, the exploration unearthed a nuanced balance between security and performance. While there is an unavoidable increment in signature generation time, evident from the earlier tables, it is crucially counterbalanced by the heightened security levels. This equilibrium between elevated security parameters and maintained operational efficiency marks a significant stride in optimizing lattice-based cryptographic systems.

The implications of these findings are profound, shaping the discourse in lattice-based post-quantum cryptography. They provide a foundational framework for further research into innovative MCMC algorithms and their incorporation into diverse cryptographic systems for enhanced security robustness and offer a refined lens through which the future trajectory of post-quantum cryptography can be envisioned and shaped.

Conclusively, the comprehensive dissection and discussion of the insights and data, particularly those rendered in Table 5.4, substantiate the elevated security through the nuanced integration of MCMC algorithms in the Falcon signature scheme. These findings provide crucial foundational knowledge and ignite a discourse that will be instrumental in spearheading innovations and advancements in post-quantum cryptography, keeping pace with the advancements in quantum computation.

5.9. Summary

This chapter presented the integration of MCMC algorithms into the Falcon signature scheme to enhance its security and performance. The incorporation of MCMC sampling as a trapdoor sampler in Falcon was explored, along with the design and analysis of the MCMC-based algorithms, namely IMHK and SMK algorithms.

The implementation of these algorithms in the Falcon signature scheme was discussed, detailing the necessary modifications in the signature generation process. The performance evaluation and analysis of the MCMC-based Falcon schemes were conducted, providing insights into the trade-offs between signature speed and security. Results demonstrated that the MCMC-based Falcon schemes can maintain acceptable signature speeds while improving security by reducing the standard deviation σ of the trapdoor sampler.

A comprehensive security assessment was carried out, focusing on the resistance of the MCMC-based Falcon schemes to known attacks, such as key recovery and signature forgery attacks. The impact of security parameters, specifically the standard deviation σ, on the system's security was also analysed. The findings revealed that the MCMC-based Falcon schemes exhibit increased bit-security, making them more robust against attacks when compared to the original Falcon scheme.

In conclusion, the incorporation of MCMC algorithms into the Falcon signature scheme has been shown to offer potential advantages in terms of security and performance. The exploration of the MCMC-based Falcon schemes presented in this chapter contributes to the ongoing research in lattice-based post-quantum cryptography, offering valuable insights and possible directions for future work. Future research may involve developing and evaluating additional MCMC algorithms, exploring other lattice-based cryptographic systems, and investigating potential optimizations and novel applications of MCMC techniques in cryptography.

Chapter 6 Conclusion and Future Works

6.1. Thesis Summary

This thesis has focused on integrating post-quantum cryptography into blockchain technology to enhance IoT security. The research began with a comprehensive review of blockchain technology, IoT, and post-quantum cryptography. Various cryptographic schemes and their performance were compared to identify the most suitable solution for integration with blockchain platforms.

Subsequently, a post-quantum Hyperledger Fabric blockchain was proposed for the IoT domain, and its implementation and evaluation were discussed in detail. The research also explored using MCMC sampling as a trapdoor sampler in the Falcon signature scheme. The design, analysis, and implementation of MCMC-based algorithms were presented, along with their performance evaluation and security assessment.

Overall, the research has demonstrated the potential of post-quantum cryptography in securing blockchain technology for IoT applications. The proposed post-quantum Hyperledger Fabric blockchain has shown promising results in terms of performance and security. Furthermore, integrating MCMC sampling in the Falcon signature scheme has provided additional security advantages while maintaining acceptable performance.

6.1.1. Research Problems

This research addressed the paramount concerns surrounding the security vulnerabilities in existing cryptographic algorithms within blockchain technology with the rise and advancements in quantum computing. The pivotal research problems revolved around the notion that the current cryptographic underpinnings of blockchain, primarily based on classical algorithms, are susceptible to quantum attacks. Thus, it presents a significant challenge in securing data and transactions, especially in the IoT domain, where security is critical.

6.1.1.1. Recapitulation of Research Problems

- **Vulnerability of Classical Cryptography:** Classical cryptographic algorithms, which form the backbone of blockchain security, are theoretically breakable using sufficiently advanced quantum computers, exposing blockchain networks to potential security risks.
- **Integration of Post-Quantum Cryptography with Blockchain:** Integrating advanced post-quantum cryptographic algorithms with blockchain technology poses theoretical and practical challenges, particularly in varied blockchain platforms and architectures.
- **Balancing Security and Efficiency:** Integrating more secure post-quantum cryptographic algorithms can affect the efficiency and performance of blockchain networks, requiring a delicate balance to maintain practicality and user experience.
- **Scalability and Implementation in IoT:** Developing scalable solutions and understanding the practical implementation aspects of integrating post-quantum cryptographic algorithms within blockchain for IoT applications is a significant problem, given the unique requirements and constraints of IoT environments.
- **Long-term Security Assurance:** Ensuring the long-term security of post-quantum cryptographic schemes against future advancements in quantum computing is crucial for developing sustainable and resilient blockchain networks.

6.1.1.2. Addressing the Problems

- **Development of Quantum-Resistant Solutions:** The research presented a post-quantum Hyperledger Fabric blockchain, providing insights and practical solutions to address the vulnerabilities in classical cryptographic algorithms.
- **Innovative Integration Approaches:** The research has addressed the integration challenges by proposing and evaluating innovative approaches for embedding post-quantum cryptographic algorithms within blockchain infrastructures, focusing on IoT applications.
- **Optimization and Performance Evaluation:** By analysing and evaluating the MCMC-based algorithms in the Falcon signature scheme, this research shed light on the trade-offs between security and efficiency and provided insights into optimizing performance while maintaining enhanced security.

- **Foundational Work for Scalable Solutions:** The research provides foundational work and insights that can be built upon to address scalability and practical implementation aspects in large-scale IoT deployments.
- **Security Evaluation and Future Insights:** The rigorous analysis and evaluation of security aspects in the proposed solutions offer a pathway to understanding the long-term security implications of post-quantum cryptographic schemes and highlight areas for future research to ensure continued resilience against quantum advancements.

6.1.2. Research Contributions

1. **Approach for assessing efficiency and operational attributes of post-quantum algorithms within blockchain environments:** This research ventured into uncharted territories by establishing a pioneering approach for evaluating the efficiency and operational characteristics of post-quantum algorithms in blockchain settings. This methodology is a critical cornerstone in delineating these algorithms' computational and memory requisites, enabling a refined and extensive understanding and paving the way for optimized integration in diverse blockchain environments. This novel approach is pivotal, enhancing the field's readiness for future quantum computing and promoting robust scrutiny and refinement of current post-quantum blockchain integrations.
2. **Development of a quantum-resistant Hyperledger Fabric framework for IoT:** To fortify the synergy of IoT and blockchain against prospective quantum threats, a state-of-the-art framework was crafted, offering a quantum-resistant Hyperledger Fabric tailored to IoT domains. This construct meticulously mitigates the intricate vulnerabilities stemming from the amalgamation of IoT and blockchain in the face of potential quantum advances, placing paramount importance on crypto-agility to warrant seamless transitions to quantum-resistant states. This contribution epitomizes innovation by facilitating an array of algorithmic choices and leveraging the MQTT protocol for data integrity during transmission, mitigating the risks and reflecting the heterogeneous needs inherent to IoT infrastructures.
3. **Method to reinforce the Falcon post-quantum signature scheme using MCMC sampling:** A method was introduced to bolster the Falcon post-quantum signature scheme by amalgamating it with MCMC sampling. This integration diminishes the inherent variance in Falcon's discrete Gaussian trapdoor sampler, reducing the standard

deviation and introducing a pioneering enhancement in cryptographic resilience against quantum assaults. This method expands the horizons for subsequent research and development endeavours, focusing on augmenting cryptographic robustness in the face of burgeoning quantum threats.

The convergence of the research contributions detailed herein illuminates the intricate and multifaceted tapestry of the research journey. They offer a beacon of innovation and insight into post-quantum cryptography and blockchain within the IoT sphere, providing seminal frameworks, strategies, and methodologies that navigate and address the complex challenges inherent to these domains. The synergistic impact of these contributions is not confined to their inherent innovative merit but extends to shaping future research trajectories and practical implementations, fostering advancements in post-quantum cryptography, blockchain technology, and IoT applications. They stand as a testament to the transformative potential of integrated and secure technologies in mitigating evolving threats and optimizing operational efficiency in the era of quantum computing.

6.1.3. Limitations

- **Specific Blockchain Platform.** This research primarily focused on integrating post-quantum cryptography within the Hyperledger Fabric blockchain platform, which could be considered a limitation considering the extensive array of available blockchain platforms. Addressing this limitation requires future research initiatives to explore the integration of post-quantum cryptographic schemes in various blockchain platforms, such as Ethereum, Corda, and others, to expand the breadth of knowledge and applications in this realm.
- **Algorithmic Focus.** The investigation primarily delved into specific post-quantum algorithms and MCMC sampling, potentially overlooking other viable algorithms and sampling methods. A more expansive approach, embracing a variety of algorithms and sampling techniques, would furnish a richer, more comprehensive insight into the possibilities for strengthening post-quantum cryptographic schemes.
- **Real-world Scalability.** While foundational and comprehensive, the research did not extend to evaluating the scalability of the proposed solutions in large-scale, real-world IoT systems. Future research endeavours should prioritize empirical studies focusing on deploying the proposed blockchain solutions in extensive and diverse IoT

environments to validate their scalability, practicality, and adaptability to evolving technological landscapes.

- **Long-Term Security Analysis.** While the research provided significant insights into the security implications of integrating post-quantum cryptography, it did not thoroughly delve into the long-term security ramifications of these cryptographic schemes in the face of advancing quantum computing capabilities. Longitudinal studies and continuous security assessments are essential to comprehend the evolving security landscape and to refine and fortify cryptographic schemes against emerging quantum threats.
- **Consensus Algorithm Impact.** This research did not scrutinize the impact of post-quantum cryptography on the consensus algorithms integral to blockchain platforms. The interplay between cryptographic schemes and consensus mechanisms is crucial to blockchain systems' overall functionality and security. Subsequent research must examine how post-quantum cryptographic schemes interact with and influence various consensus algorithms to optimize security and efficiency.

6.2. Future Works

While this thesis has provided valuable insights into integrating post-quantum cryptography in the blockchain for IoT applications, several areas of future research can be explored further to enhance the security and performance of such systems. These include:

- Extending the study to other blockchain platforms: The current research focused on the Hyperledger Fabric blockchain platform. Investigating the integration of post-quantum cryptography in other blockchain platforms, such as Ethereum or Corda, could provide a broader understanding of the applicability and limitations of post-quantum cryptography in various blockchain systems.
- Developing more efficient MCMC-based algorithms: While the proposed MCMC-based algorithms have shown potential in improving the security of the Falcon signature scheme, further research is needed to optimize their performance and explore other MCMC techniques for enhancing security.
- Studying the impact of post-quantum cryptography on consensus algorithms: This research did not delve into the impact of post-quantum cryptographic schemes on

consensus algorithms used in blockchain platforms. Future work could investigate how post-quantum cryptography might affect the efficiency and security of consensus algorithms.

- Exploring the scalability of post-quantum blockchain solutions in large-scale IoT systems: The proposed post-quantum Hyperledger Fabric blockchain should be evaluated in large-scale IoT deployments to determine its feasibility in real-world applications.
- Evaluating the long-term security of post-quantum cryptography: As quantum computing technology advances, it is essential to continuously assess the long-term security of post-quantum cryptographic schemes to ensure their effectiveness in protecting blockchain-based IoT systems against quantum attacks.

References

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” *www.bitcoin.org*, vol. 15, no. 4, pp. 580–596, 2020, doi: 10.1108/TG-06-2020-0114.

[2] M. Swan, *Blockchain: Blueprint for a new economy*. O’Reilly Media, Inc., 2015.

[3] G. Zyskind, O. Nathan, and A. S. Pentland, “Decentralizing privacy: Using blockchain to protect personal data,” *Proc. - 2015 IEEE Secur. Priv. Work. SPW 2015*, pp. 180–184, 2015, doi: 10.1109/SPW.2015.27.

[4] P. Giungato, R. Rana, A. Tarabella, and C. Tricase, “Current trends in sustainability of bitcoins and related blockchain technology,” *Sustain.*, vol. 9, no. 12, 2017, doi: 10.3390/su9122214.

[5] C. Decker and R. Wattenhofer, “Information propagation in the Bitcoin network,” *13th IEEE Int. Conf. Peer-to-Peer Comput. IEEE P2P 2013 - Proc.*, 2013, doi: 10.1109/P2P.2013.6688704.

[6] T. M. Fernandez-Carames and P. Fraga-Lamas, “A Review on the Application of Blockchain to the Next Generation of Cybersecure Industry 4.0 Smart Factories,” *IEEE Access*, vol. 7, pp. 45201–45218, 2019, doi: 10.1109/ACCESS.2019.2908780.

[7] P. Fraga-Lamas and T. M. Fernández-Caramés, “A Review on Blockchain Technologies for an Advanced and Cyber-Resilient Automotive Industry,” *IEEE Access*, vol. 7, pp. 17578–17598, 2019, doi: 10.1109/ACCESS.2019.2895302.

[8] A. Vishwakarma, M. Kumar, and M. Arvindhan, “Blockchain in Smart Healthcare Application,” *Lect. Notes Mech. Eng.*, no. February, pp. 431–440, 2022, doi: 10.1007/978-981-19-0296-3_39.

[9] F. Xiong, R. Xiao, W. Ren, R. Zheng, and J. Jiang, “A key protection scheme based on secret sharing for blockchain-based construction supply chain system,” *IEEE Access*, vol. 7, pp. 126773–126786, 2019, doi: 10.1109/ACCESS.2019.2937917.

[10] T. M. Fernández-Caramés, O. Blanco-Novoa, I. Froiz-Míguez, and P. Fraga-Lamas, “Towards an Autonomous Industry 4.0 Warehouse: A UAV and Blockchain-Based System for Inventory and Traceability Applications in Big Data-Driven Supply Chain Management,” *Sensors (Basel).*, vol. 19, no. 10, 2019, doi: 10.3390/s19102394.

[11] W. S. Melo, A. Bessani, N. Neves, A. O. Santin, and L. F. R. C. Carmo, “Using Blockchains to Implement Distributed Measuring Systems,” *IEEE Trans. Instrum. Meas.*, vol. 68, no. 5, pp. 1503–1514, 2019, doi: 10.1109/TIM.2019.2898013.

[12] S. Gao, D. Zheng, R. Guo, C. Jing, and C. Hu, “An anti-quantum e-voting protocol in blockchain with audit function,” *IEEE Access*, vol. 7, pp. 115304–115316, 2019, doi: 10.1109/ACCESS.2019.2935895.

[13] T. M. Fernández-Caramés and P. Fraga-Lamas, “A Review on the Use of Blockchain for the Internet of Things,” *IEEE Access*, vol. 6, pp. 32979–33001, 2018, doi: 10.1109/ACCESS.2018.2842685.

[14] H. Y. Shwe, T. K. Jet, and P. H. J. Chong, “An IoT-oriented data storage framework in smart city applications,” *2016 Int. Conf. Inf. Commun. Technol. Converg. ICTC 2016*, pp. 106–108, 2016, doi: 10.1109/ICTC.2016.7763446.

[15] A. Boudguiga *et al.*, “Towards better availability and accountability for IoT updates by means of a blockchain,” *Proc. - 2nd IEEE Eur. Symp. Secur. Priv. Work. EuroS PW 2017*, pp. 50–58, 2017, doi: 10.1109/EuroSPW.2017.50.

[16] J. Chow, Jerry M.; Dial, Oliver; Gambetta, “IBM Quantum breaks the 100-qubit processor barrier,” *IBM*, 2021. https://research.ibm.com/blog/127-qubit-quantum-processor-eagle (accessed Oct. 01, 2023).

[17] A. El Kaafarani, “Four Ways Quantum Computing Could Change The World,” *Forbes*, 2021. https://www.forbes.com/sites/forbestechcouncil/2021/07/30/four-ways-quantum-computing-could-change-the-world (accessed Oct. 01, 2023).

[18] M. Mosca, “Cybersecurity in a quantum world: will we be ready?,” *Nist*, no. April, pp. 13–16, 2015, [Online]. Available: csrc.nist.gov/groups/ST/post-quantum.../session8-mosca-michele.pdf.

[19] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, 1997, doi: 10.1137/S0097539795293172.

[20] C. E. Gengler, M. Rossi, W. Hui, and J. Bragge, “The Design Science Research Process : a Model for Producing and Presenting Information System Research,” in *In Proceedings of the first international conference on design science research in information systems and technology*, 2006, no. February, pp. 83–106, [Online]. Available: https://arxiv.org/abs/2006.02763.

[21] S. Gregor and A. R. Hevner, “Positioning and presenting design science research for maximum impact,” *MIS Q. Manag. Inf. Syst.*, vol. 37, no. 2, pp. 337–355, 2013, doi: 10.25300/MISQ/2013/37.2.01.

[22] P. Offermann, O. Levina, M. Schönherr, and U. Bub, “Outline of a design science

research process," *Proc. 4th Int. Conf. Des. Sci. Res. Inf. Syst. Technol. DESRIST '09*, no. June 2014, 2009, doi: 10.1145/1555619.1555629.

[23] G. Hileman, "State of Blockchain Q1 2016: Blockchain Funding Overtakes Bitcoin," *Coindesk*, 2016. https://www.coindesk.com/markets/2016/05/11/state-of-blockchain-q1-2016-blockchain-funding-overtakes-bitcoin/ (accessed Oct. 01, 2023).

[24] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A Survey," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 1–1, 2019, doi: 10.1109/jiot.2019.2920987.

[25] G. W. Peters and E. Panayi, "Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money," *New Econ. Wind.*, pp. 239–278, 2016, doi: 10.1007/978-3-319-42448-4_13.

[26] D. Johnson, A. Menezes, and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, 2001, doi: 10.1007/s102070100002.

[27] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of Blockchains in the Internet of Things: A Comprehensive Survey," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 2, pp. 1676–1717, 2019, doi: 10.1109/COMST.2018.2886932.

[28] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telemat. Informatics*, vol. 36, no. May 2018, pp. 55–81, 2019, doi: 10.1016/j.tele.2018.11.006.

[29] U. Bodkhe *et al.*, "Blockchain for Industry 4.0: A comprehensive review," *IEEE Access*, vol. 8, pp. 79764–79800, 2020, doi: 10.1109/ACCESS.2020.2988579.

[30] A. Narayanan and J. Clark, "Bitcoin's Academic Pedigree," *Queue*, vol. 15, no. 4, pp. 20–49, 2017, doi: 10.1145/3134434.3136559.

[31] V. V. Buterin, "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.," *Whitepaper*, no. January, pp. 1–36, 2014, [Online]. Available: https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf.

[32] A. Hughes, A. Park, J. Kietzmann, and C. Archer-Brown, "Beyond Bitcoin: What blockchain and distributed ledger technologies mean for firms," *Bus. Horiz.*, vol. 62, no. 3, pp. 273–281, 2019, doi: 10.1016/j.bushor.2019.01.002.

[33] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving

framework for access control and interoperability of electronic health records using blockchain technology," *Sustain. Cities Soc.*, vol. 39, no. December 2017, pp. 283–297, 2018, doi: 10.1016/j.scs.2018.02.014.

[34] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," *Int. Conf. Adv. Commun. Technol. ICACT*, pp. 464–467, 2017, doi: 10.23919/ICACT.2017.7890132.

[35] HyperLedger, "Whitepaper Introduction Hyperledger," *July 2018*, 2018, [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf.

[36] S. K. Singh and V. R. Vadi, "Blockchain 1.0 to Blockchain 4.0—The Evolutionary Transformation of Blockchain Technology," in *Blockchain Technology: Applications and Challenges*, no. August, 2022, pp. 29–49.

[37] B. Glass, "Counterfeit drugs and medical devices in developing countries," *Res. Rep. Trop. Med.*, p. 11, 2014, doi: 10.2147/rrtm.s39354.

[38] U. D. of Health and H. Services, "Summary of the HIPAA Security Rule," *HIPAA*, 2018. https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html (accessed Oct. 01, 2023).

[39] GDPR, "Complete guide to GDPR compliance," *GDPR*, 2020. https://gdpr.eu/ (accessed Oct. 01, 2023).

[40] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, and I. Fellow, "DDoS in the IoT," *Computer (Long. Beach. Calif).*, vol. 50, no. 7, pp. 80–84, 2017, [Online]. Available: http://ieeexplore.ieee.org/document/7971869/.

[41] S. Sicari, A. Rizzardi, C. Cappiello, D. Miorandi, and A. Coen-Porisini, "Toward data governance in the internet of things," *Stud. Comput. Intell.*, vol. 715, no. May 2018, pp. 59–74, 2018, doi: 10.1007/978-3-319-58190-3_4.

[42] A. Ometov *et al.*, "An Overview on Blockchain for Smartphones: State-of-the-Art, Consensus, Implementation, Challenges and Future Trends," *IEEE Access*, vol. 8, pp. 103994–104015, 2020, doi: 10.1109/ACCESS.2020.2998951.

[43] R. H. Lasseter and P. Paigi, "Microgrid: A conceptual solution," *PESC Rec. - IEEE Annu. Power Electron. Spec. Conf.*, vol. 6, pp. 4285–4290, 2004, doi: 10.1109/PESC.2004.1354758.

[44] A. Cohn, T. West, and C. Parker, "Smart after all: Blockchain, smart contracts, parametric insurance, and smart energy grids.," 2017, [Online]. Available:

https://api.semanticscholar.org/CorpusID:208262411.

[45] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE Trans. Ind. Informatics*, vol. 14, no. 8, pp. 3690–3700, 2018, doi: 10.1109/TII.2017.2786307.

[46] N. Z. Aitzhan and D. Svetinovic, "Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams," *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 5, pp. 840–852, 2018, doi: 10.1109/TDSC.2016.2616861.

[47] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and V. Santamaría, "Blockchain and smart contracts for insurance: Is the technology mature enough?," *Futur. Internet*, vol. 10, no. 2, pp. 8–13, 2018, doi: 10.3390/fi10020020.

[48] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Inf. Syst. Front.*, vol. 17, no. 2, pp. 243–259, 2015, doi: 10.1007/s10796-014-9492-7.

[49] N. Mishra, P. Singhal, and S. Kundu, "Application of IoT products in smart cities of India," *Proc. 2020 9th Int. Conf. Syst. Model. Adv. Res. Trends, SMART 2020*, pp. 155–157, 2020, doi: 10.1109/SMART50582.2020.9337150.

[50] A. Nayyar, V. Puri, and D.-N. Le, "Internet of Nano Things (IoNT): Next Evolutionary Step in Nanotechnology," *Nanosci. Nanotechnol.*, vol. 7, no. 1, pp. 4–8, 2017, doi: 10.5923/j.nn.20170701.02.

[51] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.

[52] M. Chen, Y. Miao, Y. Hao, and K. Hwang, "Narrow Band Internet of Things," *IEEE Access*, vol. 5, pp. 20557–20577, 2017, doi: 10.1109/ACCESS.2017.2751586.

[53] O. Khutsoane, B. Isong, and A. M. Abu-Mahfouz, "IoT devices and applications based on LoRa/LoRaWAN," *Proc. IECON 2017 - 43rd Annu. Conf. IEEE Ind. Electron. Soc.*, vol. 2017-Janua, pp. 6107–6112, 2017, doi: 10.1109/IECON.2017.8217061.

[54] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and Privacy for Cloud-Based IoT: Challenges, Countermeasures, and Future Directions," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 26–33, 2017, [Online]. Available: http://ieeexplore.ieee.org/document/7823334/.

[55] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy

in distributed internet of things," *Comput. Networks*, vol. 57, no. 10, pp. 2266–2279, 2013, doi: 10.1016/j.comnet.2012.12.018.

[56] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016, doi: 10.1109/ACCESS.2016.2566339.

[57] Y. Zhang and J. Wen, "An IoT electric business model based on the protocol of bitcoin," *2015 18th Int. Conf. Intell. Next Gener. Networks, ICIN 2015*, pp. 184–191, 2015, doi: 10.1109/ICIN.2015.7073830.

[58] X. Xu *et al.*, "A Taxonomy of Blockchain-Based Systems for Architecture Design," *Proc. - 2017 IEEE Int. Conf. Softw. Archit. ICSA 2017*, pp. 243–252, 2017, doi: 10.1109/ICSA.2017.33.

[59] R. Grinberg, "Bitcoin: An Innovative Alternative Digital Currency Recommended Citation Bitcoin: An Innovative Alternative Digital Currency," *Hast. Sci. Technol. Law J.*, vol. 4, no. 1, p. 159, 2012.

[60] Vitalik Buterin, "Ethereum White Paper," *Ethereum*, 2014. https://ethereum.org/en/whitepaper/ (accessed Apr. 01, 2023).

[61] D. Schwartz, N. Youngs, and A. Britto, "The Ripple Protocol Consensus Algorithm," 2018, [Online]. Available: https://ripple.com/files/ripple_consensus_whitepaper.pdf.

[62] "ERIS." https://erisindustries.com/ (accessed Apr. 01, 2023).

[63] Richard Gendal Brown, "R3-Corda platform whitepaper," *Corda Platf. White Pap.*, pp. 1–21, 2018, [Online]. Available: https://www.corda.net/content/corda-platform-whitepaper.pdf.

[64] C. Pahl, N. El Ioini, and S. Helmer, "A decision framework for blockchain platforms for iot and edge computing," *IoTBDS 2018 - Proc. 3rd Int. Conf. Internet Things, Big Data Secur.*, vol. 2018-March, no. IoTBDS 2018, pp. 105–113, 2018, doi: 10.5220/0006688601050113.

[65] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Serv.*, vol. 14, no. 4, pp. 352–375, 2018, doi: 10.1504/IJWGS.2018.095647.

[66] A. Lymbouras, "A Shallow Dive Into Bitcoin's Blockchain Part 1 - Consensus," 2019. https://towardsdatascience.com/a-shallow-dive-into-bitcoins-blockchain-part-1-consensus-48f62355681b (accessed Oct. 01, 2023).

[67] Braiins, "How Much Would it Cost to 51% Attack Bitcoin?"

https://braiins.com/blog/how-muchwould-%0Ait-cost-to-51-attack-bitcoin (accessed Oct. 01, 2023).

[68] P. Vasin, “BlackCoin’s Proof-of-Stake Protocol v2 Pavel,” *Self-published*, p. 2, 2014, [Online]. Available: https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf.

[69] S. King and S. Nadal, “PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake,” 2012, [Online]. Available: https://people.cs.georgetown.edu/~clay/classes/fall2017/835/papers/peercoin-paper.pdf.

[70] M. Castro and B. Liskov, “Practical Byzantine Fault Tolerance,” in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, 1999, no. February.

[71] M. Vukolić, “The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9591, pp. 112–125, 2016, doi: 10.1007/978-3-319-39028-4_9.

[72] Q. Wang *et al.*, “Security Analysis on dBFT Protocol of NEO,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12059 LNCS, pp. 20–31, 2020, doi: 10.1007/978-3-030-51280-4_2.

[73] D. Mazieres, “The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus,” 2015, [Online]. Available: https://assets.website-files.com/5deac75ecad2173c2ccccbc7/5df2560fba2fb0526f0ed55f_stellar-consensus-protocol.pdf.

[74] S. Pahlajani, A. Kshirsagar, and V. Pachghare, “Survey on Private Blockchain Consensus Algorithms,” in *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, 2019, pp. 1–6, doi: 10.1109/ICIICT1.2019.8741353.

[75] L. Ismail and H. Materwala, “A Review of Blockchain Architecture and Consensus Protocols : Use Cases , Challenges , and Solutions,” no. August, pp. 1–45, 2019, doi: 10.20944/preprints201908.0311.v1.

[76] JP Morgan Chase, “Quorum Whitepaper,” *New York JP Morgan Chase*, pp. 1–8, 2016, [Online]. Available: https://quorum.com/resources/whitepaper.

[77] A. I. Sanka, M. Irfan, I. Huang, and R. C. C. Cheung, “A survey of breakthrough in blockchain technology: Adoptions, applications, challenges and future research,”

Comput. Commun., vol. 169, no. September 2020, pp. 179–201, 2021, doi: 10.1016/j.comcom.2020.12.028.

[78] E. Androulaki *et al.*, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," *Proc. 13th EuroSys Conf. EuroSys 2018*, vol. 2018-Janua, 2018, doi: 10.1145/3190508.3190538.

[79] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. M. Leung, "Decentralized Applications: The Blockchain-Empowered Software System," *IEEE Access*, vol. 6, pp. 53019–53033, 2018, doi: 10.1109/ACCESS.2018.2870644.

[80] K. Olson, M. Bowman, J. Mitchell, S. Amundson, D. Middleton, and C. Montgomery, "Sawtooth: An Introduction," *Hyperledger.Org*, no. January, pp. 1–7, 2018, [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2018/01/Hyperledger_Sawtooth_WhitePaper.pdf.

[81] V. Vlachou, C. Kontzinos, O. Markaki, P. Kokkinakos, V. Karakolis, and J. Psarras, "Leveraging Hyperledger Iroha for the Issuance and Verification of Higher-Education Certificates," *Int. J. Educ. Pedagog. Sci.*, vol. 14, no. 9, pp. 755–763, 2020, [Online]. Available: https://publications.waset.org/vol/165.

[82] N. Yushkevich, A. Lebedev, R. Šketa, and M. Takemiya, "D3ledger: The Decentralized Digital Depository platform for asset management based on Hyperledger Iroha," no. July, pp. 29–36, 2019, doi: 10.18690/978-961-286-282-4.4.

[83] P. Dunphy, "A Note on the Blockchain Trilemma for Decentralized Identity: Learning from Experiments with Hyperledger Indy," 2022, [Online]. Available: http://arxiv.org/abs/2204.05784.

[84] M. P. Bhattacharya, P. Zavarsky, and S. Butakov, "Enhancing the security and privacy of self-sovereign identities on hyperledger indy blockchain," *2020 Int. Symp. Networks, Comput. Commun. ISNCC 2020*, 2020, doi: 10.1109/ISNCC49221.2020.9297357.

[85] S. D. Palma, R. Pareschi, and F. Zappone, "What is your Distributed (Hyper)Ledger?," *Proc. - 2021 IEEE/ACM 4th Int. Work. Emerg. Trends Softw. Eng. Blockchain, WETSEB 2021*, no. March, pp. 27–33, 2021, doi: 10.1109/WETSEB52558.2021.00011.

[86] E. Elrom, *The Blockchain Developer*. Apress, 2019.

[87] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, no. 2, pp. 325–328, 1997, doi: 10.1103/PhysRevLett.79.325.

[88] A. M. Antonopoulos, *Mastering Bitcoin, 2nd Edition*. O'Reilly Media, Inc., 2017.

[89] J. J. Kearney and C. A. Perez-Delgado, "Vulnerability of blockchain technologies to

quantum attacks," *Array*, vol. 10, no. November 2020, p. 100065, 2021, doi: 10.1016/j.array.2021.100065.

[90] L. K. Grover, "A fast quantum mechanical algorithm for database search," 1996, [Online]. Available: https://arxiv.org/abs/quant-ph/9605043.

[91] D. A. Bard, J. J. Kearney, and C. A. Perez-Delgado, "Quantum advantage on proof of work," *Array*, vol. 15, 2022, doi: 10.1016/j.array.2022.100225.

[92] O. Vashchuk and R. Shuwar, "Pros and cons of consensus algorithm proof of stake. Difference in the network safety in proof of work and proof of stake," *Electron. Inf. Technol.*, vol. 9, no. 9, pp. 106–112, 2018, doi: 10.30970/eli.9.106.

[93] S. Micali and J. Chen, "Algorand," pp. 51–68, 2017, doi: 10.1145/3132747.3132757.

[94] S. Park, A. Kwon, G. Fuchsbauer, P. Gaži, J. Alwen, and K. Pietrzak, "SpaceMint: A Cryptocurrency Based on Proofs of Space," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10957 LNCS, pp. 480–499, 2018, doi: 10.1007/978-3-662-58387-6_26.

[95] R. Behnia, E. W. Postlethwaite, and M. O. Ozmen, "Lattice-Based Proof-of-Work for Post-Quantum Blockchains," 2020, [Online]. Available: https://eprint.iacr.org/2020/1362.

[96] D. P. Chi, J. W. Choi, J. S. Kim, and T. Kim, "Lattice Based Cryptography for Beginners," *ePrint*, 2015, [Online]. Available: https://eprint.iacr.org/2015/938.pdf.

[97] Z. Wang and C. Ling, "On the geometric ergodicity of metropolis-hastings algorithms for lattice Gaussian sampling," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 738–751, 2018, doi: 10.1109/TIT.2017.2742509.

[98] D. Micciancio, "Cryptographic functions from worst-case complexity assumptions," *Inf. Secur. Cryptogr.*, vol. 10, pp. 427–452, 2010, doi: 10.1007/978-3-642-02295-1_13.

[99] R. J. McEliece, "A Public-Key Cryptosystem Based On Algebraic Coding Theory," *The Deep Space Network Progress Report*, vol. 42, no. 44. pp. 114–116, 1978, [Online]. Available: http://ipnpr.jpl.nasa.gov/progress_report2/42-44/44title.htm.

[100] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, "On the Inherent Intractability of Certain Coding Problems," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 384–386, 1978, doi: 10.1109/TIT.1978.1055873.

[101] H. Niederreiter, "Knapsack-Type Cryptosystems," *Probl. Control Inf. Theory*, vol. 15, pp. 159–165, 1986.

[102] R. Overbeck and N. Sendrier, "Code-based cryptography," *Springer*, pp. 2–3, 1978, doi:

10.1007/978-3-540-88702-7_4.

[103] J. Ding and B.-Y. Yang, “Multivariate Public Key Cryptography BT - Post-Quantum Cryptography,” D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 193–241.

[104] A. Petzoldt, S. Bulygin, and J. Buchmann, “Selecting Parameters for the Rainbow Signature Scheme - Extended Version -,” *Cryptol. ePrint Arch.*, 2010, [Online]. Available: https://eprint.iacr.org/2010/437.

[105] J. Ding, A. Petzoldt, and L. Wang, “LNCS 8772 - The Cubic Simple Matrix Encryption Scheme,” *Post-Quantum Cryptogr. PQCrypto 2014. Lect. Notes Comput. Sci.*, vol. 8772, no. May, pp. 76–87, 2019, doi: 10.1007/978-3-319-11659-4.

[106] J. Ding, “A New Variant of the Matsumoto-Imai Cryptosystem through Perturbation A New Variant of the Matsumoto-Imai Cryptosystem through Perturbation,” *Public Key Cryptogr. – PKC 2004. PKC 2004. Lect. Notes Comput. Sci.*, vol. 2947, no. March 2004, pp. 305–318, 2015, doi: 10.1007/978-3-540-24632-9.

[107] J. Ding and D. Schmidt, “Cryptanalysis of HFEv and internal perturbation of HFE,” *Public Key Cryptogr. - PKC 2005. PKC 2005. Lect. Notes Comput. Sci.*, vol. 3386, pp. 288–301, 2005, doi: 10.1007/978-3-540-30580-4_20.

[108] L. Lamport, “Constructing Digital Signatures from a One-Way Function,” *SRI Int. Comput. Sci. Lab.*, vol. 94025, no. October, pp. 1–8, 1979, [Online]. Available: http://research.microsoft.com/en-us/um/people/lamport/pubs/dig-sig.pdf.

[109] R. C. Merkle, “Advances in Cryptology — CRYPTO’ 89 Proceedings,” vol. 435, no. June, pp. 175–185, 1990, doi: 10.1007/0-387-34805-0.

[110] J. Buchmann, E. Dahmen, and A. Hülsing, “XMSS - A practical forward secure signature scheme based on minimal security assumptions,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7071 LNCS, no. November, pp. 117–129, 2011, doi: 10.1007/978-3-642-25405-5_8.

[111] D. J. Bernstein *et al.*, “SPHINCS : practical stateless hash-based signatures,” in *Advances in Cryptology -- EUROCRYPT 2015. EUROCRYPT 2015. Lecture Notes in Computer Science()*, 2015, vol. 9056, pp. 368–397, doi: 10.1007/978-3-662-46800-5_15.

[112] E. D. Daniel J. Bernstein, Johannes Buchmann, Ed., *Post-Quantum Cryptography*. Springer Berlin, Heidelberg, 2008.

[113] S. S. Gill *et al.*, “Quantum computing: A taxonomy, systematic review and future

directions," *Softw. - Pract. Exp.*, vol. 52, no. 1, pp. 66–114, 2022, doi: 10.1002/spe.3039.

[114] J. Wang *et al.*, "Quantum-safe cryptography: crossroads of coding theory and cryptography," *Sci. China Inf. Sci.*, vol. 65, no. 1, 2022, doi: 10.1007/s11432-021-3354-7.

[115] A. Kumar, C. Ottaviani, S. S. Gill, and R. Buyya, "Securing the future internet of things with post-quantum cryptography," *Secur. Priv.*, vol. 5, no. 2, pp. 1–10, 2022, doi: 10.1002/spy2.200.

[116] S. Misra, A. Mukherjee, A. Roy, N. Saurabh, Y. Rahulamathavan, and M. Rajarajan, "Blockchain at the Edge: Performance of Resource-Constrained IoT Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 1, pp. 174–183, 2021, doi: 10.1109/TPDS.2020.3013892.

[117] L. Chen *et al.*, *NIST Report on Post-Quantum Cryptography*. 2016.

[118] J. Proos and C. Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves," *Quantum Inf. Comput.*, vol. 3, no. 4, pp. 317–344, 2003, doi: 10.26421/qic3.4-3.

[119] CNSS, "Use of Public Standards for the Secure Sharing of Information Among National Security Systems," *Comm. Natl. Secur. Syst.*, no. July, 2015, [Online]. Available: https://cryptome.org/2015/08/CNSS_Advisory_Memo_02-15.pdf.

[120] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and weaknesses of quantum computing," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1510–1523, 1997, doi: 10.1137/S0097539796300933.

[121] S. Krendelev and P. Sazonova, "Parametric hash function resistant to attack by quantum computer," *Proc. 2018 Fed. Conf. Comput. Sci. Inf. Syst. FedCSIS 2018*, vol. 15, pp. 387–390, 2018, doi: 10.15439/2018F254.

[122] G. Brassard, P. Høyer, and A. Tapp, "Quantum cryptanalysis of hash and claw-free functions," *ACM SIGACT News*, vol. 28, no. 2, pp. 14–19, 1997, doi: 10.1145/261342.261346.

[123] D. J. Bernstein, R. Niederhagen, A. Hülsing, J. Rijneveld, S. Kölbl, and P. Schwabe, "The SpHiNCS+ signature framework," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 2129–2146, 2019, doi: 10.1145/3319535.3363229.

[124] P. Fraga-Lamas, L. Ramos, V. Mondéjar-Guerra, and T. M. Fernández-Caramés, "A review on IoT deep learning UAV systems for autonomous obstacle detection and collision avoidance," *Remote Sens.*, vol. 11, no. 18, 2019, doi: 10.3390/rs11182144.

[125] T. M. Fernández-Caramés and P. Fraga-Lamas, "Towards the internet-of-smart-

clothing: A review on IoT wearables and garments for creating intelligent connected E-textiles," *Electron.*, vol. 7, no. 12, 2018, doi: 10.3390/electronics7120405.

[126] D. L. Hernández-Rojas, T. M. Fernández-Caramés, P. Fraga-Lamas, and C. J. Escudero, "Design and practical evaluation of a family of lightweight protocols for heterogeneous sensing through BLE beacons in IoT telemetry applications," *Sensors (Switzerland)*, vol. 18, no. 1, pp. 1–33, 2018, doi: 10.3390/s18010057.

[127] M. Suárez-Albela, T. M. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, "A practical evaluation of a high-security energy-efficient gateway for IoT fog computing applications," *Sensors (Switzerland)*, vol. 17, no. 9, pp. 1–39, 2017, doi: 10.3390/s17091978.

[128] S. Ebrahimi, S. Bayat-Sarmadi, and H. Mosanaei-Boorani, "Post-quantum cryptoprocessors optimized for edge and resource-constrained devices in IoT," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5500–5507, 2019, doi: 10.1109/JIOT.2019.2903082.

[129] T. M. Fernández-Caramés, P. Fraga-Lamas, M. Suárez-Albela, and L. Castedo, "Reverse engineering and security evaluation of commercial tags for RFID-based IoT applications," *Sensors (Switzerland)*, vol. 17, no. 1, 2017, doi: 10.3390/s17010028.

[130] P. Fraga-Lamas and T. M. Fernandez-Carames, "Reverse engineering the communications protocol of an RFID public transportation card," *2017 IEEE Int. Conf. RFID, RFID 2017*, pp. 30–35, 2017, doi: 10.1109/RFID.2017.7945583.

[131] M. Suárez-Albela, P. Fraga-Lamas, and T. M. Fernández-Caramés, "A practical evaluation on RSA and ECC-based cipher suites for iot high-security energy-efficient fog and mist computing devices," *Sensors (Switzerland)*, vol. 18, no. 11, 2018, doi: 10.3390/s18113868.

[132] M. Suárez-Albela, P. Fraga-Lamas, L. Castedo, and T. M. Fernández-Caramés, "Clock frequency impact on the performance of high-security cryptographic cipher suites for energy-efficient resource-constrained IoT devices," *Sensors (Switzerland)*, vol. 19, no. 1, 2019, doi: 10.3390/s19010015.

[133] P.-A. Fouque *et al.*, "Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU Specifications v1.2," pp. 1–65, 2020, [Online]. Available: https://falcon-sign.info/falcon.pdf.

[134] L. Ducas *et al.*, "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 0, no. 0, pp. 238–268, 2018, doi: 10.46586/tches.v2018.i1.238-268.

[135] M. Baldi, P. Santini, and G. Cancellieri, “Post-quantum cryptography based on codes: State of the art and open challenges,” *2017 AEIT Int. Annu. Conf. Infrastructures Energy ICT Oppor. Foster. Innov. AEIT 2017*, vol. 2017-Janua, pp. 1–6, 2017, doi: 10.23919/AEIT.2017.8240549.

[136] Y. Qassim, M. E. Magana, and A. Yavuz, “Post-quantum hybrid security mechanism for MIMO systems,” *2017 Int. Conf. Comput. Netw. Commun. ICNC 2017*, pp. 684–689, 2017, doi: 10.1109/ICCNC.2017.7876212.

[137] V. Clupek, L. Malina, and V. Zeman, “Secure digital archiving in post-quantum era,” *2015 38th Int. Conf. Telecommun. Signal Process. TSP 2015*, pp. 622–626, 2015, doi: 10.1109/TSP.2015.7296338.

[138] J. D. Preece and J. M. Easton, “Towards Encrypting Industrial Data on Public Distributed Networks,” *Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018*, pp. 4540–4544, 2019, doi: 10.1109/BigData.2018.8622246.

[139] B. C. & T. X. Ruping Shen, Hong Xiang, Xin Zhang, “Application and Implementation of Multivariate Public Key Cryptosystem in Blockchain,” in *Collaborative Computing: Networking, Applications and Worksharing. CollaborateCom 2019. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2019, pp. 419–428, doi: https://doi.org/10.1007/978-3-030-30146-0_29.

[140] M. Semmouni *et al.*, “Bitcoin Security with Post Quantum Cryptography,” *NETYS 2019*, p. 7, 2019, doi: 10.1007/978-3-030-31277-0_19.

[141] P. S. L. M. Barreto, P. Longa, M. Naehrig, J. E. Ricardini, and G. Zanon, “Sharper Ring-LWE Signatures,” *IACR Cryptol. ePrint Arch. 2016*, pp. 1–30, 2016, [Online]. Available: https://eprint.iacr.org/2016/1026.pdf.

[142] J. P. Aumasson, S. Neves, Z. Wilcox-O’Hearn, and C. Winnerlein, “BLAKE2: Simpler, smaller, fast as MD5,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7954 LNCS, pp. 119–135, 2013, doi: 10.1007/978-3-642-38980-1_8.

[143] “FIPS PUB 202 SHA-3 Standard : Permutation-Based Hash and,” *NIST Fed. Inf. Process. Stand.*, no. August, 2015, doi: 10.6028/NIST.FIPS.202.

[144] W. Yin, Q. Wen, W. Li, H. Zhang, and Z. Jin, “An anti-quantum transaction authentication approach in blockchain,” *IEEE Access*, vol. 6, pp. 5393–5401, 2017, doi: 10.1109/ACCESS.2017.2788411.

[145] “IOTA.” https://www.iota.org/ (accessed Mar. 22, 2023).

[146] S. Popov, “The Tangle,” *New Yorker*, vol. 81, no. 8, pp. 1–28, 2018, [Online]. Available: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf.

[147] The QRL, “QRL—The Quantum Resistant Ledger.” https://www.theqrl.org/ (accessed Mar. 22, 2023).

[148] C. Cachin, “Architecture of the Hyperledger Blockchain Fabric,” *IBM Res.*, 2016, doi: 10.4230/LIPIcs.OPODIS.2016.24.

[149] Hyperledger, “Five hyperledger blockchain projects now in production,” 2018. https://www.hyperledger.org/blog/2018/11/30/six-hyperledger-blockchain-projects-now-in-production.

[150] C. Cachin, “Architecture of the Hyperledger Blockchain Fabric,” 2016, doi: 10.4230/LIPIcs.OPODIS.2016.24.

[151] P. Kampanakis, P. Panburana, E. Daw, and D. Van Geest, “The Viability of Post-Quantum X.509 Certificates,” *IACR Cryptol. ePrint Arch.*, 2018, [Online]. Available: https://eprint.iacr.org/2018/063.pdf.

[152] A. O. Salman Baset, Luc Desrosiers, Nitin Gaur, Petr Novotny, Venkatraman Ramakrishna, *Hands-On Blockchain with Hyperledger*. Packt Publishing, 2018.

[153] D. Stebila and M. Mosca, “Post-Quantum Key Exchange for the Internet and the Open Quantum Safe Project ∗,” pp. 1–22, 2017.

[154] S. Pongnumkul, C. Siripanpornchana, and S. Thajchayapong, “Performance analysis of private blockchain platforms in varying workloads,” *2017 26th Int. Conf. Comput. Commun. Networks, ICCCN 2017*, 2017, doi: 10.1109/ICCCN.2017.8038517.

[155] Hyperledger, “Hyperledger Caliper.” https://www.hyperledger.org/projects/caliper (accessed Oct. 01, 2023).

[156] J. Jeong, D. Kim, S. Y. Ihm, Y. Lee, and Y. Son, “Multilateral Personal Portfolio Authentication System Based on Hyperledger Fabric,” *ACM Trans. Internet Technol.*, vol. 21, no. 1, 2021, doi: 10.1145/3423554.

[157] NIST, “Post-Quantum Cryptography Project.” https://csrc.nist.gov/projects/post-quantum-cryptography (accessed Oct. 01, 2023).

[158] D. Moody, “Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process,” *Nistir 8309*, pp. 1–27, 2022.

[159] C. Gentry, C. Peikert, and V. Vaikuntanathan, “How to Use a Short Basis: Trapdoors for Hard Lattices and New Cryptographic Constructions,” *Proc. 40th Annu. ACM Symp.*

Theory Comput., pp. 197–206, 2009.

[160] P. Klein, “Finding the closest lattice vector when it’s unusually close,” *Proc. Annu. ACM-SIAM Symp. Discret. Algorithms*, pp. 937–941, 2000.

[161] J. Hoffstein, J. Pipher, and J. H. Silverman, “NTRU: A ring-based public key cryptosystem,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1423, pp. 267–288, 1998, doi: 10.1007/bfb0054868.

[162] Z. Wang and C. Ling, “Lattice Gaussian Sampling by Markov Chain Monte Carlo: Bounded Distance Decoding and Trapdoor Sampling,” *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3630–3645, 2019, doi: 10.1109/TIT.2019.2901497.

[163] D. van Ravenzwaaij, P. Cassey, and S. D. Brown, “A simple introduction to Markov Chain Monte–Carlo sampling,” *Psychon. Bull. Rev.*, vol. 25, no. 1, pp. 143–154, 2018, doi: 10.3758/s13423-016-1015-8.

[164] A. W. K. Hastings, “Monte Carlo Sampling Methods Using Markov Chains and Their Applications Published by : Biometrika Trust Stable URL : http://www.jstor.org/stable/2334940,” *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970, [Online]. Available: https://academic.oup.com/biomet/article-abstract/57/1/97/284580.

[165] L. Ducas, “Shortest vector from lattice sieving: A few dimensions for free,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10820 LNCS, pp. 125–145, 2018, doi: 10.1007/978-3-319-78381-9_5.

[166] D. Micciancio and M. Walter, “Practical, predictable lattice basis reduction,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9665, pp. 820–849, 2016, doi: 10.1007/978-3-662-49890-3_31.

[167] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, “Post-quantum key exchange – A new hope,” *Proc. 25th USENIX Secur. Symp.*, pp. 327–343, 2016.

[168] A. Becker, L. Ducas, N. Gama, and T. Laarhoven, “New directions in nearest neighbor searching with applications to lattice sieving,” *Proc. Annu. ACM-SIAM Symp. Discret. Algorithms*, vol. 1, pp. 10–24, 2016, doi: 10.1137/1.9781611974331.ch2.

[169] T. Laarhoven, “Search problems in cryptography: From fingerprinting to lattice sieving,” vol. 1, no. 2016, 2016, [Online]. Available: https://research.tue.nl/en/publications/search-problems-in-cryptography-from-fingerprinting-to-lattice-si.