Solving Blockchain Trilemma Using Off-Chain Storage Protocol

by

Saha Reno

Roll No: 18MCSE005P

A thesis submitted for the partial fulfillment of the requirement for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering(CSE)

CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Chattogram-4349, Bangladesh

August, 2023

CERTIFICATION

The thesis titled **"Solving Blockchain Trilemma Using Off-Chain Storage Protocol"** submitted by **Saha Reno**, Roll No **18MCSE005P**, Session **2018-2019** has been accepted as satisfactory in partial fulfillment of the requirement for the degree of MASTER OF SCIENCE IN COMPUTER SCIENCE & ENGINEERING on 24/08/2023

BOARD OF EXAMINERS

1. _____ Dr. Md. Mokammel Haque Professor Department of Computer Science & Engineering Chittagong University of Engineering & Technology	Chairman (Supervisor)
2. _____ Dr. A.H.M. Ashfak Habib Professor & Head Department of Computer Science & Engineering Chittagong University of Engineering & Technology	Member (Ex-Officio)
3. _____ Dr. Mohammed Moshiul Hoque Professor Department of Computer Science & Engineering Chittagong University of Engineering & Technology	Member
4. _____ Dr. Md. Shariful Islam Professor Institute of Information Technology (IIT) University of Dhaka	Member (External)

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Author: Saha Reno .....
(signature)

Date: 24.08.2023

It is my genuine gratefulness and warmest regard that

I dedicate this work to my beloved

Father and Mother

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Terms and Abbreviations

IPFS	InterPlanetary File System
P2P	Peer-to-Peer
DES	Data Encryption Standard
AES	Advanced Encryption Standard
RSA	Rivest–Shamir–Adleman
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature
DAG	Directed Acyclic Graph
DHT	Distributed Hash Table
CID	Content Identifier
PoW	Proof-of-Work
UTB	Uniform Tie-Breaking
SHTB	Smallest-Hash Tie-Breaking
UDTB	Unpredictable Deterministic Tie-Breaking
PoP	Publish or Perish
RS	Reward-Splitting

Acknowledgement

I am overwhelmed to get this opportunity to express my gratitude to those whose constant support and encouragement lead me to the completion of my thesis on human robot interaction. First and foremost, I feel immense proud to express my heartfelt respect and deepest sense of gratitude to my supervisor, Prof. Dr. Md. Mokammel Haque for his scholarly guidance, instruction, constructive criticism, valuable suggestions, supervision, inspiration, untiring assistance in all phases of the research work and in the preparation of the report manuscript. Without his pedagogic supervision, heartfelt help, unfailing attention, and constructive criticism this report would not have seen the light of the day. I am ever grateful to the honorable examination board member for their valuable suggestion, helpful comment, and affectionate feeling.

I am incredibly grateful to all the faculty members and the staff of department of CSE, CUET for their support. My friends, colleagues and several students of CUET deserves special thanks for their assistance to make this work possible. I express my cordial respect and sincere appreciation to my parents for their unforgettable kind co-operation, constructive suggestions, sacrifice, and cordial support in all spheres of the working period.

Finally, all praises to God, the Almighty because it is He, who give me the strength and courage to accomplish the task and achieve the goal.

Abstract

Trilemma in blockchain refers to the infamous problem of simultaneously not delivering the three critical aspects of a ledger: security, scalability and decentralization. While security and scalability hinder decentralization, security is jeopardized if the scalability is escalated. This deficiency of not maintaining a balance among these three crucial factors restricts the broader adoption of blockchain technology and cryptocurrencies in the industries. This research proposes a solution to the blockchain trilemma by implementing a public ledger using The InterPlanetary File System (IPFS) and a newly introduced strategy called the double-chain technique. The scalability and decentralization features are guaranteed by the distributed file system of IPFS and the public nature of the blockchain suggested in this thesis. Although any consensus can be plugged into our system, the proof-of-work consensus is utilized to ensure that the security is not compromised while stabilizing scalability and decentralization.

Keywords: Blockchain, Throughput, Storage Bloating, Security, IPFS.

Chapter 1

Introduction

1.1 Overview of Blockchain

Blockchain is a distributed digital ledger in which the stored assets or transaction gets immutable and decentralized. As transactions are cryptographically secured and robust security is delivered by its strong consensus protocol, blockchain is desired and intended to be utilized in real-life applications where the classical systems fail to deliver prompt security. Blockchain is guaranteed to achieve greater efficiency, accuracy and safety in the case of governmental, public and social services compared to the existing systems.

Each block in a blockchain is linked cryptographically to the preceding one, containing a unique hash of the previous block and transactional data, ensuring the integrity of the entire chain. The working procedure of blockchain technology is depicted using Figure 1.1.

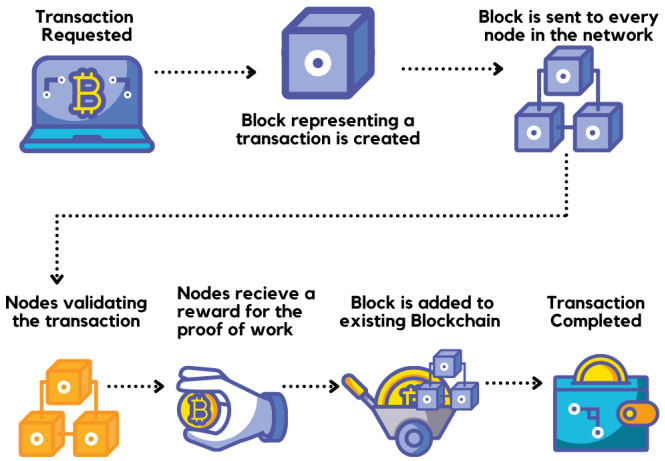


Figure 1.1. Systematic Workflow of Blockchain

1. **Blocks:** A block consists of a list of transactions, a timestamp, a reference to the previous block (via its hash), and sometimes other metadata. The combination of these elements ensures the authenticity and integrity of the information contained

within the block. The interconnection among the blocks using cryptographic has is illustrated in Figure 1.2.

2. **Decentralization:** Unlike traditional centralized databases, blockchain operates on a network of nodes. Each node has a copy of the entire blockchain, and transactions are verified by a consensus mechanism, often through a process called mining.

3. **Immutability:** Once a block is added to the blockchain, it becomes permanent and cannot be altered. This immutability is achieved through cryptographic principles and the collaborative verification process among the nodes.

4. **Transparency:** All transactions recorded on the blockchain are visible to anyone with access to the network. This transparency fosters accountability and makes fraud more difficult to perpetrate.

5. **Security:** The cryptographic linkage between blocks and the decentralized nature of the system make it highly resistant to malicious attacks. Even if one node is compromised, the integrity of the entire chain remains intact due to the distributed architecture.

6. **Use Cases:** Blockchain technology has found applications beyond cryptocurrencies, such as in supply chain management, healthcare, real estate, and governance. It offers a transparent and secure way to record, verify, and share data without the need for intermediaries.

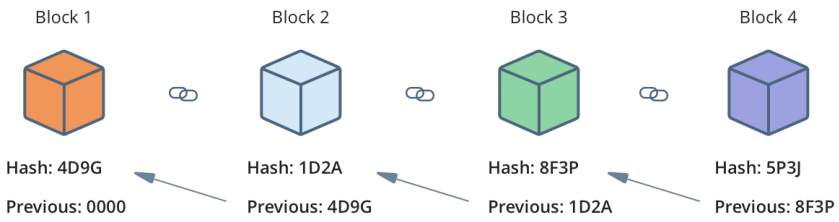


Figure 1.2. Chain of Blocks Cryptographically Linked to Each Other

1.2 Structure of Blocks

In a blockchain, blocks are the fundamental units that store the transactional information. Each block is meticulously constructed with several key elements that together ensure the security, integrity, and chronological order of the blockchain. Here's an in-depth look at the structure of a block:

1. **Previous Block Hash:** The unique digital signature of the preceding block. It acts as a cryptographic link in the blockchain, connecting the blocks in a sequential manner. This linkage ensures that tampering with one block would require altering all subsequent blocks, making unauthorized changes practically infeasible.

2. **Timestamp:** The exact time when the block was created or mined. The timestamp provides a chronological order to the blockchain, ensuring that blocks follow a linear timeline. It prevents double-spending and time-related fraud by firmly establishing the sequence of transactions.

3. **Nonce:** A random value used in the block's cryptographic process. The nonce is discovered through the proof-of-work system and is used to validate new blocks by solving a cryptographic puzzle. It adds an additional layer of complexity, ensuring that mining a block requires computational work, thereby securing the blockchain against spam and denial-of-service attacks.

4. **Current Block Hash:** A unique identifier of the current block, generated cryptographically. It is derived from the block's contents, including transaction data, the timestamp, the nonce, and the previous block hash. This hash acts as a fingerprint for the block, allowing for quick verification and ensuring the integrity of its contents.

5. **Block Data:** Refers to the specific transactional information or the payload that the block carries. This can include details such as sender, receiver, amount, and other relevant data pertaining to the transaction. Storing this information within the block allows for a transparent and immutable record of all transactions within the network, facilitating trust and verification. Blocks with their internal elements are graphically represented using Figure 1.3.

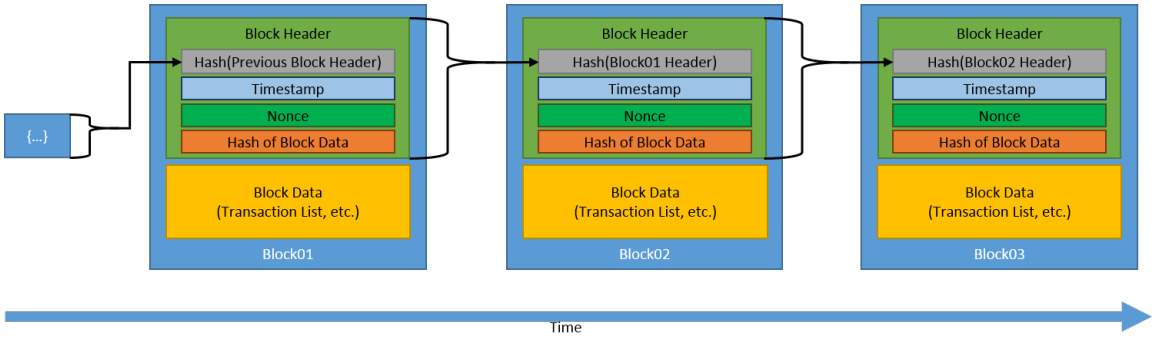


Figure 1.3. Internal Structure of Blocks in Blockchain

1.3 Applications of Blockchain

The revolutionary nature of blockchain technology has led to its adoption across a variety of industries and applications. By offering a decentralized, transparent, and immutable

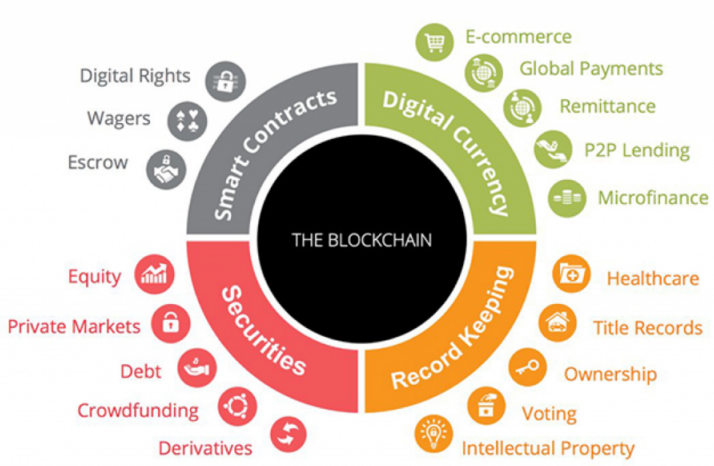


Figure 1.4. Application of Blockchain Based on Its Attributes

platform for transactional data, blockchain addresses many pain points traditionally associated with centralized systems. Here's a closer look at some of the key applications of blockchain technology:

1. **Decentralized Apps (dApps):** dApps are applications that run on a blockchain or P2P network of computers rather than a single computer. They are commonly used to deploy smart contracts, which are self-executing contracts with the terms of the agreement directly written into code. dApps and smart contracts enable trustless, automated transactions and agreements, eliminating the need for intermediaries and reducing potential fraud or discrepancies.

2. **Digital Currencies:** Digital currencies like Bitcoin and Ethereum operate on blockchain technology to ensure secure, transparent financial transactions. These currencies can be used for a wide range of transactions, from online purchases to investment assets. Digital currencies have revolutionized financial systems by enabling fast, secure, and borderless transactions.

3. **Securities:** Blockchain can be used for issuing and trading securities, bringing transparency and efficiency to the process. It can handle various types of securities like stocks, bonds, or derivatives. Blockchain in securities trading reduces the need for intermediaries, speeds up settlement times, and ensures transparency, thereby reducing fraud and increasing efficiency.

4. **Record Keeping:** Blockchain's immutable and transparent nature makes it an excellent tool for document management and record-keeping. It can be used for various records, including land registries, academic degrees, medical records, and more. Blockchain's ability to provide tamper-proof record-keeping can significantly im-

prove transparency and trust in document management systems. This classification is pictorially represented using Figure 1.4.

Blockchain technology holds immense potential across diverse sectors, including but not limited to:

- **Real Estate:** Blockchain can bring transparency and efficiency to property transactions, from proof of ownership to streamlining the buying and selling process.
- **Digital Currencies:** Blockchain can provide secure patient data management, ensuring patient privacy while enabling data sharing for improved care.
- **Securities:** Blockchain can provide a secure, immutable system for digital identity verification, reducing identity theft and fraud.
- **Record Keeping:** With blockchain, supply chains can become more traceable and efficient, reducing errors, improving product authenticity, and enabling better logistics management. Figure 1.5 depicts some of the examples of blockchain-based applications.

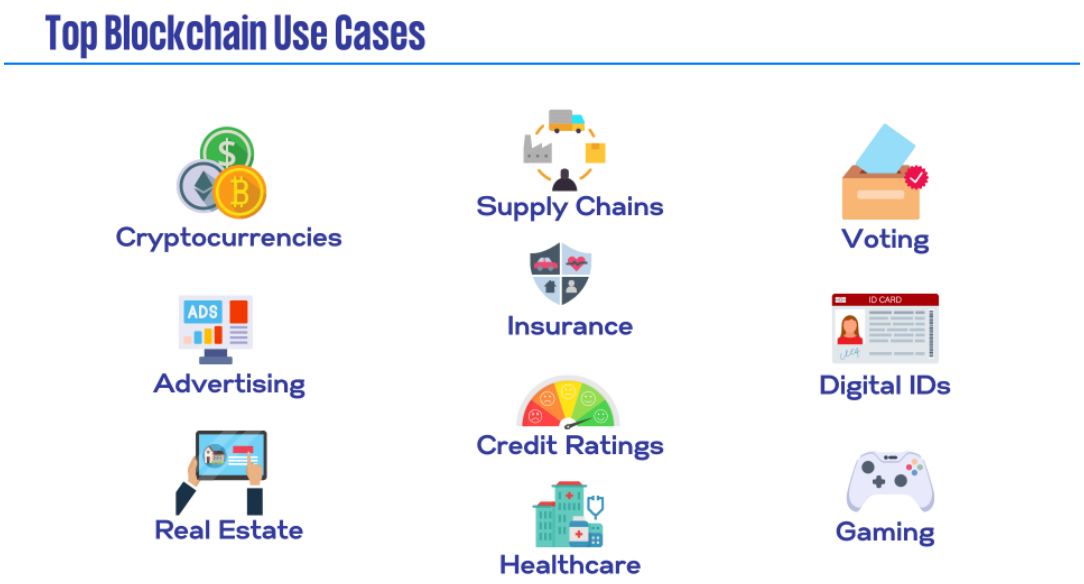


Figure 1.5. Various Applications of Blockchain

1.4 Genres of Blockchain

Blockchain technology can be categorized into several types, each with its unique characteristics, advantages, and applications. The choice among these types depends on the specific needs and trade-offs regarding decentralization, security, speed, and control. Here is an in-depth look into the four distinct types of blockchain:

1. **Public Blockchains:** Public blockchains are decentralized and open to anyone. They are permissionless, meaning anyone can participate in validating transactions and

maintaining the ledger. Bitcoin is a notable example of a public blockchain. Public blockchains are characterized by their high level of decentralization and security. They rely on consensus algorithms such as Proof of Work (PoW) or Proof of Stake (PoS) to validate transactions and create new blocks. Public blockchains are widely used for cryptocurrency transactions and decentralized applications (dApps). While offering high security and transparency, public blockchains can have scalability issues, leading to slower transaction times and higher costs.

2. **Private Blockchains:** Private blockchains are permissioned and centralized, limiting participation to specific entities. Control over the network is maintained within a single organization. Ripple (XRP) operates on a private blockchain. Private blockchains are faster and more efficient than public blockchains as they do not require extensive consensus mechanisms. They offer privacy for transactions while maintaining the benefits of blockchain technology. They are often used for business applications, where control, privacy, and speed are paramount. While providing efficiency and control, private blockchains lack the decentralization and wide trust that public blockchains offer.

3. **Hybrid Blockchains:** Hybrid blockchains combine elements of both public and private blockchains. They allow for public access in some areas while maintaining private control in others. IBM Food Trust is a hybrid blockchain solution. Hybrid blockchains offer the best of both worlds: the transparency and security of public blockchains with the control and efficiency of private blockchains. They are particularly useful in scenarios where some data needs to be public while other data needs to remain private. For example, in a supply chain, a company might want to share product tracking data with customers but keep manufacturing details private.

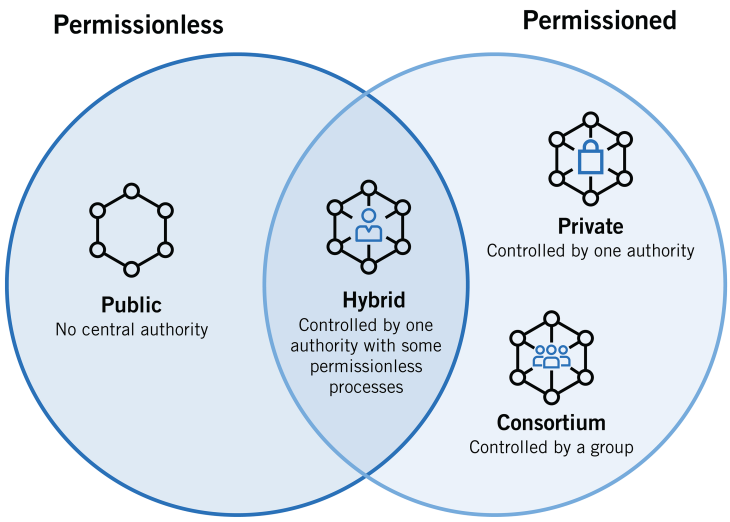


Figure 1.6. Types of Blockchain Systems

4. **Consortium Blockchains:** Consortium blockchains, also known as federated blockchains, are semi-private and operate under the leadership of a group instead of a single entity. Hyperledger is a well-known consortium blockchain. They offer a balance between the total openness of public blockchains and the closed control of private ones, with a group of pre-selected nodes achieving consensus on transactions. They are often used in the banking sector and other industries where direct control and privacy are needed, but a degree of cross-organizational transparency is beneficial. All of the aforementioned genres are illustrated in Figure 1.5.

1.5 Basics of Consensus

At the core of blockchain technology lies the concept of consensus. Blockchain consensus mechanisms are protocols that ensure all nodes in the network agree on the information recorded on the blockchain. These mechanisms are fundamental to providing a trustworthy, secure, and democratic means of validating transactions.

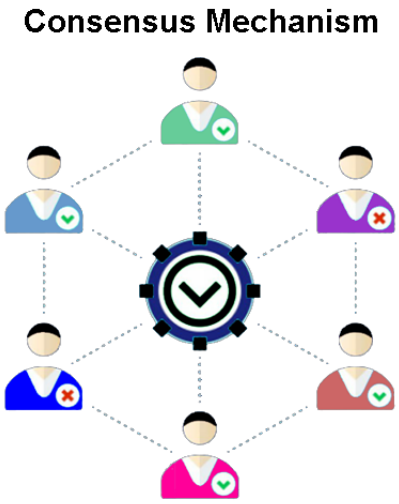


Figure 1.7. Basic Concept of Consensus in Blockchain

In the context of blockchain, consensus refers to the process through which all the nodes in a blockchain network agree on the current state of the distributed ledger. This agreement is crucial as it ensures each new block added to the chain is the one and only version of the truth accepted by all nodes. This definition of blockchain consensus is graphically represented in Figure 1.7. The main purposes of consensus mechanisms are to:

- **Ensure Consistency:** They guarantee that all nodes in the network have the same data.
- **Prevent Double Spending:** They ensure that a digital asset is spent only once by keeping a consistent ledger.

- **Security:** They make the network resilient against malicious attacks by distributing authority among multiple nodes.

There are various consensus mechanisms, each with its advantages and disadvantages, which are highlighted in Figure 1.8. The choice depends on the requirements of the specific blockchain system:

1. **Permissionless Consensus Protocols:** These protocols, used in public blockchains, allow any node to contribute to the validation process. Proof of Work (PoW) used by Bitcoin and Proof of Stake (PoS) used by Ethereum are examples of permissionless consensus protocols. They are open to anyone, highly secure, and fully decentralized, but can be resource-intensive (like PoW) or lead to centralization of power (like PoS in some cases).
2. **Permissioned Consensus Protocols:** These protocols, found in private blockchains, restrict the validation process to a select group of approved nodes. Practical Byzantine Fault Tolerance (PBFT) and Raft Consensus Algorithm are examples of permissioned consensus protocols. They are faster and more efficient but are less decentralized and may require a higher level of trust among participants.

Choosing the right consensus mechanism for a blockchain network depends on several factors, including:

- The level of control or decentralization required
- The need for security vs. speed and efficiency
- The type and number of participants in the network

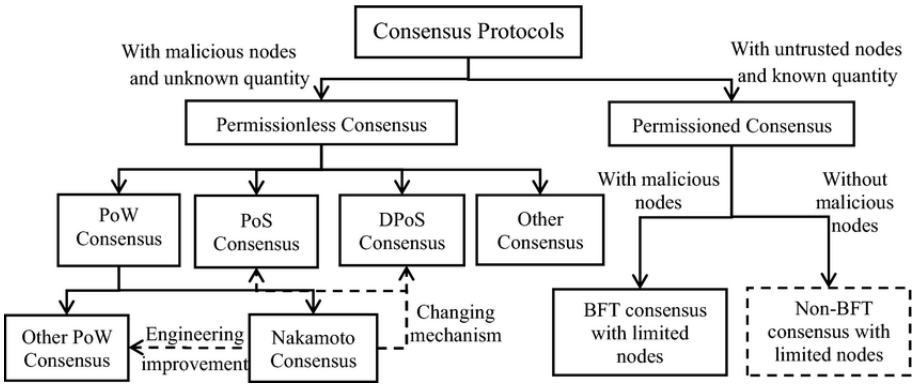


Figure 1.8. Types of Consensus in Blockchain

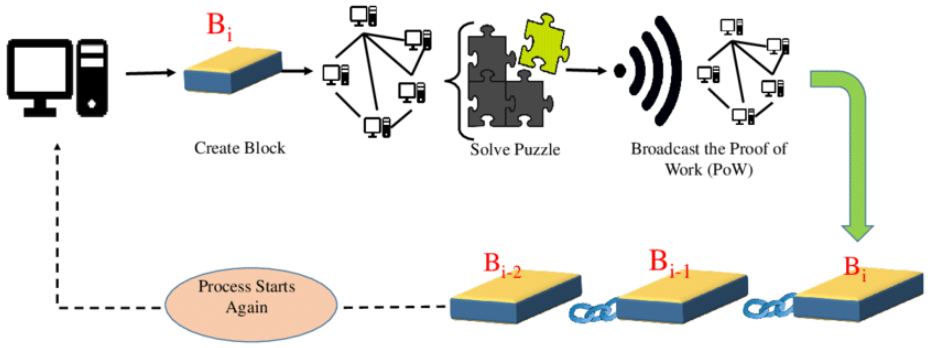


Figure 1.9. Steps in Proof-of-Work

1.6 Why Proof-of-Work?

Proof of Work (PoW) is a fundamental concept in the realm of blockchain technology. It serves as a crucial mechanism to validate and secure transactions on a decentralized network. At its core, Proof of Work is a consensus algorithm that requires participants, known as miners, to demonstrate computational effort to validate and add new blocks to the blockchain.

The process of PoW begins with the creation of a new block containing a set of pending transactions. To add this block to the blockchain, miners must compete to solve a complex mathematical puzzle known as the "hash puzzle." This puzzle is designed in a way that finding the solution is computationally demanding, and it requires a significant amount of computational power.

The goal of miners is to find a specific value (called a nonce) that, when combined with the data in the new block, results in a hash value that meets a certain predetermined criteria (such as starting with a certain number of leading zeros). The only way to find the correct nonce is through a trial-and-error process, where miners continuously make attempts until they find the correct value. This process is energy-intensive and time-consuming. The graphical representation of PoW is provided in Figure 1.9.

Why is Proof of Work considered an important consensus mechanism in the world of cryptocurrencies and blockchain networks?

1. **Immunity to Attacks:** Firstly, one of the primary advantages of PoW is its immunity to attacks. Due to the immense computational effort required to solve the hash puzzle, it becomes exceedingly challenging for adversaries to create multiple identities and execute a Sybil attack. A Sybil attack involves a malicious entity creating multiple fake identities to gain control over the network. With PoW, the cost of executing such an attack becomes prohibitively high, making the blockchain network more secure and reliable.

2. **Decentralized Consensus:** Secondly, PoW fosters decentralized consensus. Since the mining process is distributed among a network of miners, no single entity can control the majority of the computing power. This decentralized nature ensures that no central authority can dictate the fate of the blockchain. It empowers the community to participate in the decision-making process and ensures a fair and democratic validation of transactions.

3. **Enhanced Security:** Furthermore, the strength of security provided by PoW is a significant reason for its continued use. The computational work required for mining new blocks acts as a deterrent against potential attackers. Any malicious entity attempting to manipulate the blockchain would need to control a substantial portion of the total computational power, which is both technically challenging and economically impractical.

4. **Consortium Blockchains:** Consortium blockchains, also known as federated blockchains, are semi-private and operate under the leadership of a group instead of a single entity. Hyperledger is a well-known consortium blockchain. They offer a balance between the total openness of public blockchains and the closed control of private ones, with a group of pre-selected nodes achieving consensus on transactions. They are often used in the banking sector and other industries where direct control and privacy are needed, but a degree of cross-organizational transparency is beneficial.

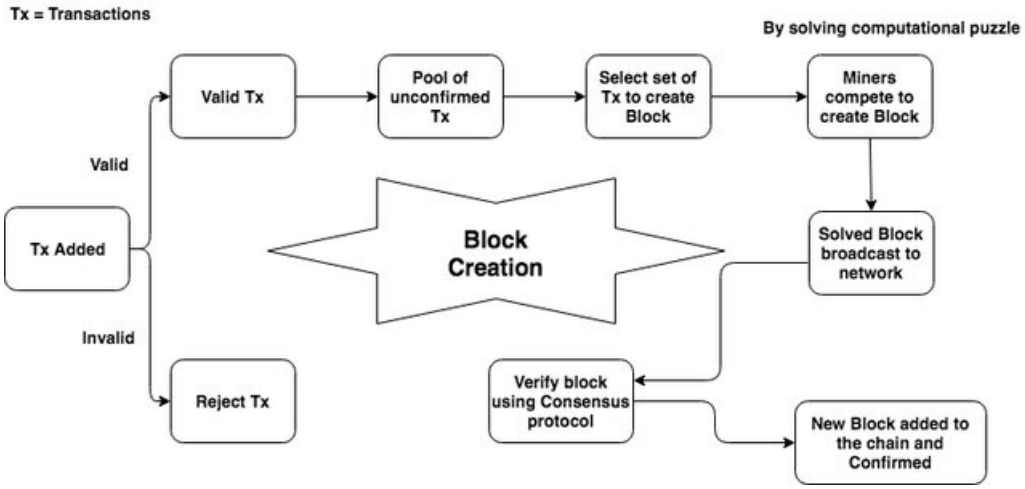


Figure 1.10. Working Methodology Proof-of-Work

However, it is worth mentioning that PoW has its limitations. The high energy consumption associated with the computational work has raised environmental concerns. As the demand for cryptocurrencies and blockchain technology grows, so does the energy consumption of PoW-based networks. This has led to the exploration of alternative consensus mechanisms,

such as Proof of Stake (PoS), which aim to provide security and validation without the same level of energy consumption. Figure 1.10 represents the flow diagram for PoW methodology.

1.7 Generations of Blockchain

Blockchain technology has captured the world's attention as one of the most transformative innovations of the digital age. As it evolves, distinct generations of blockchain have emerged, each boasting unique attributes that contribute to its widespread adoption and potential impact on various industries. Based on the applicability and utilization type, blockchain can be categorized into three different generations: Blockchain 1.0, 2.0 and 3.0 [1]. These three generations are graphically represented in Figure 1.11.

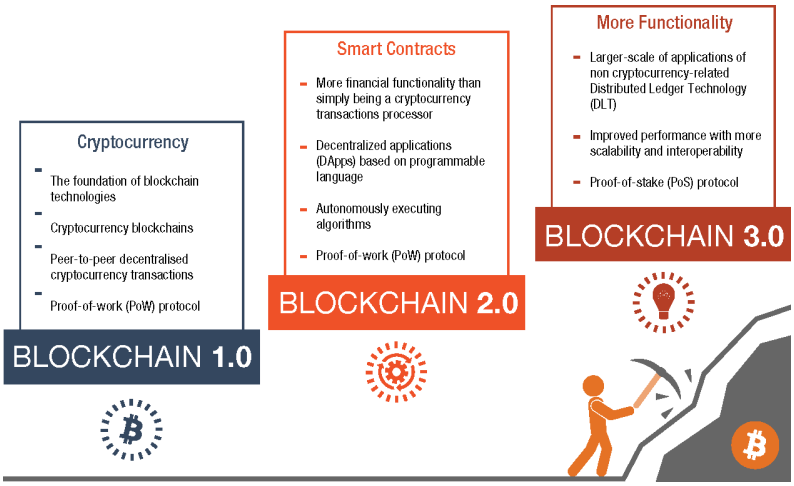


Figure 1.11. Three Different Blockchain Categories

To fully harness the potentials of blockchain in 2nd and 3rd generation applications, three core features must be delivered simultaneously: security, decentralization, and scalability. These attributes lie at the heart of blockchain's success, and compromising even a single benefit among the three could lead to the failure of blockchain deployment in real-world applications.

1. **Blockchain 1.0:** The 1st generation of blockchain emerged with the creation of Bitcoin in 2009. This initial iteration primarily focused on introducing a secure and decentralized peer-to-peer electronic cash system. Security was achieved through the ingenious use of cryptographic techniques, making the Bitcoin network resistant to tampering and fraud. Decentralization, another key attribute, was achieved through a distributed ledger system, where transactions are recorded and verified by a network of nodes, eliminating the need for a central authority. However, scalability was limited in this early stage, leading to slow transaction times and high fees during peak usage [2].

2. **Blockchain 2.0:** In response to the limitations of the 1st generation, the 2nd generation of blockchain sought to enhance scalability while maintaining security and decentralization. Smart contracts were introduced, allowing programmable and self-executing agreements on the blockchain. Ethereum, launched in 2015, became the pioneer of this generation, enabling developers to create decentralized applications (DApps) that revolutionized various industries, including finance, supply chain management, and digital identity. However, as the popularity of DApps grew, so did the issue of scalability. The network faced congestion and high gas fees during busy periods, highlighting the need for further improvement [3].

3. **Blockchain 3.0:** The 3rd generation of blockchain arose as a response to the shortcomings of its predecessors. It aimed to provide a comprehensive solution that seamlessly combined security, decentralization, and scalability. One notable example of a 3rd generation blockchain is Cardano, launched in 2017, which employs a unique consensus mechanism, Ouroboros, that allows for high throughput and scalability while maintaining security and decentralization. Other 3rd generation blockchains, such as Solana and Polkadot, have also emerged with similar goals, pushing the boundaries of what blockchain can achieve [4].

- **Security:** The attribute of security is non-negotiable in any blockchain implementation. It relies on cryptographic algorithms to ensure data integrity and protect against unauthorized access or manipulation.
- **Decentralization:** Decentralization, too, is a fundamental aspect that sets blockchain apart from traditional centralized systems. A decentralized network ensures that no single point of failure exists, enhancing resilience and immutability.
- **Scalability:** Lastly, scalability is vital for blockchain's real-world usability, as it allows for increased transaction throughput and lower fees, making it feasible for large-scale applications. The details of these atrributes is depicted pictorially in Figure 1.12.

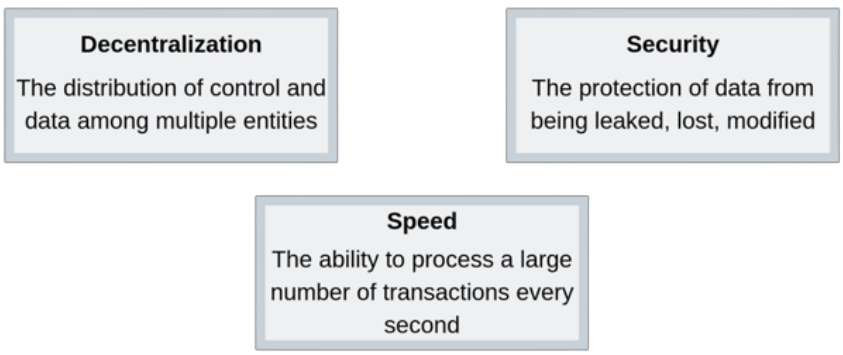


Figure 1.12. Essential and Core Attributes of Blockchain

The attribute of security is non-negotiable in any blockchain implementation. It relies on cryptographic algorithms to ensure data integrity and protect against unauthorized access or manipulation. Decentralization, too, is a fundamental aspect that sets blockchain apart from traditional centralized systems. A decentralized network ensures that no single point of failure exists, enhancing resilience and immutability. Lastly, scalability is vital for blockchain's real-world usability, as it allows for increased transaction throughput and lower fees, making it feasible for large-scale applications.

Achieving a balance between these attributes is the ultimate challenge for blockchain developers and enthusiasts. Blockchain platforms must strive to optimize each attribute without compromising the others, which often involves complex trade-offs and innovative solutions. Projects like sharding, off-chain solutions, and layer-2 protocols have been proposed and implemented to address scalability concerns while maintaining security and decentralization.

1.8 Problem Statement

1.8.1 Blockchain Attributes and Infamous Trilemma Problem

Blockchain technology has garnered widespread attention for its potential to revolutionize industries and reshape our digital landscape. Yet, beneath its promising exterior lies a fundamental challenge known as the infamous "Trilemma Problem." This predicament arises from the fact that, in its current state, blockchain can only achieve or balance two of its three critical elements simultaneously: (i) Security, (ii) Scalability, and (iii) Decentralization. As blockchain projects and platforms continue to evolve, navigating this trilemma becomes a central focus for developers and researchers seeking to optimize the technology's performance and real-world usability. The trilemma problem is graphically represented in Figure 1.13.

1. **Security:** Security, the bedrock of any blockchain system, is essential for maintaining the integrity and trustworthiness of the network. Through cryptographic algorithms and consensus mechanisms, blockchain ensures that data remains immutable and protected from malicious attacks. The immutability and tamper-resistant nature of blockchain have made it a compelling solution for applications requiring high levels of security, such as financial transactions, digital identity management, and supply chain tracking. However, achieving robust security often comes at the expense of scalability and decentralization.

2. **Decentralization:** Decentralization, one of the foundational principles of blockchain, ensures that no single entity or group holds control over the network. A decentralized

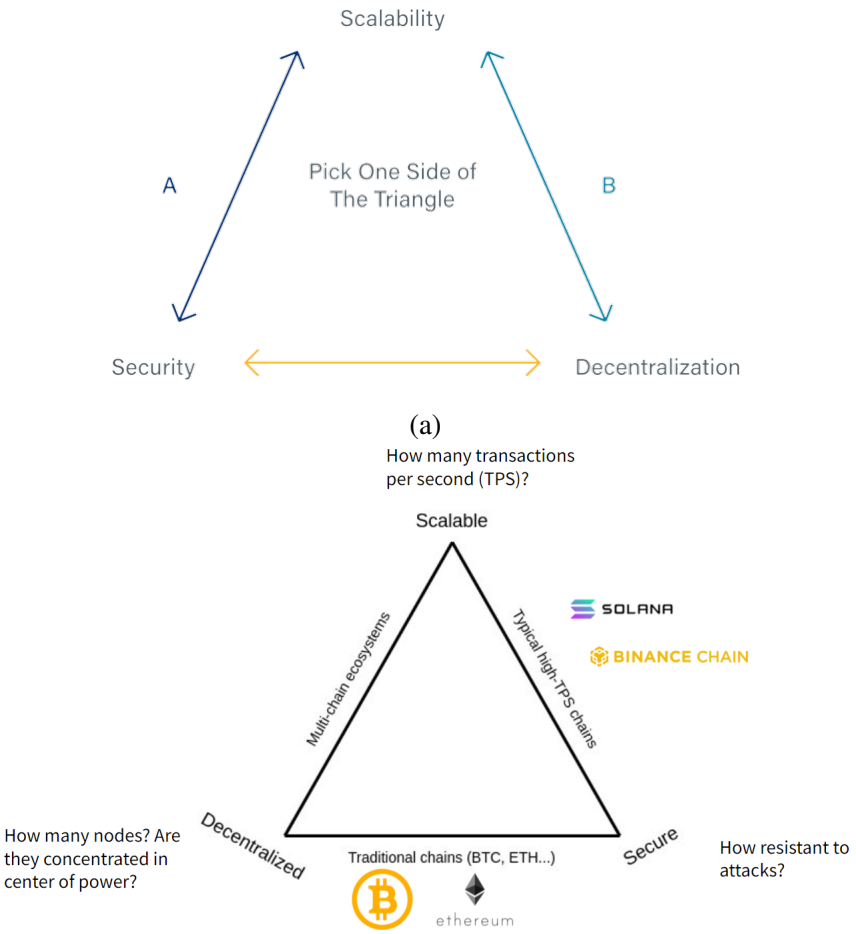


Figure 1.13. Trilemma Issue of Blockchain: At Most 2 Features Can Be Sustained

network distributes data and transaction verification across a vast network of nodes, making it highly resilient to censorship and single points of failure. This trustless nature is vital for applications where transparency, accountability, and independence are paramount. However, achieving a high level of decentralization can impede the network's efficiency and scalability. As the number of nodes increases, the consensus process becomes more complex and resource-intensive, potentially leading to slower transaction times and higher computational costs.

3. **Scalability:** Scalability, the ability to process a large number of transactions efficiently, is crucial for blockchain's mass adoption and applicability in high-throughput use cases. As blockchain networks grow and attract more users and transactions, the challenge of maintaining high transaction throughput becomes evident. Scalability bottlenecks can lead to slow confirmation times and higher transaction fees, hindering blockchain's potential to compete with traditional centralized systems on a global scale. To address scalability concerns, various solutions have been explored, including off-chain protocols, sharding, and layer-2 solutions. However,

these approaches sometimes come at the cost of compromising decentralization or security.

Vitalik Buterin, the co-founder of Ethereum, addressed the challenges mentioned above faced by the current blockchain technologies, defined by the term Trilemma. The concept of Trilemma is identical to the CAP theorem coined by the computer scientist Eric Brewer but is characterized in terms of modern distributed networks, which states that blockchain in its current state can only achieve or balance two of its three critical elements simultaneously: security, scalability and decentralization [5]. For example, public blockchains are highly decentralized and can prevent security threats but can only process a minimal amount of transactions per second. On the other hand, private blockchains carry a high amount of transactional throughput but are centralized and unable to withstand several blockchain-related attacks [6].

Some blockchain platforms prioritize security and decentralization over scalability. For instance, Bitcoin, as a first-generation blockchain, has achieved remarkable security and decentralization but faces challenges with scalability, leading to limited transaction throughput and higher fees during periods of heavy usage.

On the other hand, certain 2nd generation blockchains, like Ethereum, have prioritized decentralization and scalability, enabling the creation of smart contracts and decentralized applications (DApps). However, this has led to concerns over the network's scalability, especially during peak periods when congestion and gas fees increase [7].

Blockchain 3.0 applications require superior scalability but upscaling this feature leads to the deterioration of both security and decentralization. 3rd generation blockchains, represented by projects like Cardano, Polkadot, and Solana, aim to strike a more optimal balance between the trilemma's elements. By implementing innovative consensus mechanisms and layering solutions, they strive to enhance scalability while maintaining security and decentralization. While these efforts show promise, achieving the perfect equilibrium remains an ongoing challenge [8].

Looking ahead, resolving the trilemma problem could potentially unlock the full potential of blockchain technology, allowing it to scale and perform at levels necessary for mainstream adoption. To this end, ongoing research and development efforts focus on exploring novel consensus algorithms, off-chain scaling solutions, and interoperability protocols that might mitigate the trade-offs faced by current blockchain systems.

1.8.2 Throughput & Storage Issue

The scalability of a distributed network like blockchain refers to handling a large number of transactions within a short amount of time. Two metrics are necessary to define this

blockchain property correctly: (i) Throughput, the rate at which the blockchain can confirm or successfully processes the transactions and (ii) Storage Requirement. These two aspects are interrelated and jointly contribute to the scalability of a blockchain network [9]. For example, if we want to increase the throughput of any blockchain-based system, more transactions need to be stored in each block and therefore, the block size needs to be increased [10]. Unfortunately, expanding the block size results in a storage bloating problem where it requires considerable time to broadcast a particular block on the network for verification and storing purposes. Consequently, increasing the block size to store more transactions eventually reduces the processing rate.

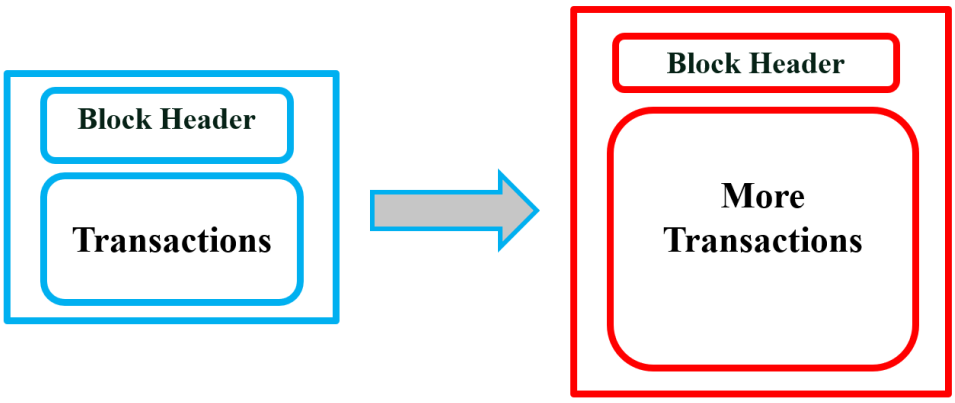


Figure 1.14. Block Size Increases with More Amount of Transactions

Moreover, the total size of the blockchain will increase to the extent that it will become difficult for a newly joined node to replicate the whole ledger into its system. For instance, the current size of the Bitcoin blockchain is approximately 386 GB and any user willing to join this network must download the entire content of the ledger, which requires almost 240 hours on average [11]. Suppose we want to mitigate the adverse effects of incrementing block size on transactions per second (TPS). In that case, the number of participating nodes, along with the miners, needs to be reduced in the network, which eventually intensifies the centralization of the system. In addition, if the number of miners declines, the system's security gets jeopardized as it becomes easier for an adversary to place security threats like 51% attack by taking over most of the network's computational resources [12]. The dilemma of increment in block size due to increased number of transactions is depicted in Figure 1.14

Balancing the trade-offs between throughput and storage is a continuous and evolving challenge for the blockchain community. Researchers and developers must consider various factors, including network decentralization, user experience, security, and storage capacity, when proposing and implementing solutions. Ultimately, achieving a sustainable balance between these elements is crucial for the widespread adoption and success of blockchain-based systems, as it ensures that the technology remains efficient, scalable, and accessible to

all participants. Through ongoing innovation and community collaboration, blockchain can continue to overcome its challenges and unlock its full potential, transforming industries and shaping the future of digital ecosystems.

1.8.3 Security & Decentralization Issue

Improving the security and decentralization by establishing a permissionless blockchain network and deploying a large number of miners and nodes lowers the throughput as it increases the transmission latency of distributing the blocks across the group of miners and system users for (i) validating the transactions and (ii) storing the newly mined block in their ends for the synchronization, respectively [13].

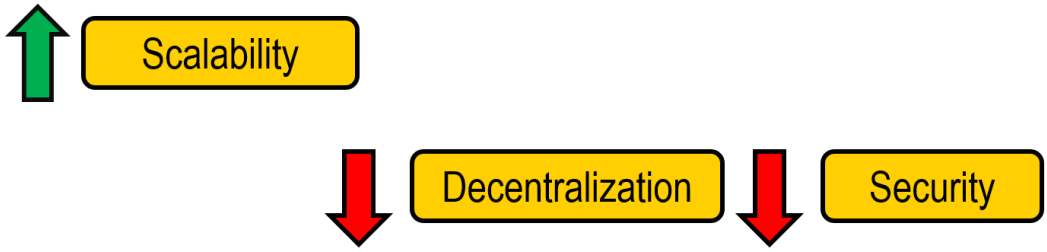


Figure 1.15. Trilemma Issue of Blockchain: Increasing TPS Costs Security and Decentralization

Among the various consensus protocols that power blockchain networks, Proof-of-Work (PoW) has been the most prevalent and well-known. However, PoW's energy-intensive nature and slower transaction throughput have spurred the exploration of alternative consensus protocols to enhance scalability and efficiency. Substituting PoW with faster consensus protocols, such as Proof-of-Stake (PoS) or Delegated Proof-of-Stake (DPoS), indeed results in higher throughput, but it also introduces critical trade-offs that jeopardize both the security and decentralization aspects of the blockchain ecosystem. PoS and DPoS introduce potential centralization risks due to the reliance on stakes or voting power, and the centralization of block production among a limited number of trusted entities raises concerns about collusion and the potential for a single point of failure. Thus, boosting one or two of the three core features of blockchain negatively impacts the rest; therefore, finding a balance among the three is still an open research problem. Degradation of both decentralization and security by improving the scalability is graphically represented in Figure 1.15

Balancing the desire for higher throughput with the critical need for security and decentralization is a complex challenge for blockchain developers and enthusiasts. Achieving a consensus protocol that optimizes all three elements is an ongoing area of research and experimentation. Some projects have opted for hybrid approaches that blend aspects of PoW and PoS to find a middle ground.

1.9 Research Objectives

In this work, we propose an off-chain solution for solving the trilemma issue of public blockchain using the InterPlanetary File System, a distributed and peer-to-peer network-based data sharing and storing service. The key contributions of our research are as follows:

- All the validated transactions are stored in IPFS (off-chain) and a Content Identifier (CID) is generated. As this CID is much smaller in size than the actual transactions, storing CIDs instead of the raw transactions in the ledger results in an enormous amount of transactions per block. Consequently, the rate of transactions per second is increased to a great extent, which is pictorially depicted in Figure 1.16.

Figure 1.16. Amount of Transaction Decreased While Reducing Storage Requirement

- We propose a new double-chain strategy, where two different ledgers, both off-chain and on-chain, are maintained. The actual block, in which all the raw transactions are loaded, is stored off-chain. The hash block, which is 3413 times smaller than the actual block and is stored on-chain, solves the storage bloating problem. A graphical representation of double-chain strategy is illustrated using Figure 1.17.

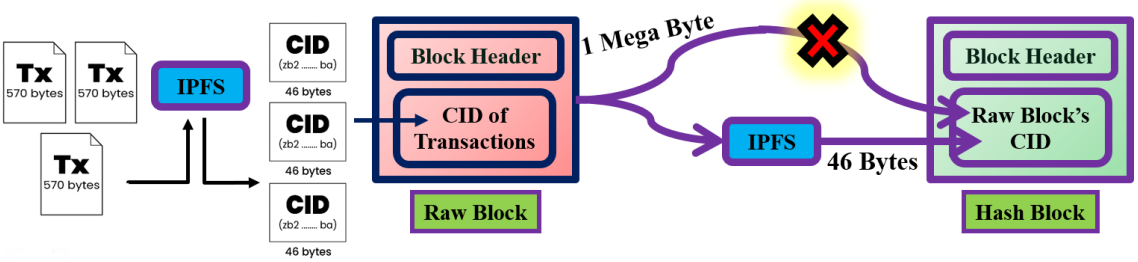


Figure 1.17. Dual-Chain Technique to Resolve High Storage Requirement Issue

- Although our system supports any consensus mechanism, we have adopted a combination of proof-of-work and Nakamoto consensus rules to strengthen the network's security for defending against the security threats towards which private blockchain is vulnerable.

Chapter 2

Literature Review

The researchers have adopted different approaches to alleviate blockchain's scalability, security and decentralization problems, such as sharding, parallel processing, off-chain solutions, hardware-assisted mechanisms, micro-payment channels, new consensus, etc. Blockchain protocols are therefore categorized into the types mentioned above. These protocols try to balance the throughput, storage bloating, security and decentralization, but each technique has its limitations in solving the trilemma problem of blockchain.

2.1 Payment Channel

In the case of Payment Channel Networks (PCN), micropayment channels are used to increase the throughput to a great extent and resolve the drawback of storage at the same time. Without connecting to the ledger of the Bitcoin network, two users of a particular channel can trade as many Bitcoins as they want within the specified time frame [14]. Decker and Wattenhofer introduced duplex micropayment channels in their paper [15], which allow off-chain transactions between two users in both directions, leading to a more scalable payment network on top of Bitcoin.

Due to the utilization of off-chain-based communication, two parties can carry out an enormous amount of transactions between them, which ultimately formulates the technique for executing nearly an infinite amount of transactions inside Lightning Network [16]. Theoretically, Lightning Network offers unlimited throughput. A total block size of only 133 MB is required if each of the seven billion people decides to create two channels annually and be capable of processing unlimited transactions throughout the year. The funds being stuck forever or for the channel's lifetime is considered one of the significant drawbacks of Lightning Network. Also, by performing a Roll-Back attack, malicious parties can settle the payment channel at any previous transactions to keep the balance stale.

To create payment channels between two parties executing off-chain transactions asynchronously, Teechain [17] uses Trusted Execution Environment (TEE) protected treasuries.

Parties of payment channels rely on TEEs rather than the blockchain to identify misbehavior while executing off-chain transactions. Deployment of a 30-machine Teechain carries out a TPS of more than one billion Bitcoin transactions. Like Lightning Network, the funds can get stuck inside a channel for a certain time or a lifetime.

Malavolta and his co-authors first analyzed the security of existing Payment Channel Networks. They reported a new attack named Wormhole Attack [18], which allows a fraudulent user to steal fees from honest ones. To secure PCNs against this attack, they proposed a new cryptographic protocol named Anonymous Multi-Hop Lock (AMHL), which requires an extra round of communication. In this approach, AMHLs are utilized in a scriptless procedure, which reduces the size of the transaction and, eventually, the storage requirement of the blockchain. While a multi-hop Hash Time-lock Contract (HTLC) requires 17 MB of communication load and five seconds to complete a payment, AHML requires less than one megabyte and a few milliseconds.

Raiden [19] offers an off-chain solution for a scalable ERC-20 token transferring scheme on Ethereum. Bidirectional PCN is used for token transfer and rather than connecting each participant directly, tokens are transferred through routes of channels using a channel topology. One of the drawbacks of the Raiden network is that it requires some of the user's tokens to be locked for the duration of the payment channel lifetime. The deposit amount in payment channels is expected to be relatively small, which makes transferring a large number of ERC-20 tokens difficult. To prevent loss of funds, channels must only be settled with the most up-to-date transaction. If not, a roll-back attack could be performed where the channel is settled at a previous transaction, thus keeping the balance unchanged.

2.2 Sharding

In sharding, the consensus is split up into multiple concurrently operating nodes. The load of processing transactions is reduced for each of the validators and the system's overall processing capacity increases proportionally with the number of shards and participants. The cross-shard coordination can arouse significant overhead for the system as well as forfeit decentralization and improve system performance by minimizing security. Also, in sharding-based protocols, all the transactions are stored in every area, or the shards function independently. This results in excessive redundant data or low resource exploitation in the latter case.

In the case of Elastico [20], the network of the miners is partitioned into multiple committees; thus, the transaction rate scales linearly with the computational capacity for mining. Blocks per epoch can be linearly elevated from 1 to 16 by increasing the number of nodes from 100 to 1600, though the epoch time rises from 600 seconds to 711 seconds due to the increment in node amount.

Omniledger [21], which runs on two proof-of-stake-based blockchain solutions named Ouroboros and Algorand, improves the performance of the blockchain by parallelly processing the transactions in an intra-shard approach. To keep the validators bias-resistant and scalable, RandHound [22] is used, which handles the second-key security channel. The checkpoint-based solution is adapted, by which the system removes the requirement of downloading the entire blockchain history for the miners to produce blocks. A throughput of 13000 TPS can be achieved with an adversary rate of 12.5% and 1800 hosts distributed across 25 shards. Unfortunately, the cost of each epoch bootstrap is notable and is in the order of minutes, thus leading to day-long epochs. The system depends on an honest leader to change those who censor the transactions unfairly, which is not enough to guarantee strong security.

Most of the sharding-based blockchains require communication of linear amounts for each transaction and, therefore, cannot exploit the full potential of sharding and experience low scalability and fault tolerance. Rapidchain [23] can tolerate Byzantine faults up to 1/3 of the participants. Block pipelining, a gossiping protocol for large-sized blocks, and intra-committee consensus ensure high throughput. An important feature missing from other sharding-based blockchains is creating a defense against a slowly-adaptive adversary, which Rapidchain achieves through the Cuckoo rule. For a block size of approximately 2MB and 4000 nodes, 7384 TPS can be achieved by pipelining, with 8.7 seconds of confirmation latency. Unfortunately, the bootstrapping overhead is substantial and takes 2.7 and 18.5 hours for 500 and 4000 nodes, respectively. Each participant in these two bootstrapping experiments consumes 29.8 GB and 86.5 GB of bandwidth.

Split-scale [24] attempts to improve the throughput by partitioning the UTXO area and splitting the whole distributed ledger without compromising decentralization. Splitting the ledger into a tree generates multiple sub-chains at every split event, which can operate independently. In each block interval, a block is mined in every sub-chain, exponentially increasing the transactional throughput.

Kan and his co-authors proposed a dynamic multi-chain network for establishing inter-blockchain communication [25]. The throughput of this system depends on the number of chains running in parallel. Sadly, this throughput is affected by the cross-chain transaction ratio, which is the proportion of the number of cross-chain transactions and total transactions. Throughput decreases firmly with the increase in the ratio. Furthermore, no Access Control component and encryption is present and utilized in this protocol.

Wenting, Alessandro, Sergey and Ghassan proposed a permissioned and private blockchain architecture that leverages satellite-chains suitable for industrial organizations, where different consensus mechanisms can run privately in parallel [26]. These satellite chains are interconnected and independent at the same time and maintain their private ledger,

which is inaccessible to other chain members. A regulator supervises the whole network and imposes policies using a smart contract.

Sharding is also utilized in ProductChain [27], where a permissioned blockchain administrated by valid FSC (Food Supply Chain) entities certified by CA (Certification Authority), government and other regulatory bodies to deliver transparency in the production and handling of food items. To improve scalability, multiple blockchains run in parallel instead of a single large blockchain. Each shard, which is denoted as a Local Ledger, maintains its own private and synchronized blockchain. Each transaction takes approximately six milliseconds to be validated. However, a local ledger that can be disconnected for various reasons can fail to generate a complete and accurate final product history due to the missing intermediary transaction.

In [28], the authors enhanced the Byzantine Fault Tolerant Consensus to improve each shard's throughput. To achieve security improvements and high performance for both consensus and shard formation mechanism, the system relies on The Trusted Execution Environment (TEE) delivered by this Intel SGX hardware, which is exploited to eliminate the prevarications of the Byzantine Failure model. The shard size is reduced using the improved fault tolerance of the system's TEE-assisted consensus. In the case of 12.5% adversarial power, 3000 transactions can be processed per second using 36 shards and for 25% adversarial power, a throughput of 954 transactions per second can be achieved.

2.3 Blockchain Delivery Network

BDN is a cloud-distribution network that aims to scale the scalability of blockchain and cryptocurrencies up to thousands of transactions (on-chain) per second. Unlike sharding, centralized and permissioned blockchain systems where the trust is placed in a subset of nodes to improve the system's scalability, BDN boosts scalability by placing trust in the entire network. The behavior of BDN is continuously being examined by the users, thus making it incapable of disfavoring against specific nodes, transactions and blocks. Instead of the raw transactions, BDN uses the cloud to disseminate transactions, index them and use these indexes instead of the original transactions in case of block transmission across the network. The block size is thus reduced by 100 times, considering each raw transaction is 100 Bytes long, whereas an index size is no more than 4 Bytes. A throughput of 1000 transactions per second can be achieved by employing the BDN-based blockchain system [29].

BloXroute [30] is the first practical BDN, a transport layer that runs beneath cryptocurrencies, increasing the throughput by three orders of magnitude. It is the first system that combines a peer-to-peer network with the BDN so that the P2P network can audit its behavior and neutrality. The confirmation time is significantly reduced by order of tens of

milliseconds. If, in any case, the BloXroute behaves maliciously, nodes can replace it by deploying the code of BloXroute at their end; but to incur low costs, it is necessary to limit the rate of test-blocks, which consequently makes the system less secure.

Chameleon [31] is a BDN-based permissioned blockchain protocol in which the non-forking principle is adopted. Unlike sharding, each area can process its transactions independently and cooperate with other areas, enhancing resource utilization and throughput. The consensus nodes are required to store only a single epoch of block and the previous one gets stored in the cloud, substantially reducing the storage requirement. An upgraded Byzantine Agreement Protocol is introduced for selecting the leaders in an efficient, non-deterministic and unpredictable method, named Random Leader Selection Based on Credit Value (RLSCV). Six hundred transactions per second on average can be achieved via Chameleon. Although it has many advantages compared to other blockchain protocols, it is yet a proof-of-concept. When an area shares transactions with another already overloaded, the solution to this overload situation is not discussed and is considered an open research problem.

2.4 Hardware-Assisted Approaches

Trusted Execution Environment (TEE) is a particular area of the main processor that restricts unauthorized access and tempering of stored data and applications using a certified hardware-based crypto engine. It also enables multiple parties to share their data in a secure collaborating environment and interactively process them using an easy-to-use and convenient cryptographic API. Blockchain-based protocols like FastBFT and Teechain [17] utilize TEE to strengthen their security.

Sharding can also make use of TEE and the authors of [28] applied TEE in their proposed sharding-based permissioned blockchain to enhance the performance of current Byzantine Fault Tolerant protocols, to generate random values which are unbiased and to reduce the shard size, which has been already discussed in the Shadring subsection.

Fast Byzantine Fault Tolerant (FastBFT) [32] also improves over the traditional BFT, which has poor efficiency and scalability and is considered infeasible for any popular enterprise-based system like Amazon and Google. Along with the optimizations like Failure Detection, Optimistic Execution and Tree Topology, the combined approach ensures high transaction throughput and low latency. Public key-based operations like Multi-signatures are not required in FastBFT's Message Aggregation technique, substantially reducing the computational load. The communication overhead is balanced by arranging nodes in a tree's topology and ensuring that message aggregation and inter-server communication take place along the edges. For efficient non-primary fault tolerance, FastBFT relies on timeouts for crash failure detection and parent nodes for detecting child

node failures. FastBFT can achieve up to 100000 TPS for a block size of one megabyte. FastBFT's performance is hardly affected by the number of replicas, which is denoted by n. For instance, for n=199, FastBFT can process 370 TPS, whereas CheapBFT, Zyzzyva and XPaxos achieve throughput of 38, 56 and 42 TPS, respectively. When n=103, only 34 faulty replicas can be tolerated by Zyzzyva. On the other hand, FastBFT can tolerate 51 faulty replicas.

2.5 Parallel Processing

In parallel processing, the consensus, mining process, or smart contract is run in parallel to improve the system's scalability and transaction processing speed significantly.

Hazari and Mahmoud, in their paper [33], proposed a method to accelerate the proof-of-work consensus via parallel mining instead of the solo approach. Miners work in parallel to calculate the nonce for the same block, but they are not allowed to find the nonce beyond the allocated range. Consequently, it requires less processing power and is less time-consuming than conventional proof-of-work. According to the preliminary results, the proposed system improves the scalability of the traditional proof-of-work by 34%. The suggested parallel proof-of-work restricts receiving mining rewards for the miners, ensuring more decentralization than the traditional PoW consensus. However, if the block manager cannot respond or gets disconnected for a certain period, a single point of failure occurs. As two miners do not get the same nonce range from the manager, what happens to that particular nonce range when its miner becomes unavailable is not discussed.

Gao and his co-authors proposed an enhanced smart contract execution strategy where multiple contract run in parallel to escalate the throughput [34]. The scheme is based on two strategies: (i) Contract Partition Algorithm, which exploits linear programming to partition the contracts into multiple subsets, (ii) Random Assignment Protocol which randomly assigns the subsets of smart contracts to the subgroups of nodes without sacrificing the strength of proof-of-work consensus. A noteworthy limitation of this system is it relies significantly on the Integer Linear Programming (ILP) solver's efficiency. Using the current solvers like GUROBI and SCIP takes considerable time for the partitioning task if given more than 1000 smart contracts.

2.6 New Consensus

Proof-of-property [35] improves scalability by explicitly storing the system's state in the latest block and relevant part of the state inside the new transaction. In this way, transactions can be validated without downloading the complete ledger. Therefore, 90% of all the preceding blocks can be removed and storage requirement and verification complexity are reduced to almost 90%. Finding a path for validation requires an intensive

amount of nodes. Forks might arise as some clients' states can be different. If a transaction being processed is going to be stored in a block that will be a part of the dismissed fork, it will be considered invalid. How to defend the security threats due to the forking of the blockchain is not considered and discussed. As the body of the previous blocks is not required to validate new transactions, the older blocks can be deleted. This leads to the failure to synchronize the validation path with the latest block's root hash. This situation can be avoided, but users must always remain online to do so.

Applying delegated proof-of-stake (DPoS) to IoT-based applications is challenging due to the deployment of IoT devices on a large scale and a large amount of data. Roll-DPoS [36] has all the advantages offered by the original DPoS but further improves the original's extensibility of dealing with large and complex blockchain-based architectures. Unlike traditional PoS, the whole network does not get involved in producing blocks. Instead, a pool of candidates is initiated and through community voting and smart contract, nodes receiving the maximum number of votes become the block producer. In contrast to the classical DPoS, it allows a comparatively larger number of blockchain nodes to produce blocks and earn rewards.

Wang and his co-authors introduced a hierarchical blockchain-based scalable architecture named SMChain [37], by which the metering in large industrial plants can be secured. It adopts a two-layer blockchain where the individual plants have independent local ledgers and a state blockchain resides in the cloud. To enhance the scalability of the local ledgers, an improved BFT consensus is proposed, which embraces the (k, n)-Threshold Signature approach to handle Byzantine faults. The threshold signatures are used to avoid the broadcasting message, which eventually reduces the plant communication complexity. However, SMChain is still a theoretical framework and its performance needs to be practically evaluated using a real-world application.

Colosseum [38] is a protocol based on a knockout tournament over a ring network for reaching consensus on the next pool of block producers. In each tournament, knockout-based two-player rounds are held where the winner receives a proof-of-win certificate as the winning evidence and advances to the next round to earn eligibility for producing blocks. The winner is selected randomly and, therefore, prevents defrauding and fairness is ensured. In each tournament, multiple blocks are proposed and as a consequence, conflicts may arise. To date, no blockchain is known to handle multiple blocks simultaneously, which is a prerequisite to fully exploiting Colosseum's potentiality. Also, to adopt Colosseum in private blockchain systems, the security analysis of this consensus against malicious practices is required.

Min, Li, Liu and Cui introduced a new consensus for permissioned blockchain systems named "Permissioned Trusted Trading Network [39]." The network is partitioned into

multiple sub-committees. Each sub-committee runs the Peer Inner Consensus consensus to process an independent set of transactions and blocks separately. A random partitioning algorithm is used to partition the network, which can limit the malicious activities of dishonest nodes. The proposed Permissioned Trusted Trading network could achieve up to million transactions per millisecond. One thousand micro blocks per second can be produced, whereas Bitcoin requires approximately 10 minutes to create a block. Also, a key block needs only one minute to be generated.

2.7 Decoupling Transaction Verification from Mining

Protocols like Bitcoin-NG, ByzCoin and TrueBit separate the verification of transactions from mining blocks, thus achieving large transaction throughput.

Bitcoin-NG [40] decouples Bitcoin's overall operation into two phases: (i) Serialization of Transactions and (ii) Electing Leaders for Each Epoch. Bitcoin-NG utilizes two types of blocks: (i) Key Blocks for the election of the epoch leaders and (ii) ledger entries are stored in Microblocks. Bitcoin-NG offers better latency and bandwidth as its latency and bandwidth depend only on the network's propagation delay and computational capacity of the nodes, respectively. Simultaneous mining of Keyblocks can give rise to temporary forks and therefore, the system can fall under an inconsistent state for approximately 10 minutes or more. Moreover, a race condition occurs for the miners as two different types of blocks reside in the same blockchain.

ByzCoin [41] applies communication trees for optimizing the commitment and verification of transactions. A collective signing mechanism is used to reduce the cost of transaction commitment verification from O(n) to O(1) and the cost of Practical Byzantine Fault Tolerance (PBFT) rounds to O(logn). Unlike Bitcoin-NG, where a corrupted epoch leader performs a double-spending attack by rewriting history, ByzCoin prevents this by ensuring the irreversibility of Microblock commitment. Also, there are two parallel blockchains in ByzCoin, one to store the Microblocks and another to store the Key Blocks to prevent race conditions for miners. Currently, Byzcoin can achieve throughput higher than PayPal and for block size of one megabyte and 144 miners, less than 20 seconds is required to reach the consensus. Being vulnerable to DoS attacks from Byzantine nodes and slowdown, Byzcoin does not perform better than proof-of-work in terms of security. Moreover, malicious leaders can deprive the victims from getting rewards and can make attempts to censor transactions. They can also eliminate nodes during the consensus process and trigger a double-spending attack by splitting the authentic nodes into two disjoint groups. Byzcoin only guarantees security when the attackers control less than 50% of the consensus group. Where Bitcoin is considered safe even at 48%, Byzcoin's security breaks at only 30%.

In the case of TrueBit [42], the decentralized group of miners and immediate applications

are operated by a smart contract. The computational overload of Ethereum is mitigated substantially as only a few entities are assigned to perform the computations rather than requiring every miner to replicate each smart contract action entirely. A resolution layer is utilized as the Verification Game to verify whether a computational task is performed appropriately or not. Unlike cloud computing, where the trust is placed on the cloud for ensuring correctness, TrueBit offers financial incentives instead of the requirement of placing trust on certain dedicated nodes. Unlike sharding and serializing-based blockchain protocols, two of the miner's tasks: (i) selecting particular transactions to include in the blockchain and (ii) validating the transactions; are decoupled to make the mining task simple and scalable. TrueBit mitigates Ethereum's storage-bloating issue by accessing the massive amount of data from Swarm, which is a P2P storage system. In the case of Big Data Applications, TrueBit's Verification Game performs poorly as its performance degrades while dealing with big data computations. Due to its One-Size-Fits-All nature, TrueBit cannot rely on its resolution layer for handling big data applications.

2.8 IPFS-based Solutions to Improve Blockchain Performance

In [43], the authors proposed a scalable blockchain storage model based on Distributed Hash Tables (DHT) and InterPlanetary File System (IPFS) named DIBS (DHT-IPFS Blockchain Storage). Their aim was to address the issue of storage scalability in blockchain technology. The model focused on reducing the storage burden on individual nodes, improving storage efficiency, and ensuring that frequently accessed ("hot") data is readily available. The authors employed a Block Least Frequently Used (BLFU) replacement algorithm to optimize storage usage, distinguishing between "hot" and "cold" data based on access frequency. They also leveraged the Chord protocol, a DHT protocol, to distribute the storage pressure among nodes. The authors conducted their experiments on a server with specific configurations, setting up five blockchain nodes to simulate and test the DIBS model. The authors found that local storage consumption in the DIBS model increases linearly when the number of blocks is small, but stops increasing beyond a certain number of blocks (greater than 500). They attributed this to the Chord protocol and the BLFU block replacement algorithm. In comparison to other models, the DIBS model occupied smaller node storage space for the same number of blocks. The DIBS model also showed higher query efficiency as it doesn't split block data, preserving the atomicity of blocks in the blockchain. The authors' model was tested only on a server with a specific CPU and memory configuration. The results might vary with different hardware configurations. The model heavily depends on the effectiveness of the Block Least Frequently Used (BLFU) replacement algorithm, and the performance of this algorithm could impact the overall performance of the model. The authors tested their model with only five blockchain nodes; the model's performance might vary with a larger number of nodes or with different node configurations.

In another paper [44], Zheng and his co-authors proposed an innovative storage model for blockchain data based on the InterPlanetary File System (IPFS). They aimed to address the issue of continuous growth of blockchain data volume, which impedes the participation of many nodes in the network and serves as a bottleneck for the development of blockchain technology. In the proposed model, miners deposit transaction data into the IPFS network and pack the returned IPFS hash of the transaction into the block. The authors applied their model to the Bitcoin blockchain for testing. he authors found that their model could achieve a compression ratio of 0.0817, indicating a significant reduction in the size of blockchain data. The model also enhanced the security of block data, as any attempt to change transaction data in a block would result in a change to the IPFS hash, causing a significant difference in the Merkle root and block header hash. (iv) Limitations: The limitations of this model were not explicitly outlined in the document, but the authors do discuss the limitations and challenges associated with some existing approaches to blockchain data storage, which provide some context. The document does not explicitly address transaction per second (TPS) rate, and its impact on transaction speed is not directly addressed, which could be an area for further investigation or future work. The document does not provide specific measurements or metrics related to security performance.

2.9 Various Approaches to Solve Trilemma

The authors of [45] present a new model called Trifecta, aimed at addressing the blockchain trilemma, which is the difficulty of concurrently ensuring decentralization, security, and scalability in a blockchain system. Trifecta, available in both Proof-of-Work and Proof-of-Stake formats, purports to provide full permissionlessness, resilience against adversaries controlling 50% of the resources, and vertical (throughput and confirmation latency) and horizontal (number of nodes) scalability. Trifecta leverages a disjoint-assemble model to tackle the trilemma and employs sharding, a method where multiple blockchains run simultaneously, each managing a separate set of accounts. Nodes self-allocate to specific shards to maintain decentralization. For security, Trifecta separates validation from consensus. While nodes agree on transaction sequences per shard, they don't all have to be valid. After consensus, any shard node can download the shard ledger, process transactions, and get the final state. Built on Prism, a blockchain protocol, Trifecta claims a throughput of 250,000 transactions per second and a confirmation latency of 20 to 30 seconds on a network of 100 EC2 nodes. However, there are potential practical issues. Cross-shard transactions could be complex and introduce latency or other issues. Security at the individual shard level is a concern, as malicious actors controlling a shard could disrupt its transactions. Real-world communication delays could be challenging with a large volume of transactions and nodes. Uneven node distribution among shards could affect load balance and performance. Lastly, Trifecta needs at least one honest node per shard; otherwise, the system's liveness could be compromised.

Monte and his co-authors in their paper [46] suggested a unique blockchain architecture that could scale to any workload, provided there is a corresponding increase in nodes. Their strategy ensures security and decentralization without compromise, thereby addressing the scalability trilemma. The proposed system includes several components such as committees for validating transactions, blocks broadcasting constant size data, storage nodes holding only parts of the state and corresponding parts of a Merkle tree, a pipelining process to spread computational load, and a truncated block history. In this architecture, nodes generate candidate transactions, which are sent to Confirmation Committees (CC) for validation. Transactions approved by these committees are processed further by Root-hash Pipeline Committees (RPC) to compute the state root-hash, which is included in the new block. By distributing computational load through pipelining and committees and removing the need for every node to store the full blockchain state, the system can scale effectively. However, there are potential implementation challenges. These include the substantial communication overhead due to the large amount of inter-node and inter-committee communication, the complexity of managing the pipelining process, scalability issues of storage nodes as the blockchain grows, and latency problems caused by network delays. The model also presumes the honesty of nodes and the accuracy of their proofs, which could be problematic in real-world applications without appropriate checks and balances. Additionally, the practical feasibility of development, deployment, and adoption could pose significant challenges.

Another protocol named Algorand aims to address the blockchain trilemma by employing cryptographic sortition to randomly select a set of voters for achieving consensus on each block, ensuring high blockchain linearity and a fast block rate. In the research paper [47], the authors provided a thorough security analysis of Algorand and presents a potential attack scenario. The focus is on exploiting a security flaw in the message validation process of the Byzantine Agreement (BA) to disrupt the protocol or isolate a small network partition. This vulnerability poses a significant security threat to the Algorand protocol, despite its promising features for addressing the blockchain trilemma. The study also identifies several other limitations of the Algorand protocol. Firstly, it relies on the Honest Majority of Money (HMM) assumption, assuming that the majority of network wealth is held by honest users. If a significant portion of wealth falls into dishonest hands, the protocol's security could be compromised. Additionally, Algorand assumes partial synchronization of honest nodes' clocks with a bounded tolerance, which could pose a vulnerability. Furthermore, the protocol's current specifications do not explicitly address whether a node must choose peers based on stake, making it susceptible to Sybil attacks.

Chapter 3

Overview of the Proposed System

3.1 Incrementing Throughput without Increasing Block Size

The user first executes a transaction and uploads this transaction to the IPFS. IPFS returns a CID for the uploaded transaction, which is about 46 Bytes in size. The user then sends this CID to a nearby miner and upon receiving the CID, this miner fetches the actual transaction from the IPFS, as demonstrated in Figure 3.1. After successfully verifying the transaction, the miner adds this CID to his mempool and sends this CID to other miners. If the transaction turns out to be invalid, then it gets discarded. After validating some transactions, a block is generated using the CIDs of all the validated transactions, which we refer to as *"raw block"*. The size of this raw block can be varied according to the requirement of the system, but we wish to keep it as same as the block size of Bitcoin, which is approximately 1.3 MB. As CIDs will be included in the raw block instead of the actual transactions, a tremendous amount of transactions can be appended. As a result, the number of transactions is increased significantly per block and eventually, the throughput is increased without expanding the block size. The diagram in Figure 3.2 accumulates the overall procedure of the proposed system.

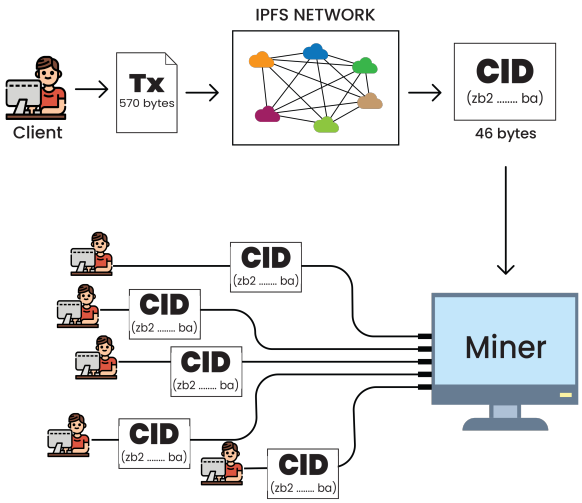


Figure 3.1. Receiving Transactions at the Miner's End from Clients in CID Form Through IPFS

3.2 Double-Chain Technique

In the case of Bitcoin, the 1.3 MB-sized blocks are distributed among all the network nodes. The total size of this ledger has exceeded 360 GB, which keeps increasing as around 144 blocks are generated daily. Subsequently, the storage bloating problem is becoming a matter of concern as the new users must replicate this 360 GB data in their system. However, in our proposed system, the users do not need to store the raw blocks on their end, which are 1.3 MB long. Instead, the miner uploads this raw block to IPFS and a corresponding CID of this block is generated. Using PoW consensus, a new block is created using only the CID of the raw block, which we refer to as the *"hash block"*. As only a single CID is inserted in this hash block, the size of this type of block is remarkably lower compared to the raw block. The distribution of this hash block becomes faster among the miners as it requires low bandwidth due to the reduced size. The term "double-chain" derives from the concept of two existing ledgers in our proposed system: one for raw blocks and the other for hash blocks. Miners can access the ledger of raw blocks through the hash blockchain and IPFS for validating the transactions. In this approach, high throughput is achieved without increasing the block size.

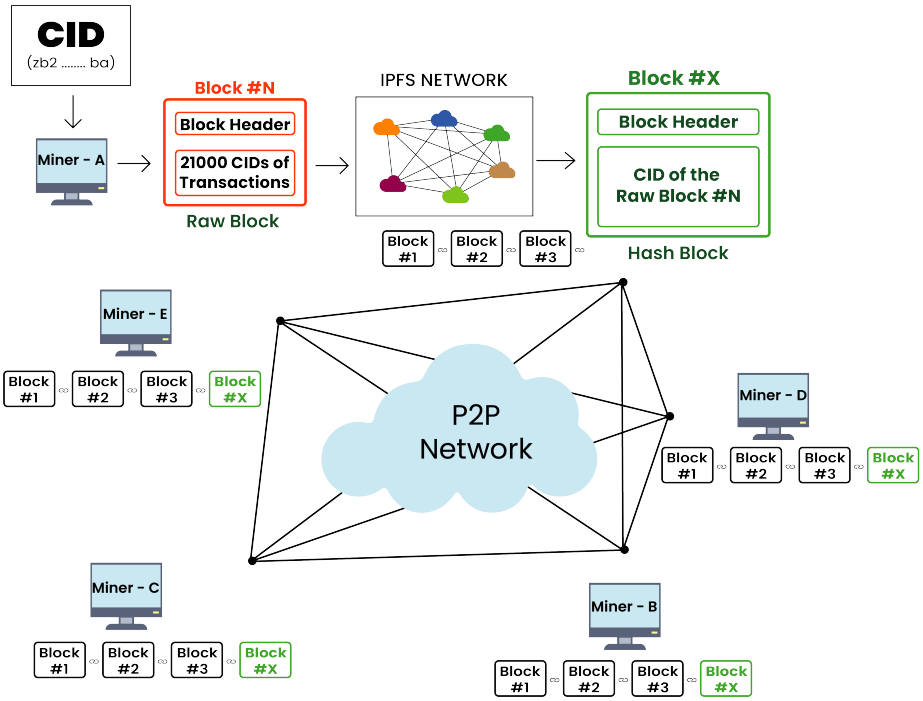


Figure 3.2. Overview of the Proposed System's Workflow

Chapter 4

Implementation Details

The following sections describe the step-by-step chronological procedures being followed by our proposed system, which is summarized and illustrated in Figure 4.1. Among these procedures, generation of key-pairs, addresses and transaction are executed at the user end, while the rest of the steps are carried out at the miner's end.

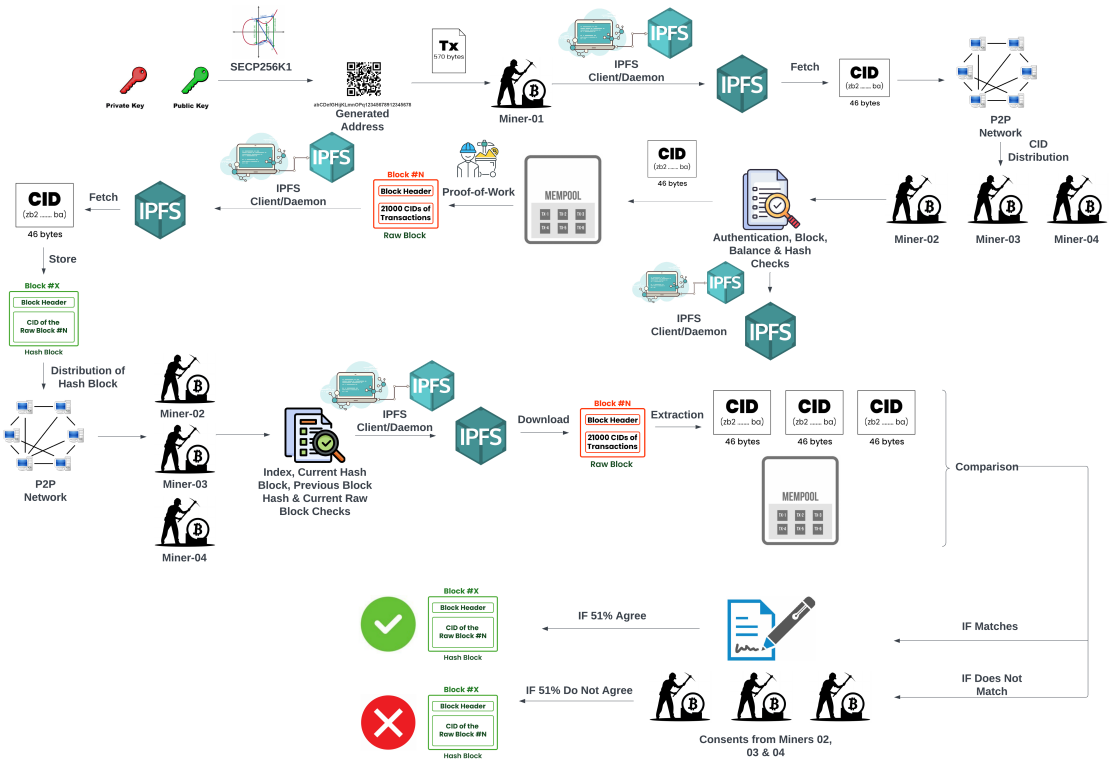


Figure 4.1. Detailed Working Procedures of the Proposed Methodology

4.1 Generating Key Pairs

Key pair is required to conserve the ownership and authenticity of any transaction. A key pair comprises two elements: (i) Public Key and (ii) Private Key, which are linked cryptographically. Two classes of key pair-generating algorithms exist: Symmetric-Key algorithms and Asymmetric-Key algorithms. Data Encryption Standard (DES) and Advanced Encryption Standard (AES) fall under the category of Symmetric-Key algorithms.

In contrast, Asymmetric techniques include Rivest–Shamir–Adleman (RSA), Digital Signature Algorithm (DSA), Elliptic Curve Digital Signature Algorithm (ECDSA), Diffie-Hellman Key Agreement Protocol, etc. Among the two types of key generation techniques, Symmetric-Key algorithms are faster than Asymmetric ones. Despite being fast, our proposed technique utilizes the Asymmetric-Key generation technique for security purposes [48].

Symmetric-Key generation algorithms are less secure as they use only a single key to encrypt and decrypt information. Whereas, Asymmetric techniques employ a public key to encrypt data and a private key to decrypt them [48]. Among the aforementioned Asymmetric-Key algorithms, ECDSA is preferred over RSA and DSA for our proposed blockchain, as the latter ones are required to generate key pairs of 3072-bits each for ensuring 128-bit security. On the other hand, similar security can be achieved in the case of ECDSA by generating key pairs with only 256-bits in size [49].

In addition, the dependency of ECDSA on randomness is of the same level as DSA. Among different ECDSA algorithms, our system employs SECP384R1. This cryptographic algorithm provides a higher level of security compared to SECP256K1, which is essential for the sensitive nature of blockchain transactions. While it may require slightly more computational resources, the increased security benefits offered by SECP384R1 make it a worthwhile trade-off. Just like other ECDSA algorithms, SECP384R1 is built in a non-random procedure, which allows highly efficient processing when optimized appropriately [50]. Furthermore, to defend and minimize the possibility of any backdoor attack, the constants of ECDSA are selected predictably [51]. It's worth noting that while Bitcoin and Ethereum use SECP256K1 for generating key pairs [52], our system opts for the higher security of SECP384R1.

4.2 Generating Addresses

To execute a transaction, the addresses of both the sender and the receiver are required. In blockchain-based systems, a user can use the same address every time for sending and receiving addresses, but in the case of our system, the same address can not be used more than once for receiving assets and a new unique receiving address must be generated each time a user receives assets from a particular node. At the time of transferring assets to another user, this receiving address will be used as the sending address, but after using this address again as the sending one, this address can not be further utilized and the user must generate a new unique address for receiving assets. If any node wants to transfer a portion of the asset, which belongs to the address that has already been used as the receiving one, the user must generate a new address to which he will transfer that specific amount of asset. This necessity of generating a new address for each transaction is a key feature in our system for two reasons:

Firstly, it greatly facilitates and speeds up the verification process of transactions. In our proposed system, when validating transactions, we only need to traverse a specific block or a few blocks to certify the transaction as valid. If we allowed the reuse of old addresses for new transactions, our system would have to traverse multiple blocks, maybe even all the blocks from IPFS to validate those addresses. This process is highly time-consuming and could significantly hamper the system's performance.

Secondly, generating a new address for each transaction significantly enhances the security of the system. By ensuring that each address is used only once, we reduce the potential for address-based attacks and increase the difficulty for attackers to trace transaction histories. The transient nature of the addresses makes it challenging to associate a transaction with a particular user, thus providing a layer of privacy and security.

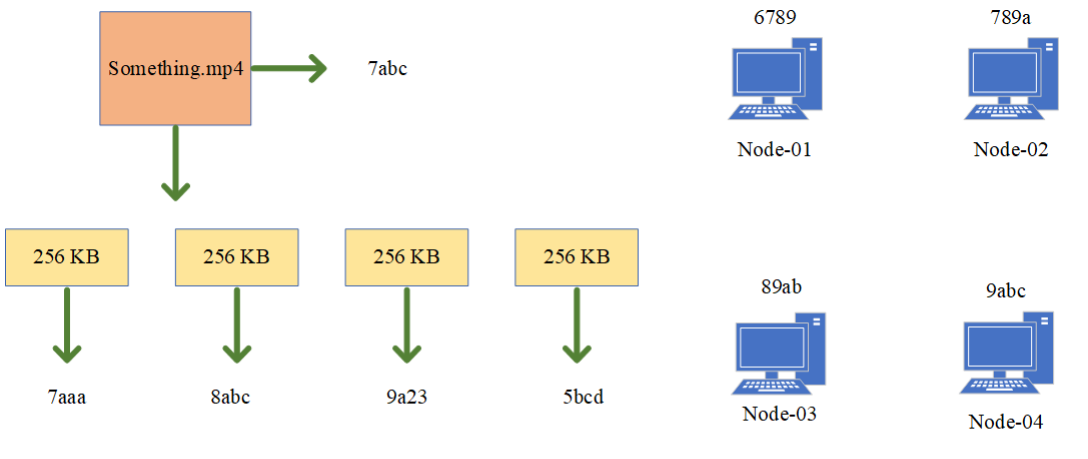
4.3 Generating Transactions

Like most blockchain-based systems, all the transactions in our system consist of two components: (i) Message and (ii) Digital Signature. The message comprises the sender's and receiver's information along with the receiver's public key. The sender's information includes the sender's address and block number, while the receiver's information contains the amount of the asset he is entitled to receive and his address. Block number refers to that particular block that serves as the evidence of the sender's ownership of that particular asset he wants to send to the receiving address. The block number is provided to avoid traversing the entire blockchain, implicitly contributing to the improvement of the system's throughput. This whole message is encrypted using the sender's private key and this encrypted data is referred to as the digital signature of the transaction. Verification of a transaction's authenticity is carried out using this digital signature.

4.4 Uploading Transactions to IPFS

After the generation of a transaction, it gets uploaded by the user to IPFS. Using the Merkle Directed Acyclic Graph (DAG), the transaction is split into multiple chunks, each of which is 256 KB in size. This chunking is performed in a deterministic manner, meaning that the same transaction will always be divided into the same chunks. The order of chunks is determined by their position in the original transaction data, starting from the first byte and proceeding sequentially. Thus, the integrity of the original data ordering is preserved. IPFS then distributes these chunks to several users of the network. Data-retrieval information like the location of the peer where a particular chunk is stored and the routes to reach each of the peers are stored in a Distributed Hash Table (DHT). Every chunk is assigned a unique identifier, an SHA-256 hash of the contents of that specific chunk.

Once all the chunks of a particular transaction have been hashed and their unique identifiers have been assigned, these identifiers are assembled in their original order and fed into the



Distributed Hash Table

Chunk	Peer ID
7aaa	789a (Node-02)
8abc	789a (Node-02)
9a23	9abc (Node-04)
5bcd	6789 (Node-01)
7abc	On My Node

Figure 4.2. Working Procedure of IPFS

SHA-256 hashing algorithm to generate a Content Identifier (CID). This CID is a unique identifier for the entire transaction as it exists in the network. Using this CID, anyone can retrieve the transactions from the IPFS. As all the chunks of a transaction get verified by the checksum, any occurrence of tampering a transaction can easily be traced. Additionally, storing duplicate copies of transactions is also prevented and discarded by IPFS. Upon receiving the CID of the transaction, the user then sends this identifier to the miner for verification purposes. After successful verification, this transaction is stored in the ledger. This IPFS working procedure is illustrated in Figure 4.2.

4.5 Transferring Transactions to Nearby Miners using Peer-to-Peer Network

A peer-to-peer network is essential for any blockchain-based system as no centralized server exists and every node of the system serves both as client and server. In a P2P network, there is no single point of failure and it outperforms any centralized network in terms of performance, file-sharing speed, cost-effectiveness and adaptability, as system resources are distributed among the peers of this decentralized network. The CIDs received from the IPFS are required to send to the nearby miners for verification of the transactions. The way these CIDs are transferred is a P2P network for ensuring the facilities mentioned above.

4.6 Fetching Transaction from IPFS

After receiving the CIDs from the users via the P2P network, miners fetch the actual transactions from the IPFS. For an individual CID, the IPFS node at the miner's end utilizes the Merkle DAG to discover all the 256 KB-sized chunks that make up the desired transaction to be fetched.

In the event that some IPFS nodes are offline during the process of fetching transactions, our system leverages the inherent redundancy of IPFS to ensure data availability. As transactions are chunked and distributed across the network, multiple copies of each chunk are stored on different nodes. This redundancy allows for the retrieval of the desired transaction even if some nodes are offline. Continuing with the process, the DHT is examined by the miner not only to find the identities of the peer nodes which store those particular chunks but also to identify alternative sources in case some nodes are unreachable. Thus, if a node that stores a particular chunk is offline, the miner retrieves that chunk from another node that also stores it. This feature ensures high data availability and resilience to node failures, enhancing the reliability of the system.

After determining the nodes containing the necessary chunks, the miner retrieves the routing information to reach out to those IPFS nodes that keep the desired chunks at their end. IPFS finally assembles all the chunks to restore the original transaction, which the miners utilize for validation. As tampering with the transactions in IPFS completely alters the corresponding CID, it is apparent that tampering with the transactional data is impossible.

4.7 Verification of Transactions

Four different checks are required to verify each of the transactions. A transaction is considered invalid if it fails to pass any of the four verification checks. The followings are the essential checks needed to ensure the credibility of the transactions in our system:

- **Authentication Check:** The sender's authenticity is validated using the Authentication Check. All the transactions are encrypted using the private key and therefore, a digital signature is created. The authenticity of the sender's identity is certified using the sender's public key, transactional data, and digital signature.
- **Block Check:** If a user wants to send assets to any address while executing a transaction, the sender must provide a block number. This block includes the transaction, which serves as the sender's receipt of the assets he wants to transfer. The sender served as the receiver in that particular block's transaction while claiming the assets. To qualify for the test, the receiving address of the formerly verified transaction must be the same as the sender's address of the transaction currently

being verified. If the addresses match, then the transaction is forwarded to the next verification check.

- **Balance Check:** This check's purpose is to ensure that the sender does not spend more assets than he owns. The number of the block, which is the evidence of the sender's ownership of the assets, provided in the Authentication Check is again used by the miner to calculate whether the quantity of the assets in that particular block is less than, equals to, or greater than the amount he wants to transfer. A transaction passes this test if the sender owns an adequate amount of assets before sending them to a specific address.

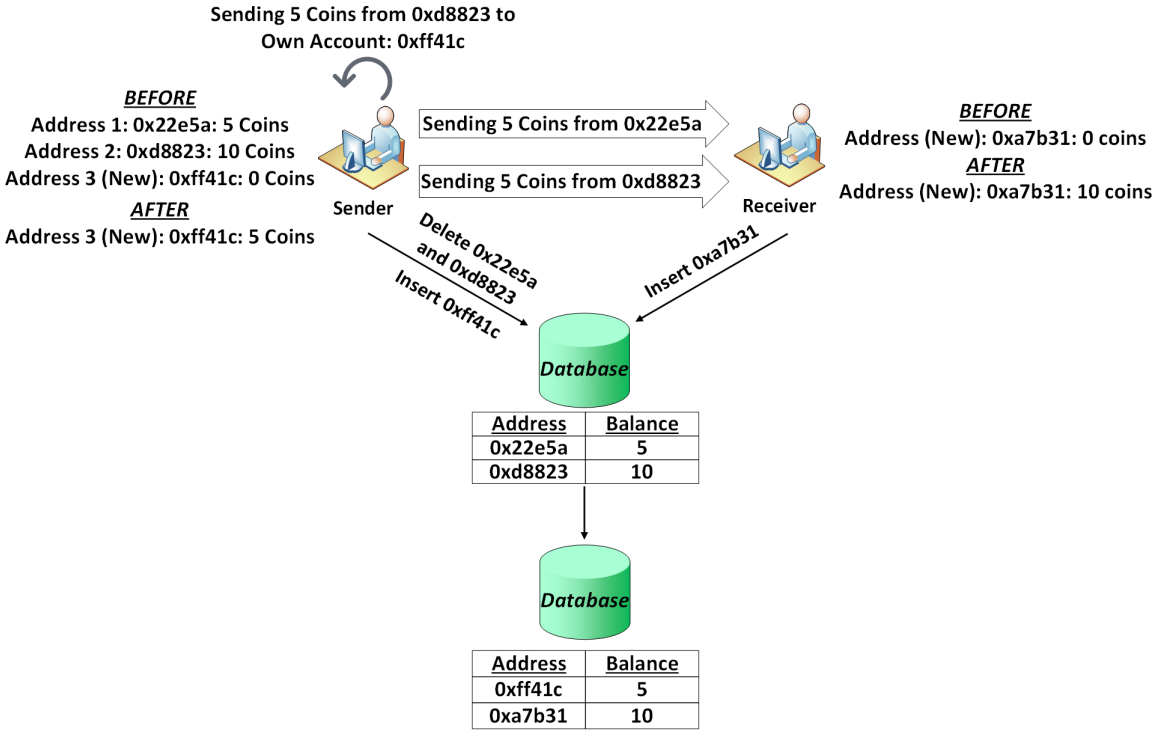


Figure 4.3. Utilization of Database to Prevent Double Spending Attack

- **Double Spending Check:** This check is required to ensure that any user does not use a particular asset more than once. Our system maintains a database that only stores those addresses with a certain amount of assets. Before transferring assets to another user, the sender's address must be present in the database as the receiving one, as the assets have been owned as the receiver in a previously validated transaction. Suppose the sender's address is found in the database, along with passing all the aforementioned verification tests. In that case, the transaction is considered valid and the sender's address is then removed from the database. Meanwhile, the receiver address in the newly verified transaction is added to the database. The database is updated every time a transaction gets confirmed. It is to be noted that a sender can use multiple receiving addresses he owns for transferring assets in a single transaction, as a particular receiving address may not contain an adequate amount of

Algorithm 1 Authentication + Block + Amount Checks

Input: Original Transaction from IPFS, $ipfsRawTxn$
Output: Partially Verified Transaction, $ipfsRawTxn$

$count \leftarrow 0$
procedure AUTHBLOCKAMOUNTCHECKS($ipfsRawTxn$)
    $msgHash \leftarrow$ sha3_256Hash($ipfsRawTxnMsg$)
    $valid \leftarrow$ verify($ipfsRawTxnPubKey, msgHash,$
$ipfsRawTxnSign$, SigningKey.generate(curve=NIST384p))
    if $(valid = true)$ **then**
        $count \leftarrow count + 1$
    end if
    $allSndrAddrAmt \leftarrow 0$
    for each sender address i in $ipfsRawTxnSndrAddrList$ **do**
        $blockNum \leftarrow i[-1]$
        $senderAddress \leftarrow i[0]$
        $ultBlock \leftarrow$ os.path.join($ultBlockDir$, ($blockNum+$
    ".json"))
        $ultBlockCont \leftarrow ultBlock.read()$
        $rawBlockCont \leftarrow$ subprocess.check_output(f"ipfs cat
    $ultBlockCont$["CID"]", shell=true, text=true)
        for each transaction j in rawBlockCont["Transactions"] **do**
            $txnRcvrList \leftarrow j$["Receiver Address"]
            for each receiver address k in txnReceiverList **do**
                if $(k = senderAddress)$ **then**
                    $count \leftarrow count + 1$
                    $rawTxnCont \leftarrow$ subprocess.check_output(f"ipfs cat k["CID"]",
shell=true, text=true)
                    $rawTxnRcvrList \leftarrow rawTxnCont$["Message"]["Receiver Address"]
                    for each receiver address l in $rawTxnRcvrList$ **do**
                        for each receiverAddress m in l.keys() **do**
                            if $(m = senderAddress)$ **then**
                                $allSndrAddrAmt \leftarrow$
                            $allSndrAddrAmt + l[m]$
                            end if
                        end for
                    end for
                end if
            end for
        end for
        $rcvrAddrTotAmt \leftarrow 0$
        for each receiver address i in $ipfsRawTxnRcvrAddrList$ **do**
            for key, val in i.items() **do**
                $rcvrAddrTotAmt \leftarrow rcvrAddrTotAmt + val$
            end for
        end for
        if $(allSndrAddrAmt \geq rcvrAddrTotAmt)$ **then**
            $count \leftarrow count + 1$
        end if
    end for
    if $(count = 4)$ **then return** $ipfsRawTxn$
    end if
end procedure

Algorithm 2 Double Spending Check

Input: Partially Verified Transaction, $ipfsRawTxn$
Output: Completely Verified Transaction, $ipfsRawTxn$

$verifiedSndrList \leftarrow []$
procedure DOUBLESPENDINGCHECK($ipfsRawTxn$)
  for each sender address i in ipfsRawTxnSndrAddrList **do**
    $senderAddress \leftarrow i[0]$
    $connToDB \leftarrow sqlite3.connect(dbDir)$
    $cur \leftarrow connToDB.cursor()$
    $cur.execute$("SELECT * from $table$_1")
    $rows \leftarrow cur$.fetchall()
    $receiverList \leftarrow []$
    for each row k in rows l **do**
      for each tuple item m in rows n **do**
        $receiverList$.append(m)
      end for
    end for
    if (k IS IN $receiverListl$) **then**
      $verifiedSndrList$.append(k)
      $receiverList$.remove(k)
      cur.executemany("DELETE from table_1 where
    rcvr_addr = ?", k)
      $connToDB$.commit()
    end if
  end for
  if ($verifiedSndrList = ipfsRawTxnSndrAddrList$) **then**
    $count \leftarrow count + 1$
    for each receiver address i in $ipfsRawTxnRecvrAddrList$ **do**
      for key, val of i **do**
        $receiverAddress \leftarrow key$
        cur.executemany("INSERT into table_1 VALUES
      (?)", $receiverAddress$)
      end for
    end for
  end if
  if ($count = 4$) **then**
    $mempoolFile \leftarrow$ os.path.join ($mempoolDir, serialNoOfRawTxnFromIpfs$)
    with open($mempoolFile$, "w") as $file$:
    $file$.write($rcvdTxnFromP2P$)
  end if
end procedure

assets to be transferred. Suppose a user uses multiple receiving addresses, among which one of the addressees is not present in the database. Then as a penalty, all the receiving addresses which are valid and used in the transaction being verified will be removed from the database. In short, this test ensures that the address already used to send a certain amount of assets can not be re-utilized in future transactions. Adopting this approach prevents double-spending in our system, and the overall technique is illustrated using Figure 4.3.

Suppose a transaction passes all of the earlier-mentioned tests. In that case, the CID of this verified transaction is added to the mempool, where the verified but unconfirmed and pending transactions get stored. Pseudocodes for authentication, balance and block checks are represented using Algorithm 1 and Algorithm 2 represents the pseudocode for the double-spending check. The overall procedure of transaction verification (including hash block verification, which is discussed in a later section of the same chapter) is graphically represented using Figure 4.4.

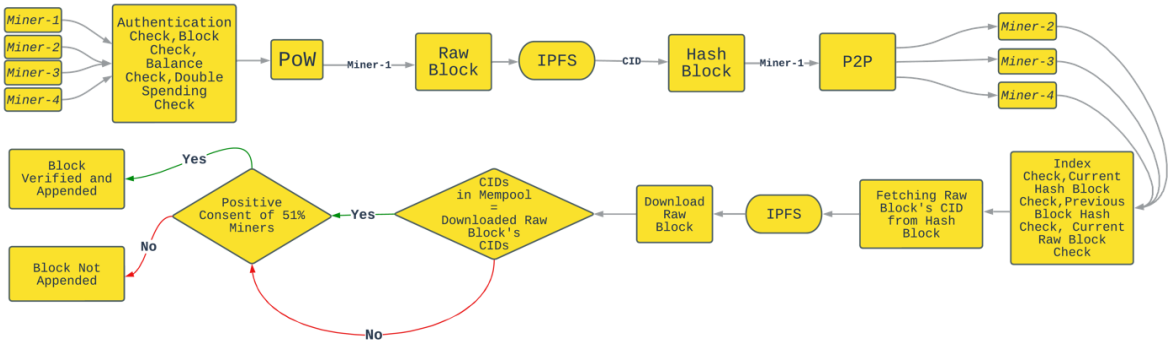


Figure 4.4. Ensuring Security and Decentralization By Dual-Block Verification

4.8 Generating Raw Blocks

After verifying and confirming the transactions from the mempool, miners generate a block that includes all the CIDs of the validated and confirmed raw transactions. This block is denoted as “raw block" and the ledger, which comprises the raw blocks, is referred to as “raw blockchain". This raw blockchain is stored in the IPFS instead of at the miner’s end. Our system employs proof-of-work (PoW) consensus to generate the raw blocks, as this crypto consensus mechanism ensures robust security and decentralization. Figure 4.5 depicts this raw block generation process. As PoW has some drawbacks, which include high-energy consumption, slow and costly, our system is developed in such a way that any consensus is pluggable in it. Therefore, our proposed system supports any consensus algorithm according to the necessity and requirement of the application.

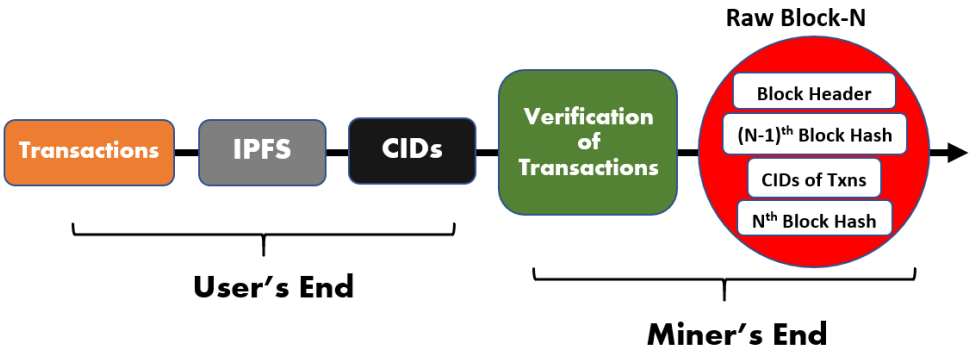


Figure 4.5. Raw Block Generation Using CIDs of Transactions

4.9 Generating Hash Blocks

The following process is executed only by the miner who has generated the raw block, as he is the only authorized person to perform this task: Miner uploads the raw block to IPFS and a CID for this raw block is generated. Using PoW, this miner creates a block that only contains the CID of the uploaded raw block. This block is called “hash block" and the miner distributes this block among the nearby miners using a P2P network. All the miners store this block locally at their end and maintain a new ledger named “hash blockchain". The whole process is demonstrated using Figure 4.6.

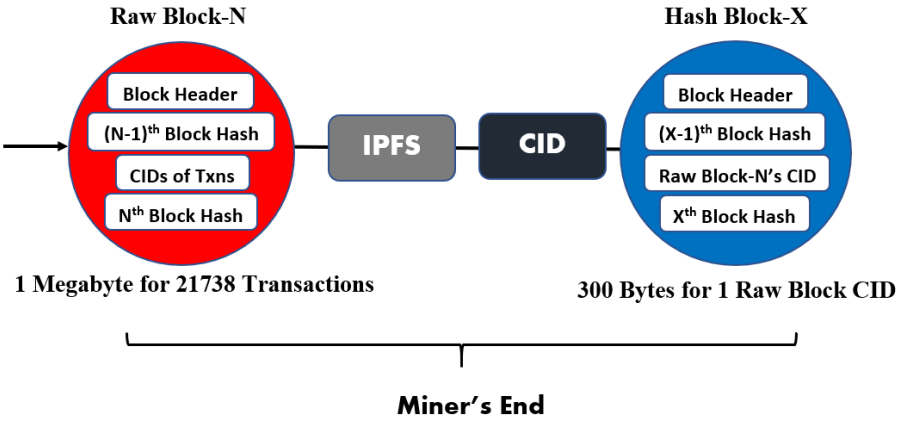


Figure 4.6. Generating Hash Block Using the CID of Raw Block

4.10 Verification of the Hash Block

After receiving the hash block, each miner has to decide whether they will add this block to their local hash blockchain or not. This decision is taken based on the test result of Reverse Block Verification. This verification includes: (i) Index Check, (ii) Current Hash Block Check, (iii) Previous Block Hash Check & (iv) Verified Transaction Check. Firstly, the miner ensures that the current block index is higher than the previous one. Secondly, the block’s hash is examined to ensure that it contains a predetermined number of preceding zeroes. Thirdly, the block hash is calculated to check if the recalculated block hash matches

the previously computed hash added to the block header. After that, the previous block's hash is checked to determine if it matches the value of the 'Previous Hash' attribute in the current block's header. If the hash block passes the first three of the four aforementioned tests, the miner fetches the corresponding raw block from the IPFS using the CID saved inside the hash block. After downloading the raw block, it undergoes the same tests as the hash block. However, an additional test named Verification Transaction Check is required in the case of the raw block to find out whether all the CIDs of the transactions inside this raw block are present in the verifier's mempool or not. All these checks are performed sequentially and Algorithm 3 represents this reverse block verification pseudocode for the hash block. If any block fails to pass one of the verification tests, then the block is discarded and does not get added to the ledger. Upon passing all the tests, the CIDs of the transactions are deleted from the verifying miner's mempool. Finally, the verified hash block is added to the miner's local hash blockchain.

Algorithm 3 Reverse Block Verification: Hash Block

Input: Last Hash Block, $lastUltHashBlock$
Output: Block Removed from Hash Block Directory

$count \leftarrow 0$
procedure REMOVEBLOCK($hashBlockDir, rcvdHashBlockIdx$)
    $rcvdHashBlockLoc \leftarrow$ os.path.join($hashBlockDir$,
    ($rcvdHashBlockIdx$ + ".json"))
    os.remove($rcvdHashBlockLoc$)
    Break the code
end procedure
procedure REVERSEBLOCKVERIFICATION
    $ultHashBlocks \leftarrow os.listdir(ultHashBlockDir)$
    $lastUltHashBlock \leftarrow ultHashBlocks$[-1]
    $lastUltHashBlockIdx \leftarrow lastUltHashBlock$["Index"]
    $lastUltHashBlockCID \leftarrow lastUltHashBlock$["CID"]
    if $(rcvdHashBlockIdx == lastUltHashBlockIdx + 1)$ **then**
        $count \leftarrow count + 1$
    else
        removeBlock($hashBlockDir, rcvdHashBlockIdx$)
        $return$
    end if
    if ($rcvdHashBlockHash[:5]$ == 00000) **then**
        $count \leftarrow count + 1$
    else
        removeBlock($hashBlockDir, rcvdHashBlockIdx$)
        $return$
    end if
    $calcRcvdHashBlockHash \leftarrow$
    hashlib.sha256($rcvdHashBlock$).hexdigest()
    if $(calcRcvdHashBlockHash == rcvdHashBlockHash)$ **then**
        $count \leftarrow count + 1$
    else
        removeBlock($hashBlockDir, rcvdHashBlockIdx$)
        return
    end if
    if $(lastUltHashBlockIdx == rcvdHashBlockPrevHash)$ **then**
        $count \leftarrow count + 1$
    else
        $removeBlock(hashBlockDir, rcvdHashBlockIdx)$
        $return$
    end if
end procedure

Chapter 5

Result Analysis

5.1 Experimental Setup

Our testing environment involved 12 desktop computers, each of them are equipped with AMD Ryzen 7 5800X processor (boosted to 4.7 GHz by overclocking), MSI Gaming GeForce RTX 3060 12 GB 15 Gbps GDRR6 GPU, 32 GB RAM, Transcend 110S 1 TB NVMe M.2 2280 PCle SSD and 50 Mbps 5 GHz Internet. Four of these devices were linked through a same residential Wi-Fi network, while the remaining eight nodes utilized a separate Wi-Fi connection. Both Wi-Fi networks offered an average bandwidth of 100 Mbps and were provided by distinct Internet Service Providers.

5.1.1 Installing IPFS-Go and Running IPFS-Daemon

After downloading IPFS-Go from the official repository, the 'Environment Variables' section of the operating system is updated for the IPFS-client to be found. Then, a Command Prompt is run as the 'Administrator Privilege.' The command 'ipfs init' is executed only once. This command initializes the client settings and make it eligible to run the daemon. After successful initialization, the command 'ipfs-daemon' is executed to make the system connected to other nodes in an IPFS network.

5.1.2 Setting Up Python, Relevant Libraries & IDE

Anaconda 3 (Python 3.8) is installed and PyCharm Community Edition 2022 is used as the IDE. Although most of the essential libraries automatically comes with the Anaconda, three extra libraries which are unavailable in Anaconda are installed: (i) pycoin: this library is used to generate public-private key pairs using SECP256K1 algorithm and also for generating and verifying digital signatures, (ii) p2pnetwork: this library is for creating a peer-to-peer network for sending transactions and blocks across a home/private network, (iii) jsonpickle: this package is used to create and process JSON formatted files using several efficient built-in functions offered by the library. In our system, transactions and blocks are perceived as JSON files.

5.1.3 Construction of a Distributed Database

Our system utilizes and maintains a distributed database among the miners to prevent double spending attack. For this reason, 'sqlite3' package is used to create the database. This database is synchronized across the network using the P2P network developed from the built-in functions of the python library 'p2pnetwork.'

5.1.4 Communication with IPFS with 'subprocess'

To upload and retrieve both transactions and blocks, "ipfs add [fileName]" and "ipfs cat [CID]" commands need to be used, respectively, from the operating system's Command Prompt. To use the Command Prompt directly from the Python interpreter, 'subprocess' library is imported. This package lets the interpreter handles the Command Prompt in the background while executing the codes simultaneously.

5.1.5 Updating the Router Configuration

To send and receive transactions and blocks using the P2P network, the 'Port Forwarding' feature of the router must be turned on. Also, the firewall settings of the operating system must be updated to allow traffic from other PCs. Without applying these necessary changes, no transactions and blocks can be transferred from one user to another. Under the 'Port Forwarding' section, the user must provide the information of IP addresses of the nodes to which they will send data and vice-versa.

5.2 Simulation Parameters

To simulate and evaluate our system, the following parameters were used to test out our proposed technique's efficiency in handling blockchain trilemma:

- Number of Key-Pairs and Addresses: The number of transactions depend on the amount of key-pairs and addresses generated. Every test-cases involved different numnber of transactions and therefore, different combinations of key-pairs and addresses were utilized.
- Raw Block and Hash Block Size: The size of each raw block depends on the number of transactions used in a particular test case. The hash block size is always fixed as it always stores only one CID of the raw block.
- Number of Miners: Number of PCs used in the simulation process refers to the number of miners involved in validating the transactions and mining the block.
- Starting Time & Ending Time: using the ''time' package of the Python, the starting and ending time of each operation executed by our system are recorded. These two values are subtracted to calculate the duration of a particular process.

- Mempool Size: This depends on the number of verified transactions. Invalidated transactions are not kept inside the mempool.
- Throughput: By aggregating the time required by each of the process, the total required time is calculated which is used to divide the number of transactions to get the throughput rate.
- Latency: Latency is calculated using the time taken by IPFS communication overhead.
- Consensus: Any consensus can be plugged into our system. In case of our experiment, we utilized Proof-of-Work consensus.

5.3 Evaluation of Our System

Our proposed system improves the scalability aspect with respect to the storage bloating and throughput issues without compromising the core characteristics of blockchain: decentralization and security. First, we present a theoretical analysis of our system's performance. After that, a practical evaluation is discussed and analyzed. Finally, the theoretical analysis is compared with the practical one and other relevant blockchain frameworks.

5.4 Evaluating Storage Efficiency

Table 5.1. Raw Block Details (Practical)

Content	Size
Header	87 Bytes (Approx.)
Block Hash	64 Bytes
CIDs	966000 Bytes (21000 Txns X 46 Bytes)
Receiver's Address	1344000 Byes (21000 Txns X 64 Bytes)
Indicating Terms for Above Contents	989849 Bytes (Approx.)
Total Size	3.3 Megabytes

Practically for 21000 transactions, a raw block size of 3.3 MB is required. The header of the block and block hash take up 87 and 64 bytes of storage, respectively. Each receiver's address requires 46 Bytes and every CID is 64 bytes in size. The indexing terms for header, block hash, CIDs, receiver addresses and all of the 21000 transactions occupy almost 989849 Bytes of the raw block. A detailed breakdown of these storage requirements is presented in Table 5.1.

In the case of hash blocks, the header, block hash and the CID of the raw block take 86, 64 and 46 bytes, respectively. The indexing terms for the aforementioned fields require approximately 87 bytes. Each of the hash blocks thus needs 283 bytes of storage. A summary of these storage requirements for hash blocks can be found in Table 5.2.

Table 5.2. Hash Block Details (Practical)

Content	Size
Header	86 Bytes (Approx.)
Block Hash	64 Bytes
CID	46 Bytes
Indicating Terms for Above Contents	87 Bytes (Approx.)
Total Size	283 Bytes

Two different types of ledgers exist in our system: (i) raw blockchain and (ii) hash blockchain. The first one consists of raw blocks containing CIDs of multiple verified transactions and this raw blockchain resides in IPFS and not on the miner's end. On the other hand, the hash blocks contain only a single CID of a particular raw block which is 46 Bytes in size and including the metadata, the size per hash block becomes approximately 300 Bytes. As the size of the hash blockchain or hash blocks are relatively way too small compared to the Raw one and each miner only needs to store a copy of the hash blockchain at their end and not the ledger of raw blocks, the storage requirement is remarkably low in our system.

Table 5.3. Comparison of Bitcoin and Proposed System with Respect to Total Size of Ledger

Total Number of Blocks in Bitcoin (Block Height)	736930 Blocks
Total Number of Transactions in Bitcoin Height	734968323 Txns
Bitcoin Blockchain	
Block Size	1 MB (average)
Total Size Requirement	687 Gigabytes
Our Proposed Blockchain	
Block Size	300 Bytes
Total Size Requirement (At Miner's End)	0.206 Gigabytes
Total Size Requirement (At IPFS)	33.018 Gigabytes
Multiplication Factor of the Proposed System	3335 Times Less

The size of a block of Bitcoin blockchain is approximately 1 MB on average. Currently, the bitcoin blockchain height is 736930 blocks. Consequently, the total size of the Bitcoin ledger is around 687 GB. On the other hand, the size per block in our system is 300 Bytes and this size remains constant. Compared with the Bitcoin blockchain, our system only requires 0.206 GB for the block height of Bitcoin. Therefore, our system requires almost 3335 times less storage than the Bitcoin blockchain. This comparison of storage requirements is summarized in Table 5.3.

5.4.1 Evaluationg Throughput

Raw blocks contain the CIDs of the verified transactions. Each CID has a constant size of 46 Bytes and does not vary according to the size of the transactions. The size of CID is considerably smaller compared to the actual transaction size. Consequently, we can add more CIDs in a block of one megabyte (considered the standard size of a block in the Bitcoin blockchain) compared to the number of raw transactions.

There are approximately 734,968,323 transactions stored inside 736930 blocks, which is the current Bitcoin blockchain height. Therefore, 997 transactions are contained in each block, on average. Thus the throughput of the Bitcoin blockchain is around two transactions/second, according to equation 5.1.

$$Throughput = \frac{NumberofTransactionsPerBlock}{BlockGenerationTimeinSeconds} \tag{5.1}$$

This amount of transactions processed per second is significantly lesser than that of centralized and popular financial services like VISA, PayPal, etc. Following equation 5.2 on the contrary, our proposed blockchain is capable of holding around 21738 transactions per block, where each CID is of size 46 bytes, block size is same as the Bitcoin's one, which is one megabyte on average and the size of the block header is 80 bytes.

$$TxnsPerBlock = \frac{SizeofEachBlock - BlockHeaderSize}{PerCIDSize} \tag{5.2}$$

Table 5.4. Comparison of Bitcoin and Proposed System with Respect to Number of Transactions Per Block

Bitcoin Blockchain	
Block Header Size	80 Bytes
Per Block Size (On Average)	1 MB
Per Transaction Size (On Average)	250 Bytes
Number of Transactions Per Block	997 Transactions
Our Proposed Blockchain	
Block Header Size	80 Bytes
Per (Raw) Block Size (On Average)	1 MB
Per Transaction (CID) Size (On Average)	46 Bytes
Number of Transactions Per (Raw) Block	21738 Transactions
Multiplication Factor of the Proposed System	22 Times More

It is apparent from Table 5.4 that each block of our system can hold 22 times more transactions than the Bitcoin blockchain. This multiplying factor can be further increased

if the block size is extended. As the size of the hash block remains constant despite incrementing the raw block size, increasing the block size does not originate the storage bloating issue of Bitcoin in our system.

Our blockchain's integration with IPFS introduces a crucial aspect that impacts its overall throughput. This interaction with IPFS becomes particularly evident during various stages of the blockchain's operation.

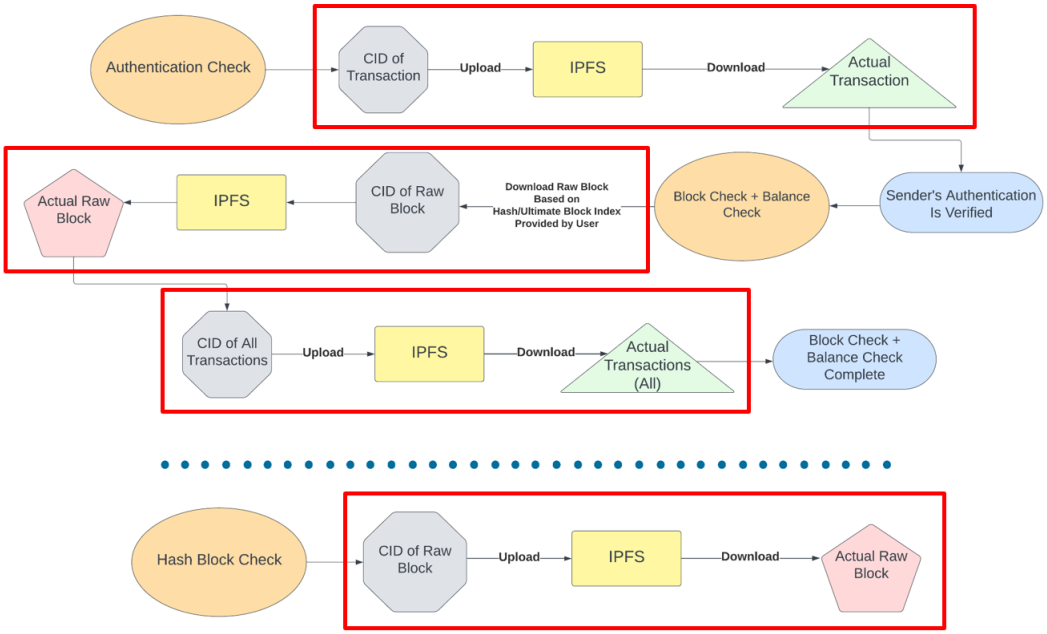


Figure 5.1. IPFS Overhead in Terms of Both Transaction + Block Verification

During the Authentication Check process, our blockchain establishes communication with IPFS to retrieve the precise transactions by utilizing Content IDs (CIDs). This step ensures the validity and authenticity of transactions, requiring the blockchain to access IPFS for essential data retrieval. This IPFS complexity is depicted using Figure 5.1.

Moving on to the Block Check and Balance Check phases, a similar pattern emerges. In these instances, IPFS receives the CID of the raw block, prompting the download of the actual Raw Block itself. Upon obtaining the Raw Block, the blockchain extracts all associated CIDs, initiating another round of communication with IPFS to retrieve the complete set of actual transactions referenced by these CIDs. This Block-Balance check overhead is graphically represnted in Figure 5.2

Even during the Hash Block Checking stage, the blockchain's reliance on IPFS remains evident. Here, the blockchain must once again connect to IPFS to fetch the corresponding Raw Block, using the provided CID for verification purposes.

Evidently, the repeated interactions with IPFS throughout the various stages of block verification result in a cumulative effect – an IPFS overhead that directly impacts the

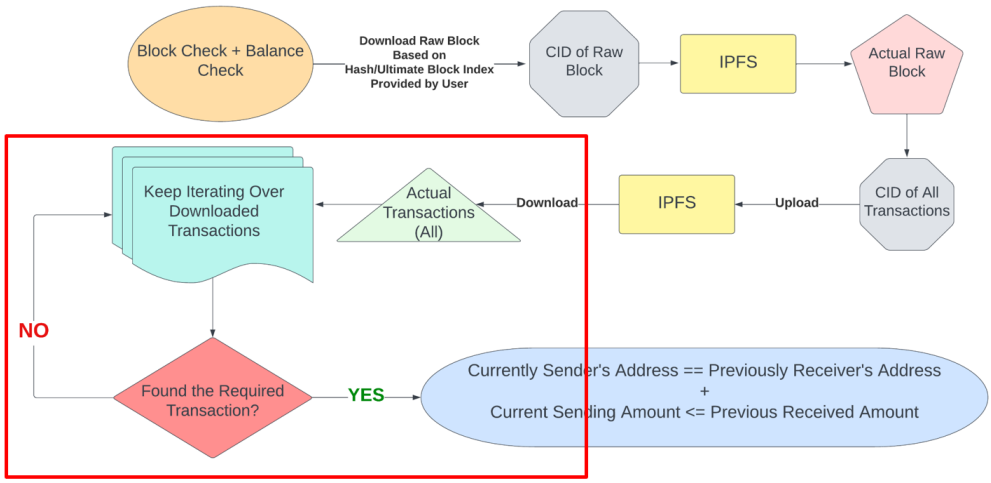


Figure 5.2. Overhead in Transaction Verification (Block Check + Balance Check)

blockchain's throughput. This overhead arises from the need to establish connections, transmit data, and await responses from IPFS multiple times during the verification process. Consequently, the efficiency and speed at which the blockchain processes transactions and maintains its integrity are inevitably influenced by this IPFS overhead.

The results of the performance analysis of our dual-blockchain system are summarized in the Table 5.5. This table indicates the system's throughput as measured across different transaction volumes, including the significant overhead introduced by the IPFS in fetching and verifying transactions.

For example, when processing 21,000 transactions, the IPFS fetching time was around 721 seconds, and the total verification time was approximately 1,127 seconds. The raw block generation took around 862 seconds, while uploading the raw block to the IPFS took only about 0.45 seconds. The hash block generation time was nearly negligible at around 0.005 seconds. The total time required for all these processes, which run concurrently, was approximately 1,990 seconds. This resulted in a transaction per second (TPS) rate of around 10.55 for our proposed system, compared to a TPS of 35 for a conventional blockchain system. When the system was scaled up to handle 40,000 transactions, the

Table 5.5. TPS Rate and Its Relation to IPFS and Verification Overheads

Amount of Txns	IPFS Fetching Time (in Seconds)	Total Verification Time (in Seconds)	Raw Block Generation Time (in Seconds)	Time to Upload Raw Block to IPFS (in Seconds)	Hash Block Generation Time (in Seconds)	Total Time Required (in Seconds) (Processes Are Concurrent)	Proposed System's TPS	Conventional Blockchain's TPS	Required Raw Block Size (in MB)	Required Conventional Block Size (in MB)
21000	720.878082	1127.375601	861.767147	0.44999	0.004526	1989.601038	10.554880	35	3.3	11.97
25000	866.961301	1308.793142	924.375239	0.484211843	0.004870203	2235.239331	11.184485	41.67	3.93	14.25
30000	939.837729	1414.180346	983.210614	0.51024	0.004818	2397.249796	12.514340	50	4.62	17.10
35000	1004.797376	1476.985091	1007.654082	0.570491	0.005109	2483.773183	14.091464	58.33	5.23	19.95
40000	1076.757023	1548.789835	1031.097549	0.630741	0.005401	2575.414701	15.531479	66.67	6.3	22.8
45000	1132.716674	1611.594584	1082.541016	0.690992	0.005692	2693.796562	16.705048	75	7.09	25.65
50000	1196.676317	1701.399325	1106.984484	0.751242	0.005984	2807.345883	17.810417	83.33	7.88	28.5
55000	1239.635964	1813.204071	1173.427951	0.811493	0.006276	2986.872164	18.413912	91.67	8.67	31.35
60000	1297.595611	1947.008815	1216.871418	0.871743	0.006567	3164.643648	18.959481	100	9.46	34.20
65000	1346.555258	2053.813596	1298.314885	0.931994	0.006859	3352.723019	19.391408	108.33	10.25	37.05

IPFS fetching time increased to about 1,077 seconds, and the total verification time reached approximately 1,549 seconds. The raw block generation time also increased to roughly 1,031 seconds, while the time required to upload the raw block to IPFS was about 0.63 seconds. Again, the hash block generation time was negligible at approximately 0.005 seconds. The total time required for these concurrent processes was about 2,575 seconds, resulting in a TPS of approximately 15.53 for our system, compared to a TPS of 66.67 for a conventional blockchain system.

By observing the column values of “IPFS Fetching Time (in Seconds)" and “Total Verification Time (in Seconds)" it is evident that the process of fetching transactions from the IPFS and verifying them constitutes a significant portion of the total processing time. For instance, when processing 25,000 transactions, the IPFS fetching time was approximately 867 seconds, and the total verification time was about 1,309 seconds. Similarly, for 45,000 transactions, the IPFS fetching time rose to nearly 1,133 seconds, and the total verification time extended to about 1,612 seconds. Another important aspect highlighted by the table is the time consumed by the Proof-of-Work process, which can be observed from the column values of “Raw Block Generation Time (in Seconds)". For example, the raw block generation time, which essentially represents the Proof-of-Work stage, was approximately 983 seconds for 30,000 transactions and roughly 1,082 seconds for 45,000 transactions. This indicates that the Proof-of-Work stage is a time-intensive process, contributing significantly to the overall processing time.

The table clearly shows that while our system’s TPS is lower than that of a conventional blockchain, it significantly reduces the block size required for storing the transactions. For instance, the size of the required raw block for 21,000 transactions was only 3.3 MB, while the size required for a conventional block was approximately 11.97 MB. As the number of transactions increased, the size of the required raw block also increased, but it remained significantly smaller than the size of the conventional block. For instance, for 40,000 transactions, the required raw block size was about 6.3 MB, compared to 22.8 MB for a conventional block.

5.4.2 Security Evaluation

We analysed the security of our proposed system using three formal measurements, and compares it with other consensus strategies, such as uniform tie-breaking (UTB), smallest-hash tie-breaking (SHTB), unpredictable deterministic tie-breaking (UDTB), and Publish or Perish (PoP), Fruitchains, Reward-Splitting Protocol (RS) and Subchains:

1. Chain Quality
2. Incentive Compatibility

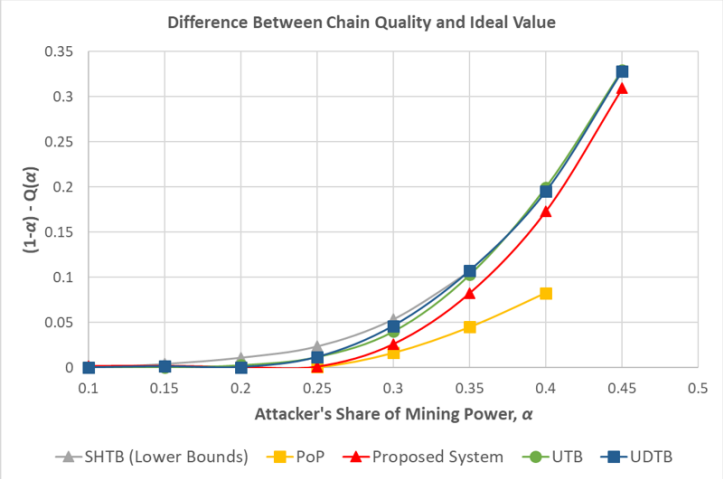
3. Subversion Gain

Chain quality is a crucial metric for evaluating the robustness of a consensus protocol. It quantifies the challenge in substituting honest main chain blocks by an adversary. Equation 5.3 is used to calculate any blockchain's chain quality.

$$Q(\alpha) = \min_{s} \lim_{t \to \infty} \frac{B_c}{B_a + B_c} \tag{5.3}$$

Here, B_c and B_a are the total number of main chain blocks mined by compliant miners and the adversary, respectively, and s represents the adversary's strategy. Besides α, another critical input in our proposed system is γ, representing the proportion of compliant mining power that works on the adversary's chain during a tie. We calculate $Q(\alpha)$ for all five protocols and the graph in Figure 5.3 represents the chain quality of our system, along with these consensus protocols. The $Q(\alpha)$ of UTB and UDTB are nearly identical and perform no better than our Proposed System. Their chain quality is lower than our Proposed System. On the contrary, only PoP has better chain quality compared to our approach.

Figure 5.3. Comparison of Our System with Other Models Based on Chain Quality



Incentive Compatibility is a crucial metric used to assess a protocol's resistance against selfish mining. It represents the expected minimum revenue that compliant miners can achieve. The value of this metric for a blockchain system can be measured using Equation 5.4.

$$I(\alpha) = \min_{s} \lim_{t \to \infty} \frac{\sum R_c}{\sum R_a + \sum R_c} \tag{5.4}$$

To calculate this metric, we utilize the cumulative rewards received by both the attacker ($\sum R_a$) and the compliant miners ($\sum R_c$). To evaluate the attack resistance of our proposed

system, as well as three other influential designs, namely Fruitchains, Subchains, and Reward-Splitting protocol (RS), we have conducted a thorough analysis. The comparison results have been tabulated in three distinct tables: Table 5.6, 5.7 and 5.8, with entries highlighted in red italic when they perform worse than our proposed system. We considered three parameters for the aforementioned protocols to be evaluated against our system. A block becomes visible if the time gap between its creation and the current time is strictly smaller than the defined Timeout Threshold, T_0. Ratio of Fruit Difficulty Target to Block Difficulty Target, $Ratio_{f2b}$ represents the ratio between the difficulty target of the fruit chain and the difficulty target of the block chain. Compliant Mining Proportion, γ is the proportion of compliant mining power that actively participates in mining on the attacker chain during a tie situation.

Table 5.6. I(α) Of FRUITCHAINS

(T_0, $Ratio_{f2b}$, γ)	α=**0.15** (**0.8357**)	α=**0.2** (**0.7794**)	α=**0.25** (**0.7544**)	α=**0.3** (**0.6821**)	α=**0.35** (**0.6153**)
(7,1,0)	*0.8494*	*0.7961*	*0.7656*	*0.6914*	*0.6358*
(7, 1, 1)	0.8193	0.7556	0.7337	0.6557	0.5932
(13,1,0)	0.7500	*0.7822*	*0.7772*	*0.6864*	*0.6168*
(13,1,1)	0.7500	0.7597	0.7470	0.6754	0.6036
(13,2,0)	0.7500	0.7597	0.7472	0.6766	0.6072
(13,2,1)	0.7500	0.7597	0.7470	0.6756	0.6040
(13,0.5,0)	0.7500	*0.8107*	*0.7772*	*0.6864*	*0.6265*
(13,0.5,1)	0.7500	0.7597	0.7470	0.6653	0.6033

Table 5.7. I(α) Of RS

(T_0, γ)	α=**0.3** (**0.6821**)	α=**0.35** (**0.6153**)	α=**0.4** (**0.5736**)	α=**0.45** (**0.5497**)
(3,0)		0.6084	0.4842	0.3097
(3,0.5)		0.5997	0.4534	0.2575
(3,1)	0.6527	0.5771	0.4292	0.2406
(6,0)			0.5283	0.3454
(6,0.5)			0.5056	0.2945
(6,1)		0.6097	0.4899	0.2816
(9,0)			0.5566	0.3690
(9,0.5)			0.5388	0.3210
(9,1)			0.5269	0.3098

When considering the metric I(α), it has been observed that Fruitchains underperforms in comparison to our Proposed System for various parameter choices when $\gamma = 0$. However, Fruitchains outperforms the NC (Non-Cooperative) strategy when $\gamma = 1$. Furthermore, our Proposed System demonstrates its superiority over the Reward-Splitting protocol (RS) in every possible parametric combination.

Subversion gain is a vital metric assessing the profitability of double-spending attacks. The

Table 5.8. I(α) Of Subchains

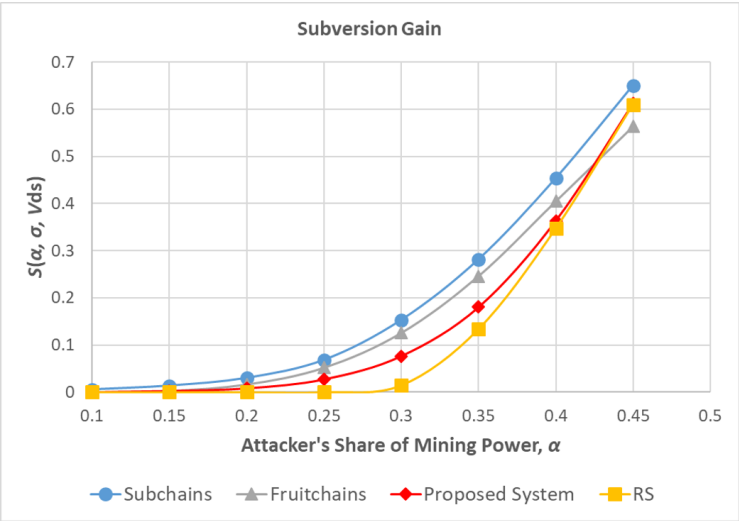
$(Ratio_{w2b}, \gamma)$	α=**0.1** **(0.8613)**	α=**0.15** **(0.8357)**	α=**0.2** **(0.7794)**	α=**0.25** **(0.7544)**	α=**0.3** **(0.6821)**
(2,0)	0.8990	0.8467	0.7922	0.7642	0.6912
(2,0.5)	0.8470	0.8326	0.7753	0.7241	0.6570
(2,1)	0.8489	0.8235	0.7500	0.6667	0.5714
(3,0)	0.8987	0.8456	0.7895	0.7588	0.6813
(3,0.5)	0.8960	0.8401	0.7804	0.7556	0.6732
(3,1)	0.8589	0.8235	0.7500	0.6667	0.5714

measurement is based on the time-averaged illegal upper bound profit in a specific attack model. In this model, every honest block contains a payment transaction to the merchant. A conflicting version of this transaction is embedded in the block's secret competitor, provided such a competitor exists. The subversion gain of the attacker is defined as:

$$S(\alpha, \sigma, V_{ds}) = \max_{s} \lim_{t \to \infty} \frac{\sum R_a + \sum R_{ds}}{t} - \alpha \tag{5.5}$$

In this equation, t represents the duration, measured in the number of block generation intervals, and α is the time-averaged mining reward in the absence of a double-spending attack. The analysis of the subversion gain $S(\alpha, \sigma, V_{ds})$ of our proposed system, illustrated in Figure 5.4, reveals interesting findings when compared to Fruitchains and Subchains. The subversion gains of these two protocols are higher than that of the proposed System, and that of RS is lower. In other words, Fruitchains and Subchains perform worse than our system, while RS surpasses our protocol.

Figure 5.4. Comparison of Our System's Subversion Gain with Other Blockchains



5.4.3 Evaluation of Decentralization

The degree of decentralization of any blockchain protocol is influenced by several factors, which vary depending on the consensus mechanism and the specific nature of the protocol. Our proposed system has been tested with the proof-of-work consensus mechanism, with the cost of a mining node being around $950 (this includes the cost of an AMD Ryzen 7 5800X and an MSI GeForce RTX 3060 12GB). The governance of our system is off-chain, and it is a public blockchain.

One of the key advantages of our system is its scalability: it can handle an average of 15.52 transactions per second (TPS) on average, which is eight times more than Bitcoin. Moreover, the storage requirement of our system is significantly lower than Bitcoin, with each block requiring only 283 bytes, which is approximately 43478 times less than Bitcoin. In terms of resilience, our system is resistant to various types of attacks, including double-spending, Sybil, DoS, timejacking, long-range, and selfish-mining attacks when the proof-of-work consensus mechanism is utilized. Given these advantages, we believe that our system has the potential to achieve higher decentralization than existing blockchain protocols. The combination of high scalability, low storage requirements, resilience to various types of attacks, and low mining node cost makes our system an attractive option for nodes interested in joining a blockchain network. Even though we have only tested our system with 10 nodes, we believe that the level of decentralization of our proposed system has the potential to surpass that of Bitcoin, Ethereum, Litecoin, and other leading blockchain protocols.

In our study, a broad range of blockchain systems were evaluated for their level of decen-

Table 5.9. Evaluating The Proposed System's Decentralization using Parameters for Measuring Decentralization and Comparing It with Established Blockchains

Blockchain System	Consensus	Mining Node Cost (USD)	Number of Active Nodes	Geographical Distribution of Nodes	Presence of Mining Pools	Governance	Nature	Throughput (TPS)	Block Size	Resilience to Attacks	Censorship Resistance	Decentralization Score (0-10)
Proposed Blockchain	Pluggable	$950 (Ryzen 7 5800X + RTX 3060)	12	Global	No	Off-chain	Public	15.52	283 Bytes	Resistant to various attacks (assuming PoW)	High	7.27
Bitcoin	Proof-of-Work (PoW)	$5,000-15,000$ (ASICs required)	10,000+	Global	Yes	Off-chain	Public	3-7	1MB	Resistant to various attacks (PoW)	High	7.73
Ethereum	Proof-of-Work (PoW)	$5,000-15,000$ (ASICs required)	30,000+	Global	Yes	Off-chain	Public	15-20	20-30 KB	Resistant to various attacks (PoW)	High	7.95
Bitcoin Cash	Proof-of-Work (PoW)	$5,000-15,000$ (ASICs required)	3,000+	Global	Yes	Off-chain	Public	60	32 MB	Resistant to various attacks (PoW)	Medium	6.82
Bitcoin SV	Proof-of-Work (PoW)	$5,000-15,000$ (ASICs required)	1,000+	Global	Yes	Off-chain	Public	100	128 MB	Resistant to various attacks (PoW)	Medium	6.36
Dash	Proof-of-Work (PoW)	$2,000-4,000$ (CPU/GPU mining)	1,500+	Global	Yes	Off-chain	Public	56	2 MB	Resistant to various attacks (PoW)	Medium	7.27
NEO	Delegated Byzantine Fault Tolerance (dBFT)	$5,000-10,000$ (BFT nodes)	20	Global	No	On-chain (dBFT)	Public	1,000	2 MB	Resistant to various attacks (dBFT)	Low	5.91
Tezos	Liquid Proof-of-Stake (LPoS)	$5,000-10,000$ (Bakers)	100	Global	No	On-chain (LPoS)	Public	30	512 KB	Resistant to various attacks (LPoS)	Low	5.68
Nano	Delegated Proof-of-Stake (dPoS)	$100-500$ (Representatives)	1,000+	Global	Yes	On-chain (dPoS)	Public	1,000	4 MB	Resistant to various attacks (dPoS)	High	6.59
IOTA	Tangle (Directed Acyclic Graph)	$100-500$ (CPU/GPU)	500+	Global	Yes	Off-chain	Public	1,000	1 MB	Resistant to various attacks (Tangle)	High	6.82

tralization, including our blockchain system, Bitcoin, Ethereum, Bitcoin Cash, Bitcoin SV, Dash, NEO, Tezos, Nano, and IOTA; which is represented in Table 5.9. The decentralization score was calculated based on eleven factors, with a range from 0 to 10.

The user's blockchain system achieved a decentralization score of 7.27, which is comparable to many established blockchain systems such as Bitcoin (7.73), Ethereum (7.95), and Dash (7.27). This score suggests that the user's blockchain has a fairly high level of decentralization. Among the established blockchain systems, Ethereum scored the highest in terms of decentralization (7.95), followed closely by Bitcoin (7.73). The relatively high decentralization scores of Ethereum and Bitcoin can be attributed to their use of Proof of Work consensus, global distribution of nodes, off-chain governance, public nature, resilience to various attacks, and high censorship resistance. On the other end of the spectrum, NEO and Tezos scored the lowest in terms of decentralization, with scores of 5.91 and 5.68, respectively. Their lower scores can be attributed to their use of on-chain governance and lower censorship resistance compared to other blockchain systems.

In general, the results suggest that the choice of consensus mechanism, cost of mining nodes, number and geographical distribution of active nodes, presence of mining pools, type of governance, nature of the blockchain, throughput, block size, resilience to attacks, and censorship resistance all play significant roles in determining the level of decentralization in a blockchain system.

Additionally, we believe that if our system is practically implemented with a proof-of-stake consensus mechanism, it would achieve even higher levels of decentralization. Proof-of-stake does not require validators to calculate block hashes with a predefined number of zeros, a process that consumes a significant amount of time in proof-of-work consensus.

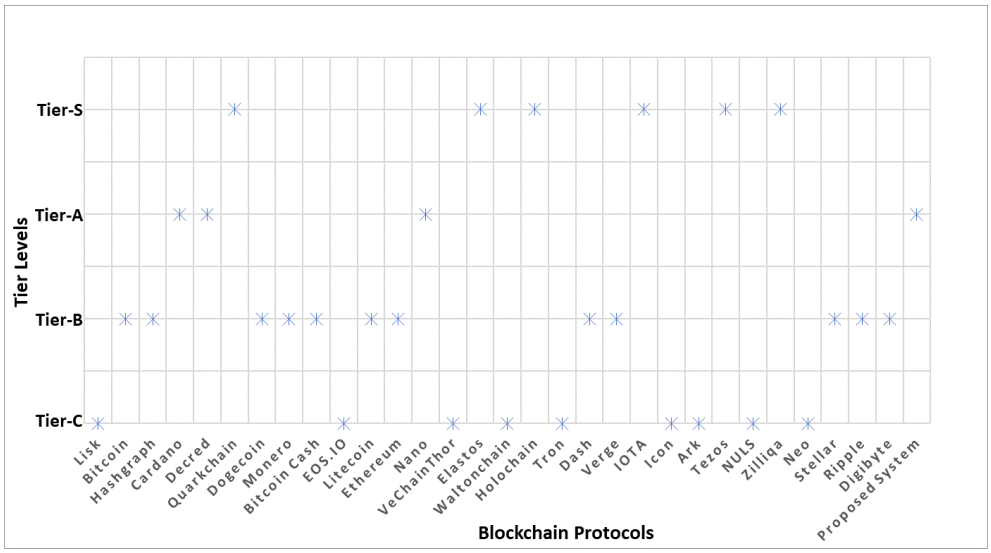


Figure 5.5. Decentralization Levels of Various Blockchain-Based Platforms

Therefore, the hardware requirements for running a proof-of-stake consensus on our system would be much less demanding and costly than those for a proof-of-work consensus, further contributing to its potential for high levels of decentralization.

The scatter chart in Figure 5.5 represents some tiered blockchain protocols categorized based on their level of decentralization. Tier-S provides the topmost decentralization and the level decreases while descending to the lower tiers. Tier-C offers the minimum decentralization and this degree of decentralization is measured by several characteristics of a blockchain system, some of which are presented in Table 5.9. Some of the notable features for the categorization are the number of nodes, permissioned or permissionless, trustless or non-trustless, the number of entities dominating the mining process, type of consensus, etc. For example, two mining pools control 51% computational power in Bitcoin-Cash, whereas the same mining power is distributed among six entities in Nano. Consequently, Bitcoin-Cash falls under a lower tier (Tier-B) than Nano (Tier-A). On the other hand, systems under Tier-S are controlled by millions of entities and some of them utilize sharding-based consensus, thus offering the upmost and excellent decentralization. As our system supports both proof-of-work and proof-of-stake and as sharding can also be exploited by implementing our system in different local chains, our system can be placed between Tier-B and Tier-S.

Finally, we have compared our blockchain with some of the recent contributions which aim to solve blockchain trilemma. From Table 5.10, it is observable that our system can be considered as the most complete trilemma solver as its capability to keep a balance among security, scalability and decentralization surpasses the existing mechanisms. In addition, some of these works are theoretical but we have demonstrated the practical analysis, which keeps our system ahead of the other approaches.

Table 5.10. Comparison of Our System with Recent Works Intend to Solve Blockchain Trilemma

Title of Contribution	Technique	Throughput	Block Size	Consensus	Security	Decentralization
Scalable Blockchain Storage Model Based on DHT and IPFS [43]	IPFS + Chord	Not Calculated and Discussed	400 KiloBytes	No Mention of Consensus	Security is not evaluated	No performance metrics other than block size is discussed. Therefore, decentralization cannot be measured
A Novel Scheme to Improve the Scalability of Bitcoin Combining IPFS With Block Compression [53]	IPFS	4.605 TPS	14.86 KiloBytes	Proof-of-Work	Double Spending, Sybil, Selfish Mining etc. due to PoW utilization	Better decentralization than Bitcoin but less TPS and higher block size compared to our system, hence comparatively low decentralization than proposed system
Trifecta: the Blockchain TriLemma Solved [45]	Sharding	250,000 TPS (Minimum 5 Shards and 300 Nodes Required)	No mention of block size	Sharding + Longest Chain Consensus	Malicious actors controlling an individual shard could disrupt its transactions and could even shut it down	Medium (sharding involves large number of nodes, but Cross-Shard transactions are complex and introduce latency)
SymBChainSim: A Novel Simulation Tool for Dynamic and Adaptive Blockchain Management and its Trilemma Tradeoff [54]	Simulation tool for dynamic updates of blockchain parameters and consensus	600 seconds per block	1 MegaByte	PBFT, Bigfoot	In case of Bigfoot, only a single faulty node drastically reduces the throughput	Although it intends to solve Trilemma, there is no mention of decentralization
Scaling Blockchains Without Giving up Decentralization and Security [46]	Pipelining (computation can be distributed across several committees)	Theoretical solution, therefore not mentioned	Theoretical solution, therefore not calculated	Randomized committee-based consensus	Committees are randomly selected, making it harder for the attackers to control committee members	All nodes cooperate in the creation of a new block, therefore better decentralization is offered
Our System	IPFS + Kademila	15 TPS	283 Bytes	Proof-of-Work	Double Spending, Sybil, Selfish Mining etc. due to PoW utilization. Can defend Transaction Privacy Leakage using distributed database	Better Decentralization Level Than Bitcoin Cash, Dash, NEO, Tezos, Nano, IOTA etc.

Chapter 6

Conclusion

The proposed system achieves an average throughput rate of 10.55 transactions per second for 21000 transactions in a 3.3 MB block and this rate can be increased to a great extent by expanding the size of raw blocks. Our system also requires substantially small storage, as only 283 bytes are needed for each block, regardless of the number of transactions. These two factors increase the scalability of the system and due to its extremely low storage dependency, public nature, low-cost mining node and off-chain governance, the decentralization level is escalated and a large number of participants is ensured. As more and more participants are encouraged to join the network due to the system's aforementioned features to tackle the limitations and drawbacks of popular public blockchains like Bitcoin, it becomes easy to prevent 51% and Sybil attacks. Some unique countermeasures for establishing defense against double-spending, eclipse, selfish mining, private key security, DDoS attacks, etc., are introduced and thus, security is also guaranteed. The throughput can be further increased by establishing sharding or payment channels on top of our protocol, which can be considered an open and future research direction. Also, any scalable consensus other than proof-of-work can be plugged into our system for better scalability but with the expense of security and decentralization.

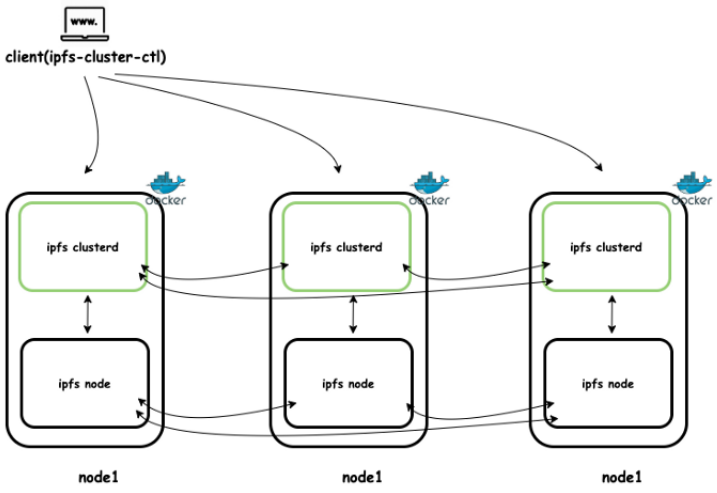


Figure 6.1. Running Multiple IPFS Daemon to Reduce IPFS Overhead

To address the challenges posed by IPFS overhead and optimize the process of fetching CIDs from IPFS, employing multiple IPFS Daemons within a single PC can significantly improve efficiency. By distributing the workload across multiple daemons, the retrieval process becomes more streamlined, reducing latency and enhancing overall performance. This technique is illustrated in Figure 6.1. Furthermore, to expedite the verification of blocks in the system, leveraging multi-threading capabilities can prove highly beneficial. Running multiple Python processes in parallel enables the verification of multiple blocks simultaneously. This parallel processing approach takes full advantage of modern multi-core processors, maximizing computational resources and drastically reducing verification times, which is depicted in Figure 6.2.

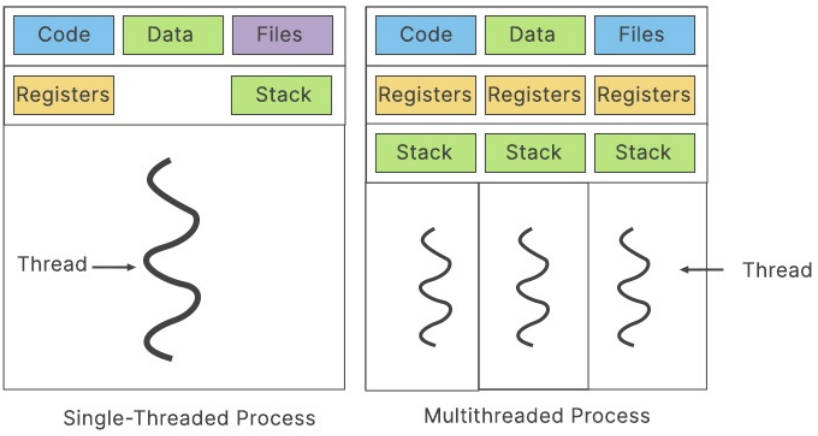


Figure 6.2. Utilizing Multiple Threads to Speed Up Verification

Bibliography

[1] M. Xu, X. Chen, and G. Kou, "A systematic review of blockchain," *Financial Innovation*, vol. 5, no. 1, pp. 1–14, 2019.

[2] M. Padmavathi and R. Suresh, "Secure p2p intelligent network transaction using litecoin," *Mobile Networks and Applications*, vol. 24, no. 2, pp. 318–326, 2019.

[3] A. Vacca, A. Di Sorbo, C. A. Visaggio, and G. Canfora, "A systematic literature review of blockchain and smart contract development: Techniques, tools, and open challenges," *Journal of Systems and Software*, vol. 174, p. 110 891, 2021.

[4] Q. Wang, X. Zhu, Y. Ni, L. Gu, and H. Zhu, "Blockchain for the iot and industrial iot: A review," *Internet of Things*, vol. 10, p. 100 081, 2020.

[5] P. Dunphy, "A note on the blockchain trilemma for decentralized identity: Learning from experiments with hyperledger indy," *arXiv preprint arXiv:2204.05784*, 2022.

[6] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *Ieee Access*, vol. 8, pp. 16 440–16 455, 2020.

[7] S. Aggarwal and N. Kumar, "Blockchain 2.0: Smart contracts," in *Advances in Computers*, vol. 121, Elsevier, 2021, pp. 301–322.

[8] D. D. F. Maesa and P. Mori, "Blockchain 3.0 applications survey," *Journal of Parallel and Distributed Computing*, vol. 138, pp. 99–114, 2020.

[9] A. Singh, R. M. Parizi, M. Han, A. Dehghantanha, H. Karimipour, and K.-K. R. Choo, "Public blockchains scalability: An examination of sharding and segregated witness," in *Blockchain Cybersecurity, Trust and Privacy*, Springer, 2020, pp. 203–232.

[10] Y. Kwon, H. Kim, J. Shin, and Y. Kim, "Bitcoin vs. bitcoin cash: Coexistence or downfall of bitcoin cash?" In *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2019, pp. 935–951.

[11] K. John, M. O'Hara, and F. Saleh, "Bitcoin and beyond," *Annual Review of Financial Economics*, vol. 14, 2021.

[12] Y. Peng, X. Yang, and H. Zhou, "Blockchain technology and higher education: Characteristics, dilemma and development path," in *2020 The 4th International Conference on Education and E-Learning*, 2020, pp. 173–176.

[13] A. Urquhart, "The inefficiency of bitcoin," *Economics Letters*, vol. 148, pp. 80–82, 2016.

[14] F. Rezaeibagha and Y. Mu, "Efficient micropayment of cryptocurrency from blockchains," *The Computer Journal*, vol. 62, no. 4, pp. 507–517, 2019.

[15] C. Decker and R. Wattenhofer, "A fast and scalable payment network with bitcoin duplex micropayment channels," in *Symposium on Self-Stabilizing Systems*, Springer, Cham, 2015, pp. 3–18.

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

[17] J. Lind, O. Naor, I. Eyal, F. Kelbert, E. G. Sirer, and P. Pietzuch, "Teechain: A secure payment network with asynchronous blockchain access," in *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, 2019, pp. 63–79.

[18] G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate, and M. Maffei, "Anonymous multi-hop locks for blockchain scalability and interoperability," *Cryptology ePrint Archive*, 2018.

[19] H. Hees, "Raiden network: Off-chain state network for fast dapps," in *Devcon two*, Ethereum Foundation, 2016.

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

[21] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in *2018 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2018, pp. 583–598.

[22] E. Syta, P. Jovanovic, E. K. Kogias, *et al.*, "Scalable bias-resistant distributed randomness," in *2017 IEEE Symposium on Security and Privacy (SP)*, Ieee, 2017, pp. 444–460.

[23] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 931–948.

[24] K. R. Özyılmaz, H. Patel, and A. Malik, "Split-scale: Scaling bitcoin by partitioning the utxo space," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, IEEE, 2018, pp. 41–45.

[25] L. Kan, Y. Wei, A. H. Muhammad, W. Siyuan, L. C. Gao, and H. Kai, "A multiple blockchains architecture on inter-blockchain communication," in *2018 IEEE international conference on software quality, reliability and security companion (QRS-C)*, IEEE, 2018, pp. 139–145.

[26] W. Li, A. Sforzin, S. Fedorov, and G. O. Karame, "Towards scalable and private industrial blockchains," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, 2017, pp. 9–14.

[27] S. Malik, S. S. Kanhere, and R. Jurdak, "Productchain: Scalable blockchain framework to support provenance in supply chains," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, IEEE, 2018, pp. 1–10.

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

[29] A. Kuzmanovic, "Net neutrality: Unexpected solution to blockchain scaling," *Communications of the ACM*, vol. 62, no. 5, pp. 50–55, 2019.

[30] U. Klarman, S. Basu, A. Kuzmanovic, and E. G. Sirer, "Bloxroute: A scalable trustless blockchain distribution network whitepaper," *IEEE Internet of Things Journal*, 2018.

[31] G. He, W. Su, and S. Gao, "Chameleon: A scalable and adaptive permissioned blockchain architecture," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, IEEE, 2018, pp. 87–93.

[32] J. Liu, W. Li, G. O. Karame, and N. Asokan, "Scalable byzantine consensus via hardware-assisted secret sharing," *IEEE Transactions on Computers*, vol. 68, no. 1, pp. 139–151, 2018.

[33] S. S. Hazari and Q. H. Mahmoud, "A parallel proof of work to improve transaction speed and scalability in blockchain systems," in *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)*, IEEE, 2019, pp. 0916–0921.

[34] Z. Gao, L. Xu, L. Chen, N. Shah, Y. Lu, and W. Shi, "Scalable blockchain based smart contract execution," in *2017 IEEE 23Rd international conference on parallel and distributed systems (ICPADS)*, IEEE, 2017, pp. 352–359.

[35] C. Ehmke, F. Wessling, and C. M. Friedrich, "Proof-of-property: A lightweight and scalable blockchain protocol," in *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, 2018, pp. 48–51.

[36] X. Fan and Q. Chai, "Roll-dpos: A randomized delegated proof of stake scheme for scalable blockchain-based internet of things systems," in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2018, pp. 482–484.

[37] G. Wang, Z. J. Shi, M. Nixon, and S. Han, "Smchain: A scalable blockchain protocol for secure metering systems in distributed industrial plants," in *Proceedings of the International Conference on Internet of Things Design and Implementation*, 2019, pp. 249–254.

[38] H. Gupta and D. Janakiram, "Colosseum: A scalable permissioned blockchain over structured network," in *Proceedings of the Third ACM Workshop on Blockchains, Cryptocurrencies and Contracts*, 2019, pp. 23–25.

[39] X. Min, Q. Li, L. Liu, and L. Cui, "A permissioned blockchain framework for supporting instant transaction and dynamic block size," in *2016 IEEE Trustcom/BigDataSE/ISPA*, IEEE, 2016, pp. 90–96.

[40] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "{Bitcoin-ng}: A scalable blockchain protocol," in *13th USENIX symposium on networked systems design and implementation (NSDI 16)*, 2016, pp. 45–59.

[41] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *25th usenix security symposium (usenix security 16)*, 2016, pp. 279–296.

[42] J. Teutsch and C. Reitwießner, "A scalable verification solution for blockchains," *arXiv preprint arXiv:1908.04756*, 2019.

[43] L. Chen, X. Zhang, and Z. Sun, "Scalable blockchain storage model based on dht and ipfs," *KSII Trans. Internet Inf. Syst*, vol. 16, pp. 2286–2304, 2022.

[44] Q. Zheng, Y. Li, P. Chen, and X. Dong, "An innovative ipfs-based storage model for blockchain," in *2018 IEEE/WIC/ACM international conference on web intelligence (WI)*, IEEE, 2018, pp. 704–708.

[45] T. Team, *Trifecta: The blockchain trilemma solved*, 2019.

[46] G. D. Monte, D. Pennino, and M. Pizzonia, "Scaling blockchains without giving up decentralization and security: A solution to the blockchain scalability trilemma," in *Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, 2020, pp. 71–76.

[47] M. Conti, A. Gangwal, and M. Todero, "Blockchain trilemma solver algorand has dilemma over undecidable messages," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019, pp. 1–8.

[48] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC press, 1997.

[49] E. Barker, L. Chen, A. Roginsky, and A. Vassilev, "Recommendation for key management, part 1: General (revision 4)," *NIST special publication*, vol. 800, no. 57, pp. 1–147, 2016.

[50] SECG, "Sec 2: Recommended elliptic curve domain parameters," Standards for Efficient Cryptography Group, Tech. Rep., 2000.

[51] D. J. Bernstein and T. Lange, "Safecurves: Choosing safe curves for elliptic-curve cryptography," 2014.

[52] S. Nakamoto. "Bitcoin: A peer-to-peer electronic cash system." (2008), [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[53] K. Zhou, C. Wang, X. Wang, S. Chen, and H. Cheng, "A novel scheme to improve the scalability of bitcoin combining ipfs with block compression," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 3694–3705, 2022.

[54] G. Diamantopoulos, R. Bahsoon, N. Tziritas, and G. Theodoropoulos, "Symbchain-sim: A novel simulation tool for dynamic and adaptive blockchain management and its trilemma tradeoff," in *Proceedings of the 2023 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, 2023, pp. 118–127.

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

[56] J. Lind, O. Naor, I. Eyal, F. Kelbert, E. G. Sirer, and P. Pietzuch, "Teechain: A secure payment network with asynchronous blockchain access," in *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, 2019, pp. 63–79.