

**Prototype of Blockchain Mining and Hashing with the
help of Secured Hashing Algorithm of SHA256 bits**

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Technology degree in
Information Technology

Submitted by

**BHUVANESHWARAN S (40120017)
DON INIGO S (40120031)**



DEPARTMENT OF INFORMATION TECHNOLOGYSCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY(DEEMED TO BE UNIVERSITY)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,CHENNAI,

TAMILNADU – 600119

April 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Bhuvaneshwaran S** (Reg.No. 40120017) and **Don Inigo S** (Reg.No. 40120031) who carried out the project as a "Prototype of Blockchain Mining and Hashing with the help of secured Hashing Algorithm of SHA 256 bits" under our supervision from January 2024 to April 2024.

Internal Guide

Dr.C. Geetha, M.Tech., Ph.D.,

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 17/04/2024

Internal Examiner

External Examiner

DECLARATION

I **Bhuvaneshwaran.S (Reg. No.40120017)** hereby declare that the Project Report entitled “**Prototype of Blockchain Mining and Hashing with the Help of Secured Hashing Algorithm of SHA 256 bits**” is done by me under the guidance of **Dr.C.GEETHA, M.Tech., Ph.D.**, and is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE:

PLACE: Chennai

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. R. Subhashini, M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. C. Geetha, M.Tech., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

Peer-to-peer computer process blockchain mining is used to secure and verify Bitcoin transactions. Mining involves blockchain miners who add Bitcoin transaction data to Bitcoin's global public Ledger of past transaction. In the ledger's blockchain are secured by blockchain miners and are connected to each other forming a chain. When we talk in depth, as opposed to traditional financial services systems, bitcoins have no central clearing house. Bitcoin transactions are generally verified in decentralized clearing systems wherein people contribute computing resources to verify the same. This process of verifying transaction is called as mining. A lot of computing power is consumed in the process of mining bitcoins the number of bitcoins that are generated from mining dwindles over time. In words of Satoshi Nakamoto there is only a limited supply of bitcoins only 21 million bitcoins will be ever created. As its core, the term "blockchain mining" is used to describe the process of adding transactions records to the Bitcoin blockchain. The process of adding blocks to the blockchain is how transactions are processed and how money moves securely around bitcoins. The process of blockchain mining is performed by a community of people around the world called "blockchain miners."

TABLE OF CONTENT

CHAPTER	TITLE	PAGE NO
1	INTRODUCTION	1
1.1	BLOCKCHAIN	1
1.2	HEURISTIC	2
1.3	SYSTEM OVERVIEW	4
1.4	SCOPE OF THE PROJECT	5
2	LITERATURE SURVEY	7
3	SYSTEM ANALYSIS	18
3.1	EXISTING SYSTEM	18
3.2	PROPOSED SYSTEM	20
3.3	REQUIREMENTS SPECIFICATIONS	22
3.3.1	Hardware requirements	22
3.3.2	Software requirements	22
3.3.3	Language Specification	22
4	SYSTEM DESIGN	24
4.1	SYSTEM ARCHITECTURE	24
5	MODULE DESCRIPTION	27
5.1	PROJECT MODULES	27
5.2	MODULE DESCRIPTION	27
5.2.1	Block	27
5.2.2	Block hashing	29
5.2.3	Blockchain Architecture	31
	CONCLUSION	32

FUTURE WORKS	34
APPENDIX 1	35
APPENDIX 2	51
RESEARCH PAPER	59

ABBREVIATIONS

ABBREVIATION	DEFINITION
ADB	Actual Data Block
AES	Advanced Encryption Standard
BC	Block Content
CBC	Cypher Block Chaining
DEA	Data Encryption Algorithm
SHA	Secured Hash Algorithm
PKI	Public Key Infrastructure
NMK	New Master Key
ICB	Initial Chaining Value
KEK	Key-Encryption Key

CHAPTER 1

INTRODUCTION

1.1 BLOCKCHAIN

Blockchain is the backbone Technology of Digital Cryptocurrency Bitcoin. The blockchain is a distributed database of records of all transactions or digital events that have been executed and shared among participating parties. Each transaction is verified by the majority of participants of the system. It contains every single record of each transaction. In order to understand how blockchain works, it's very important to understand in detail the concept of Hashing.

There are three main parts of any blockchain network:

- **Node/ Block**: It is the main building block of any blockchain. It acts as a database to store the information related to all transactions. The size, period, and triggering event for blocks are different for every blockchain. Each block or node contains a complete record of all the transactions that were ever recorded in that blockchain.
- **Network**: The network is composed of “full nodes”. Think of them as the computer running an algorithm that is securing the network.
- **Hash**: It acts as a chain that links one block to another, mathematically one can say that it's “chaining” all blocks together. This is one of the most difficult concepts in blockchain to comprehend. It's like magic that glues blockchains together and allows them to create mathematical trust and maintain privacy as well as security in the network. The hash in the blockchain is created from the data that was in the previous block. So we can say that the hash is a fingerprint of this data and locks blocks in order and time.

Blockchain did not initially boom when it was discovered. People thought blockchain was a new form of money laundering when it was first discovered.

But the benefits of blockchain was later understood by many people when the riches started converting their physical money into bitcoins.

1.2 **HEURISITCS**

Here are some heuristics for blockchain hashing:

1. **SHA-256 Algorithm**: SHA-256 (Secure Hash Algorithm 256-bit) is widely used for hashing in blockchain. It is a cryptographic hash function that takes an input and produces a fixed-size 256-bit (32-byte) hash value.
2. **Mining Difficulty**: The hash function used in blockchain is designed to be difficult to compute. Mining difficulty is adjusted to ensure that the time taken to compute a new block is consistent with the desired block time.
3. **Randomness**: A good hash function should be able to produce a unique output for each input. In blockchain, randomness is achieved by including a nonce in the block header.
4. **Hash Rate**: Hash rate is the speed at which a miner can compute a hash function. It is measured in hashes per second (H/s) or more commonly in terahashes per second (TH/s).
5. **Merkle Tree**: A Merkle tree is a data structure used in blockchain that allows for efficient verification of the integrity of large amounts of data. It works by hashing pairs of transactions together, then hashing those results together, and so on, until a single hash is produced.
6. **Proof of Work**: Proof of Work (PoW) is a consensus algorithm used in blockchain that requires miners to perform a computationally expensive calculation in order to add a new block to the chain. This helps to ensure that the network is secure and resistant to attacks.
7. **Hash Function Collision Resistance**: A good hash function should be collision-resistant, meaning that it should be difficult to find two inputs that

produce the same output. This helps to ensure the integrity of the blockchain by preventing fraudulent transactions.

8. **Hashing Efficiency**: Hashing efficiency refers to the speed and resource utilization of the hashing algorithm. In order to ensure scalability of the blockchain network, it is important to use a hashing algorithm that is both efficient and secure.
9. **Security**: The security of the blockchain network depends heavily on the security of the hashing algorithm used. The hashing algorithm should be resistant to attacks such as preimage attacks, collision attacks, and birthday attacks.
10. **Hashing Algorithm Diversity**: It is important for the blockchain network to have diversity in the hashing algorithms used to prevent a single point of failure. This also helps to ensure that the network is resistant to attacks.
11. **Hash Function Verification**: The integrity of the blockchain network depends on the correctness of the hash functions used. It is important to verify the correctness of the hash functions used in the network by using cryptographic proofs such as hash function verification.
12. **Hash Function Migration**: Hash functions used in blockchain can become vulnerable over time as new attacks are discovered. It is important to have a mechanism for migrating to a new hashing algorithm when the existing algorithm becomes vulnerable.
13. **Hashing Algorithm Consensus**: In order to ensure the security and integrity of the blockchain network, it is important to have a consensus mechanism for selecting the hashing algorithm used. This consensus mechanism should involve all stakeholders in the network to ensure that the selected algorithm is secure and efficient.
14. **Hash Function Standardization**: Standardization of hash functions used in blockchain is important to ensure interoperability and compatibility.

1.3 **SYSTEM OVERVIEW**

Blockchain hashing is a critical component of the blockchain technology. It is the process of generating a fixed-length output, or hash, from an input data using a cryptographic hashing function. The hash function is designed to produce a unique output for each input and should be computationally expensive to compute, making it difficult for attackers to modify the data in the blockchain.

Here is an overview of the system architecture of blockchain hashing:

1. **Input data**: The input data to the blockchain hashing system is the data that needs to be stored in the blockchain, such as transaction data.
2. **Hash function**: The hash function is the core component of the blockchain hashing system. It is a cryptographic algorithm that takes an input data and generates a fixed-length output, or hash.
3. **Hashing process**: The hashing process involves applying the hash function to the input data to generate the hash. The hash function is designed to be computationally expensive, requiring significant computational resources to compute the hash. This makes it difficult for attackers to modify the data in the blockchain.
4. **Hashed data storage**: The hashed data is then stored in the blockchain. The hash is stored along with other transaction information in a block, and the block is added to the blockchain using a consensus algorithm such as proof of work or proof of stake.
5. **Verification**: In order to verify the integrity of the blockchain, the hashes of each block are compared to the previous block's hash. If the hashes match, it is a strong indication that the data has not been tampered with.
6. **Mining**: Mining is the process of adding new blocks to the blockchain. Miners use specialized hardware to perform the computationally

expensive hashing process required to add a new block to the blockchain.

7. **Hashing algorithm selection**: The selection of the hashing algorithm used in the blockchain is critical to its security and scalability. Different hashing algorithms have different properties, such as computation speed, security, and resistance to attacks.

Overall, blockchain hashing is a critical component of the blockchain technology, providing a secure and efficient way to store and verify data on the blockchain.

1.4 **SCOPE OF THE PROJECT**

The scope of a project for blockchain hashing can vary depending on the specific goals and objectives of the project. However, here are some general aspects that may be included in the scope of such a project:

1. **Requirements gathering**: The project scope should include a thorough analysis of the requirements for the blockchain hashing system. This may involve consultation with stakeholders to identify the data to be stored on the blockchain and the hashing algorithms to be used.
2. **Hashing algorithm selection**: The project may involve selecting an appropriate hashing algorithm for the blockchain based on factors such as security, scalability, and efficiency.
3. **System architecture design**: The scope of the project may include designing the system architecture for the blockchain hashing system, including the components of the system, the interactions between the components, and the data flow within the system.
4. **Implementation**: The project may involve developing the software components for the blockchain hashing system, including the hash function and the blockchain database.

5. **Testing**: The project should include testing the blockchain hashing system to ensure that it meets the requirements and specifications. This may include unit testing, integration testing, and acceptance testing.
6. **Deployment and maintenance**: Once the system is tested and approved, the project scope may include deploying the blockchain hashing system and providing ongoing maintenance and support for the system.
7. **Security considerations**: The project should include an assessment of the security risks associated with the blockchain hashing system and the implementation of appropriate security measures to mitigate those risks.
8. **Performance optimization**: The project may involve optimizing the performance of the blockchain hashing system, such as reducing the time required to generate a hash or minimizing the amount of computational resources required.
9. **Integration with blockchain network**: The project may involve integrating the blockchain hashing system with the larger blockchain network, including ensuring compatibility with existing blockchain software and protocols.
10. **Scalability considerations**: The project should include consideration of the scalability of the blockchain hashing system, such as the ability to handle large amounts of data or high transaction volumes.
11. **Regulatory compliance**: Depending on the specific use case of the blockchain hashing system, the project may need to include compliance with regulatory requirements, such as data privacy laws or financial regulations.
12. **Interoperability**: The project may involve designing the blockchain hashing system to be interoperable with other blockchain networks or systems, allowing for seamless data transfer and communication.
13. **User interface design**: If the blockchain hashing system is intended to be used by end-users, the project may include designing a user interface that is intuitive and easy to use.

CHAPTER 2

LITERATURE REVIEW

Jinhua Fu, Sihai Qiao, Yongzhong Huang, Xueming Si, Bin Li,¹ and Chao Yuan, “A Study On The Optimization Of Blockchain Hashing Algorithm Based On PRCA”, 2020

Blockchain is a kind of distributed general ledger technology, originated from the literature. Initially, it was mainly used in the field of cryptocurrency, the most representative of which were Bitcoin and Litecoin, Monroe, and Zcash. Amid its rapid development, blockchain technology can effectively guarantee the authenticity, security, and reliability of data. It also has been widely used in medical data, personal data protection, and data allocation scheme. As the basic unit of blockchain, block consists of partition header including original data and block body including transaction data. Among them, block data are used to connect the previous block and index the data from the hash value of range block. Each blockchain transaction is conducted by using hash function interaction. It guarantees the security of blockchain.

However, with the continuous development of blockchain, its security issues become increasingly prominent. The lightweight hash function SHA1 in the blockchain is no longer regarded as an attacker that can withstand sufficient funds and computing resources. SHA256 can replace SHA1 for information exchange with good anticollision ability, while it cannot be changed at will. To avoid chain breakage, it is necessary to modify the hash values of all blocks behind the block at the same time. As a result, a large computational complexity is needed and the security of the blockchain is not guaranteed.

In the process of executing operations, the PRCA (Proactive Reconfigurable Computing Architecture) generates the optimal computation structure set by self-perception and dynamic selection. All the software and hardware variants are dynamically variable. Therefore, in the process of application processing, they can select optimal solutions according to the independent variables in the program to get the variable optimal solution sets with equivalent function and different computing efficiency. Combining with blockchain, it can improve the performance of the algorithm, improve the transmission efficiency, and enhance the security of hash algorithm.

PRCA is an operation mechanism based on multidimensional reconstructed functional structure and dynamic multibody. When proactive reconfigurable

computation is processing data, execution structures, such as computing, storage, and interconnection, are changing dynamically with the efficiency of transaction processing, instead of improving the algorithm to improve the operation performance without changing the basic hardware. There are many functional equivalents in PRCA, but they are accomplished by combining different hardware structures with this algorithm. The purpose is to achieve the high performance of computing, that is, how to automatically perceive variables to generate the optimal computing set and autonomously reconstruct the computing in the processing algorithm.

PRCA has variable infrastructure and algorithm, which makes it possible to obtain optimal solutions to different problems. It pursues different services and comprehensive high performance under different loads or other conditions, builds the most appropriate processing components, and forms the most appropriate architecture. Proactive reconfigurable computation combines the advantages of general computing and special computing to achieve the goal of solving problems efficiently. In terms of the general computing structure, it is characterized by its determined structures and variable algorithm and may calculate any computable problems with high efficiency.

John p. Conley, “Encryption, Hashing, PPK and Blockchain : A Simple Introduction”, 2019

Hashing is a method used to verify the integrity of a message or file. The hash function itself is public knowledge and does not need any kind of key to work. Through a hash function it produces a fixed-length output string. The Secure Hash Algorithm-256 (SHA-256) is one widely used approach and has the following properties: Running any file, regardless of length, through SHA-256 returns a 256-bit binary string. Very similar files produce quite different hashes in an unpredictable way. In fact, SHA-256 is designed so that hashes of different files are, in effect, randomly and uniformly distributed⁴ over the set of all possible 256 binary strings of which there are 2^{256} or about 10^{77} . The hash of any file is unique. The same input always produces the same output. For this reason, the hash is sometimes referred to as a file’s “fingerprint”. Although hashing a file always gives the same result, two files may have the same hash. This is called a “collision”. In practice, however, this extremely unlikely to occur. Hash functions are noninvertible. It is impossible to recover a file from its hash. Applications of Hash Functions Verifying Documents Suppose you sign a partnership agreement and later have a dispute. You and your partner both present copies of the contract to a judge, but they say different things. How can the judge determine which is

genuine? Suppose the judge had access to a hash of the contract as it was on the day it was signed. He could then compare it to hashes of the contracts presented to him by you and your partner. The genuine contract will produce the same hash, and any altered contract will produce something else. You may ask how the judge might have access to a hash of the contract he knows was taken on the day it was signed. Read on about digital signatures and blockchain below.

Verifying Logins

When a user tries to log into a system, the system compares the password the user provides to the one stored in its files. The problem with this is that the passwords stored by the system might be stolen by hackers. Thus, it is good security practice to store only hashes of user credentials. Users still type in passwords, but the system immediately takes a hash and compares it to the hashed password stored in its files. If the password is correct, the hashed password will match a stored hash. If the hashed password file is stolen from the system, it is of no use. The password cannot be determined from the hash, and the system requires a correct plaintext password to grant access, not the hash itself.

Securely Resetting Passwords

Sometimes when you forget a password, you can request that it be sent from the website to your verified email address. This is not the most secure practice since email often transits the Internet in 4 Okay, not randomly; hash functions are deterministic. However, if you took a file, modified it one bit at a time, and then looked at the distribution of the resulting hashes, they would be approximately uniformly distributed between 0. ..0 and 1. .. 1, where these are each 256 bit long binary strings. 5 Collisions must occur in theory. A typical MP3 file is tens of millions of bits long. Obviously, it is impossible to do a one-to-one mapping between the set of every possible 10M bit sting and every possible 256 bit string. However, If I took a hash of any file and then started looking for another file with the same hash, I could test $10^{77/2}$ files and still have only a 50% chance of finding one. Thus, it is extremely unlikely that you will ever actually see a collision. 4 August 17, 2019 John P. Conley plaintext and can be stored on any intermediate server that passes the message along. However, if a website follows the practice of only storing hashes of passwords, it literally cannot send a user his password since it does not know it itself. This is why you sometimes get a message asking you to click on a one-time link that quickly expires and and choose a new password. This password is immediately hashed and stored. This approach relies on three things to make it secure: that the email address has previously been validated, the use of two part authentication in many cases, and the fact that a hacker would have to intercept and use the link immediately while having access to your email account.

Privacy Preserving Data Matching

Suppose we all joined a social network and wanted to know if any of the people in our

address books had also joined. None of us, however, wanted to reveal our own email address or the contents of our address books to the network. We could instead submit a hashed version of our address books. The network would be able to see that one person's email address was in another person's address book by finding identical hashes. However, the network would not be able to read the email addresses themselves and so user privacy would be preserved.

Efficient Database Search Suppose we had a database that contained names, email addresses, email message texts, documents, audio recording, case histories, and other large and dissimilar items. Suppose we wanted to find the record that contained a specific document or email message. Running a standard search looking for an identical document in the database would be computationally intensive since it would require comparing data elements that are many kilobytes or megabytes long against the document we wanted to find. We could instead create a database that contained hashed versions of all the data elements and search for a match with a hash of the document we were looking for. This would require comparing data elements that are only 256 bits long. The bigger the average size of the original data elements, the greater the computational savings from searching for matching hashes.

Dipankar Dasgupta, John M. Shrein & Kishor Datta Gupta , “A Survey Of Blockchain With Security Perspective”, 2019

The report starts with an overview of the blockchain security system and then highlights the specific security threats and summarizes them. We review with some comments and possible research direction. This survey, we examine the security issues of blockchain model related technologies and their applications. The blockchain is considered a still growing like the internet in 1990. It has the potential to disrupt so many technology areas in the future. But as a new underdeveloped field, it is suffering many setbacks mostly resulting from the security area. Its security concerns coming not only from distributed/decentralized computing issue or Cryptography algorithm issue, from some unexpected field too. Here, in this paper, we tried to classify the security concerns for the blockchain based on our survey from recent research papers. We also tried to show which way blockchain development trends are going.

David Derler, Kai Samelin, Daniel Slamanig, and Christoph Striecks , “A Fine-Grained And Controlled Rewriting In Blockchains : Chameleon-Hashing Gone Attribute-Based”, 2019

Blockchain technologies recently received a considerable amount of attention. While the initial focus was mainly on the use of blockchains in the context of cryptocurrencies such as Bitcoin, application scenarios now go far beyond this. Most blockchains have the property that once some object, e.g., a block or a transaction, has been registered to be included into the blockchain, it is persisted and there are no means to modify it again. While this is an essential feature of most blockchain scenarios, it is still often desirable - at times it may be even legally required - to allow for breaking this immutability in a controlled way. Only recently, Ateniese et al. (EuroS&P 2017) proposed an elegant solution to this problem on the block level. Thereby, the authors replace standard hash functions with so-called chameleon-hashes (Krawczyk and Rabin, NDSS 2000). While their work seems to offer a suitable solution to the problem of controlled re-writing of blockchains, their approach is too coarse-grained in that it only offers an all-or-nothing solution. We revisit this idea and introduce the novel concept of policy-based chameleonhashes (PCH). PCHs generalize the notion of chameleon-hashes by giving the party computing a hash the ability to associate access policies to the generated hashes. Anyone who possesses enough privileges to satisfy the policy can then find arbitrary collisions for a given hash. We then apply this concept to transaction-level rewriting within blockchains, and thus support fine-grained and controlled modifiability of blockchain objects. Besides modeling PCHs, we present a generic construction of PCHs (using a strengthened version of chameleon-hashes with ephemeral trapdoors which we also introduce), rigorously prove its security, and instantiate it with efficient building blocks. We report first implementation results.

Sarwar Sayeed and Hector Marco-Gisbert, “Assessing Blockchain Consensus and Security Mechanisms against the 51% Attack”, 2019

To understand the significant contributions of this paper, it is necessary to elaborate on the consensus mechanism and blockchain technology in depth. This section enhances the quality of experience (QoE) of the blockchain network as well as discusses the mining and hashing context.

1. Blockchain Technology: The Digital Concept

Blockchain technology consists of records of data which are referred to as a ledger. It employs a distributed system for the verification of records. Blocks are verified by thousands of network participants around the globe to keep the network active. Here we discuss the full concept behind blockchain technology.

2. Consensus Mechanism

A consensus mechanism is a protocol which is in place to ensure that all the participants in the blockchain network are complying with the agreed rules [4]. It ensures that the transactions emerge from a legitimate source by having every participant consent to the state of the distributed ledger. The public blockchain is a decentralized technology, and no centralized authority is in place to regulate the required act. Therefore, the network requires authorizations from the network participants for the verification and authentication of any activities that occur in the blockchain network. The whole process is done based on the consensus of the network participants, and it makes the blockchain a trustless, secure, and reliable technology for digital transactions. Distinct consensus mechanisms follow different principles, which enables the network participants to comply with those rules. Several consensus mechanisms have been introduced considering the requirements of secure digital transactions. However, proof of work (PoW), proof of stake (PoS), and delegated proof of stake (DPoS) are the few consensus protocols used by the main cryptocurrencies.

3. Blockchain Functionality

Blockchain is a digital technology which stores digital information in blocks. Each block is chained to the blocks adjacent to it, called a blockchain. Any information inside the blocks can never be changed due to the immutable feature. Altering a single piece of information of a block invalidates every block after the one that was altered. The block information is available publicly, and one can easily track almost all key information, such as client records, property contracts, and payment information. Every block comprises a digital signature which correlates with the strings in that particular block. The whole process occurs through a hashing procedure, and a change to a single digit of the block string produces an entirely new digital signature. The blockchain blocks are available for verification by anyone who is part of the network. As a result, any malicious alterations can be immediately detected by the network participants. Regardless of the failure of any part of the network, the blockchain network still keeps

running as normal. A block in the bitcoin blockchain typically has a size of 1 MB, and it can contain many transactions as long as it is within the size limit.

4. Hashing and Digital Signatures

A digital signature is a mechanism that is used to validate the accuracy and integrity of a data string [6]. Every block comprises one unique digital signature based on the strings that the block contains. Figure 2 presents a partial view of 3 blocks that are chained to each other. Each block produces a digital signature which is generated considering the contents, such as Merkle hash and the digital signature of previous blocks. A digital signature does not always qualify to validate the next block, as the bitcoin blockchain sets a level of difficulty to determine the block creation time. Hence, having fewer transactions in a block does not necessarily help to create a block faster than the other nodes. If the difficulty level requires having 17 zeros in front of the digital signature, then the miners require meeting that condition to validate a block. An attacker making modifications to the data string changes the digital signature and will fail to meet the requirement for the next blocks that are chained to it [7]. To overcome the scenario, a new signature must be generated for every subsequent block on the chain. The process becomes very costly and nearly impossible, as the attacker is required to generate new signatures not only for the blocks which have been corrupted but also any blocks that are being added on a regular interval.

5. Blockchain Mining

Blockchain mining is a process which involves computing a new block to be added to the distributed ledger [8]. The mining is mainly correlated to bitcoin, as the block generation process differs on distinct cryptocurrencies. Mining computers are often powerful machines that are used in generating the correct hash values. Nodes that are involved with mining are required to solve a mathematical puzzle utilizing their machines to mine. As soon as a miner successfully produces a new block, the miner notifies the network about it. Every miner in the network stops mining that block and again restarts solving equations for the next block. In the mining process, a miner takes the hash Merkle root and adds it to the nonce value. A new hash is generated from the alteration of the nonce and then compared with the target each time. If the hash value is less than the target, the mathematical puzzle is solved. However, while the hash value is higher than the target, the nonce value needs to be changed and matched with the target value until the conditions are met. A mining pool involves hundreds of

thousands of mining machines. Therefore, a pool usually comprises more hashing power than other miners and has the advantage to find the required hash faster than other miners [9]. The mining time of bitcoin is about 10 min, requiring extensive hardware for the process. Currently, miners receive a mining reward of 12.5 BTC for every successful block they generate and also the fees associated with each transaction. The reward reduces to half about every four years.

In this section, we presented the full overview of the blockchain technology. This technology is at a high risk and has been exploited by various attacks in recent time. Therefore, in the next section, we present various security threats to reveal their attacking strategies on the blockchain network.

Tsung-Ting Kuo, Hyeon-Eui Kim, Lucila Ohno-Machado, “Blockchain Distributed Ledger Technologies for Biomedical And Healthcare Applications”, 2017

To solve the double-spending problem, each computation node in the blockchain network not only needs to store every transaction to enable the distributed verification of the transactions, but also to follow a distributed timestamp mechanism to determine which transactions should be accepted and which should be rejected. The Bitcoin blockchain exploits hash-chain¹⁴ as a distributed timestamp mechanism, and every node maintains a copy of the chain to store every transaction.

An example of simplified blockchain (hash-chain). Each transaction of coins is enclosed in a *block*. A block may contain multiple transactions and is a basic unit to be verified. Each block also contains a hash value of the previous block's header, and thus forms a *hash-chain* or *blockchain*. As all blocks are chained, the order of the blocks is deterministic; therefore, each block can serve as a timestamp of the enclosed transactions to solve the double-spending problem. Note that each node maintains a copy of the whole blockchain, thus every node can verify every transaction. For example, suppose the transaction from Alice to Charlie is enclosed in block *B1* and the one from Alice to Bob is enclosed in block *B2*; everyone in the network can verify that *B1* happened before *B2* by checking the hashed blockchain, and thus the double-spend transaction from Alice to Bob should be rejected.

Additionally, the “mining” process (ie, creating a block with enclosed transactions) should be relatively difficult to do, but relatively easy to check.¹⁵ The process should be difficult (ie, time-consuming and costly) in order to make attempts to create invalid blocks prohibitively expensive (at the cost of increasing the time available to create valid blocks). To implement such a design, Bitcoin blockchain adopts a proof-of-work protocol¹ that each block creator has to follow. Figure 4 illustrates how a chain of blocks is created with this protocol.

An additional benefit of the proof-of-work consensus protocol used in blockchain is the ability to resolve disagreement of the chains, and thus let blockchains be immutable audit trails.^{1,15} That is, when an attacker modifies a block, all the blocks after that block are recomputed, because each block contains the hash value of the previous block’s header,¹⁶ and the computational cost of such modification should be high enough to prohibit attacks (Figure 4).

On the other hand, when an attacker creates a malicious chain to compete with an honest chain and tries to replace the honest one, the proof-of-work majority voting mechanism can also significantly reduce the probability of such an attack to succeed (Figure 5). Detailed analyses of the attack-resisting ability of the blockchain proof-of-work consensus protocol are proposed in several recent studies.^{1,18–22} The Bitcoin mechanism also provides rewards to the nodes, as an incentive to compensate the high cost associated with “mining” a new block and verifying transactions.

Kaimeng ding, Shiping Cheng, Jiming Yu, Yanan Liu and Jie Zhu “Hasing Algorithm Based On MultiRes-RCF for Blockchains of HRRS images”, 2022

A blockchain [13] is a chain structure with data blocks as the basic unit. Blockchains were initially used in the field of digital currency and later expanded to many industries such as the Internet of Things [14,15], supply chain management [16,17], medical care [18], education [19] and e-commerce [20]. Essentially, a blockchain is a decentralized database: instead of just one master database storing the blockchain’s data, multiple copies of the same blockchain data are distributed across a peer-to-peer network. In a blockchain, data are stored on the chain in the form of blocks. Each block is connected in chronological order

and is mainly composed of a block header containing metadata and a block body containing the specific block data stored in this block [21]. The implementation process of a blockchain integrates various technologies such as cryptography and distributed systems to build a distributed environment, in which transactions can be performed safely without mutual trust and solves the problems of data storage security that may be caused by central institutions. With the in-depth development of blockchain technology, it has also been initially developed and applied in remote sensing [7,8]. The immutability guarantees the integrity of the on-chain HRRS image to a large extent.

However, there are still many deficiencies in the authentication technology of a HRRS images' blockchain. The outstanding problem is that the blockchain is too sensitive to changes in the content of HRRS images. For example, after the format conversion of the HRRS image on the chain, although the image has changed a lot at the binary level, its effective content has not changed significantly. In fact, the users of HRRS images are more concerned about whether the effective content of the HRRS images has been tampered with, rather than whether the carrier of the content has changed. The root of the above problem is that the blockchain uses a cryptographic hash function (SHA-1 or SHA-256) to generate the digital digest of the HRRS image, which can be solved by subject-sensitive hashing.

2.2. Subject-Sensitive Hashing

Subject-sensitive hashing [9], also known as the subject-sensitive hash algorithm, is a new hashing technology based on perceptual hashing. Subject-sensitive hashing focuses the goal of integrity authentication on the types of features that users are interested in. It overcomes the insufficiency of perceptual hashing and can achieve “customized” integrity authentication according to the actual needs of the users. A subject-sensitive hashing algorithm generally includes the following steps: subject-sensitive feature extraction, feature quantization, and feature encoding. In the above steps, subject-sensitive feature extraction focuses on extracting the type of information that users pay attention to, taking into account other types of feature information; feature quantization is conducted to remove the redundancy of the extracted features; and feature encoding is performed to compress and encrypt the quantized features to obtain a subject-sensitive hash sequence. Among them, feature extraction is the key step.

The existing subject-sensitive hashing algorithms for HRRS images are mainly implemented based on deep learning. In [9], a deep neural network model MUM-Net for subject-sensitive hashing is proposed. The subject-sensitive hash algorithm based on MUM-Net has good robustness and tamper sensitivity, but the model has high computational complexity. In [22], a lightweight deep neural network Semi-U-net is proposed, which overcomes the shortcomings of MUM-Net's low computational efficiency and overly complex model. In [23], a deep neural network model AAU-Net based on the Attention mechanism was proposed, which significantly enhanced the distinction between "subject-related tampering" and "subject-unrelated tampering".

Different subject-sensitive hashing algorithms often have significant differences in specific properties. For example, the Attention ResU-Net-based subject-sensitive hashing algorithm has good robustness, but its tamper sensitivity is not good, making it unsuitable for HRRS image integrity authentication [23]. This means that the specific subject-sensitive hashing algorithm implementation process should be determined according to the application requirements of the algorithm.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

System analysis is a process of studying and evaluating a system to understand how it works, identify areas for improvement, and make recommendations for changes. In the case of blockchain hashing, system analysis involves studying the various components of a blockchain system and how they interact to ensure the integrity and security of the data stored on the blockchain.

Existing systems that use blockchain hashing include popular cryptocurrencies like Bitcoin and Ethereum. These systems use a variety of hashing algorithms to secure the data stored on the blockchain and to validate transactions. Bitcoin, for example, uses the SHA-256 hashing algorithm, while Ethereum uses the Ethash algorithm.

To analyze the effectiveness of these existing systems, several factors should be considered, including security, scalability, and efficiency. Security is a critical concern for blockchain systems, as any vulnerabilities in the hashing algorithm or other components of the system could lead to loss or manipulation of data. Therefore, the security of the hashing algorithm and other components should be evaluated and any vulnerabilities identified.

Scalability is another important consideration in the analysis of blockchain hashing systems, as these systems must be able to handle large amounts of data and high transaction volumes. The efficiency of the hashing algorithm and other components should also be evaluated to ensure that the system can operate quickly and with minimal energy consumption.

Enhancing security: As blockchain technology continues to evolve, new security threats may emerge, requiring constant updates to the hashing algorithm and other security measures. System analysis can help identify potential vulnerabilities in the system and recommend changes to mitigate these risks.

Increasing scalability: As blockchain systems become more widely adopted, the volume of data and transactions on the network will increase. System analysis can help identify potential scalability issues and recommend changes to improve the system's ability to handle larger volumes of data and transactions.

Improving efficiency: The energy consumption required to validate transactions on a blockchain network can be significant. System analysis can help identify areas where energy consumption can be reduced, such as optimizing the hashing algorithm or implementing more efficient consensus mechanisms.

Interoperability: Currently, different blockchain networks may use different hashing algorithms, which can make it difficult to transfer data between different networks. System analysis can help identify ways to promote interoperability between different networks, such as developing standardized hashing algorithms.

Privacy and regulatory compliance: As blockchain technology becomes more widely adopted, there will be increasing regulatory requirements around data privacy and security. System analysis can help identify potential compliance issues and recommend changes to ensure that the system is compliant with relevant regulations.

Disadvantages of Existing System:

Potential for collisions: As mentioned earlier, collisions can occur where different data inputs generate the same hash output. This can compromise the integrity of the blockchain and lead to security vulnerabilities.

Computational complexity: The time and resources required to compute and verify hashes can become significant as more data is added to the blockchain, leading to slower transaction processing times and higher energy consumption.

3.2 PROPOSED SYSTEM

Improved hashing algorithms: New and more efficient hashing algorithms such as SHA-3, Blake2, and Argon2 can provide stronger security and faster processing times. These algorithms are designed to resist brute-force attacks and provide better resistance to collision attacks. For instance, SHA-3 is based on the sponge construction, which makes it more resistant to collision attacks compared to SHA-2.

Hybrid consensus mechanisms: Hybrid consensus mechanisms that combine Proof-of-Work (PoW) and Proof-of-Stake (PoS) algorithms can improve security and reduce energy consumption. PoS algorithms are less energy-intensive than PoW, but they may be less secure in certain scenarios.

Segregated Witness (SegWit): SegWit can improve the efficiency of hashing by separating transaction data from signature data, reducing the size of each transaction. This can increase the transaction throughput and reduce the transaction fees. For instance, SegWit is implemented in the Bitcoin network to reduce the transaction fees and improve the transaction capacity.

Sharding: Sharding can improve the scalability of blockchain hashing by dividing the blockchain into smaller, more manageable pieces called shards. Each shard can be processed separately, reducing the time and resources required for verification. For instance, Ethereum 2.0 uses sharding to improve the scalability of the network.

Quantum-resistant hashing: Quantum-resistant hashing algorithms can protect against potential attacks on the blockchain by quantum computers. These algorithms are designed to resist attacks based on the principles of quantum computing and provide better long-term security. For instance, BLAKE3 is a quantum-resistant hashing algorithm that is designed to resist attacks from quantum computers.

Advantages:

Improved security: The use of stronger and more efficient hashing algorithms, hybrid consensus mechanisms, and quantum-resistant hashing can provide better security and protect against potential attacks on the blockchain.

Faster transaction processing: The use of SegWit and sharding can improve the efficiency of hashing and increase the transaction throughput, resulting in faster transaction processing times.

Improved scalability: The use of sharding can improve the scalability of the blockchain network, enabling it to handle a larger number of transactions.

Lower transaction fees: The use of SegWit can reduce the size of each transaction, resulting in lower transaction fees for users.

Reduced energy consumption: The use of hybrid consensus mechanisms and sharding can reduce the energy consumption of the blockchain network, making it more environmentally friendly.

3.3 REQUIREMENT SPECIFICATION

3.3.1 Hardware Requirements:

The hardware requirements for the proposed system are as follows:

Processor: The system requires a modern processor capable of supporting the proposed hashing algorithms, such as SHA-3, Blake2, and Argon2.

Memory: The system requires sufficient RAM to handle the complex mathematical computations involved in hashing.

Storage: The system requires high-speed storage devices, such as solid-state drives (SSDs), to ensure fast data access and processing.

Network: The system requires a reliable and high-speed network connection to ensure fast and secure data transmission between nodes.

3.3.2 Software Requirements:

The software requirements for the proposed system are as follows:

Operating System: The system can run on any modern operating system that supports the proposed hashing algorithms and network protocols, such as Windows, Linux, and macOS.

Blockchain Framework: The system requires a blockchain framework that can support the proposed improvements, such as hybrid consensus mechanisms, SegWit, sharding, and quantum-resistant hashing.

Database Management System: The system requires a database management system (DBMS) to store and manage the blockchain data.

Development Tools: The system requires development tools such as code editors, compilers, and debuggers to develop and test the system.

3.3.3 Language Specification:

The proposed system can be implemented using various programming languages, such as:

C++: C++ is a popular language for blockchain development and can support the proposed improvements.

Rust: Rust is a language that provides memory safety and thread safety, making it suitable for building secure blockchain systems.

Python: Python is a high-level language that is easy to learn and can support the proposed improvements.

Solidity: Solidity is a language specifically designed for building smart contracts on the Ethereum blockchain.

React JSON: React JSON is a JavaScript library that can be used to visualize and interact with JSON data structures in a web application. It can be used to build user interfaces for blockchain applications that involve JSON data.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

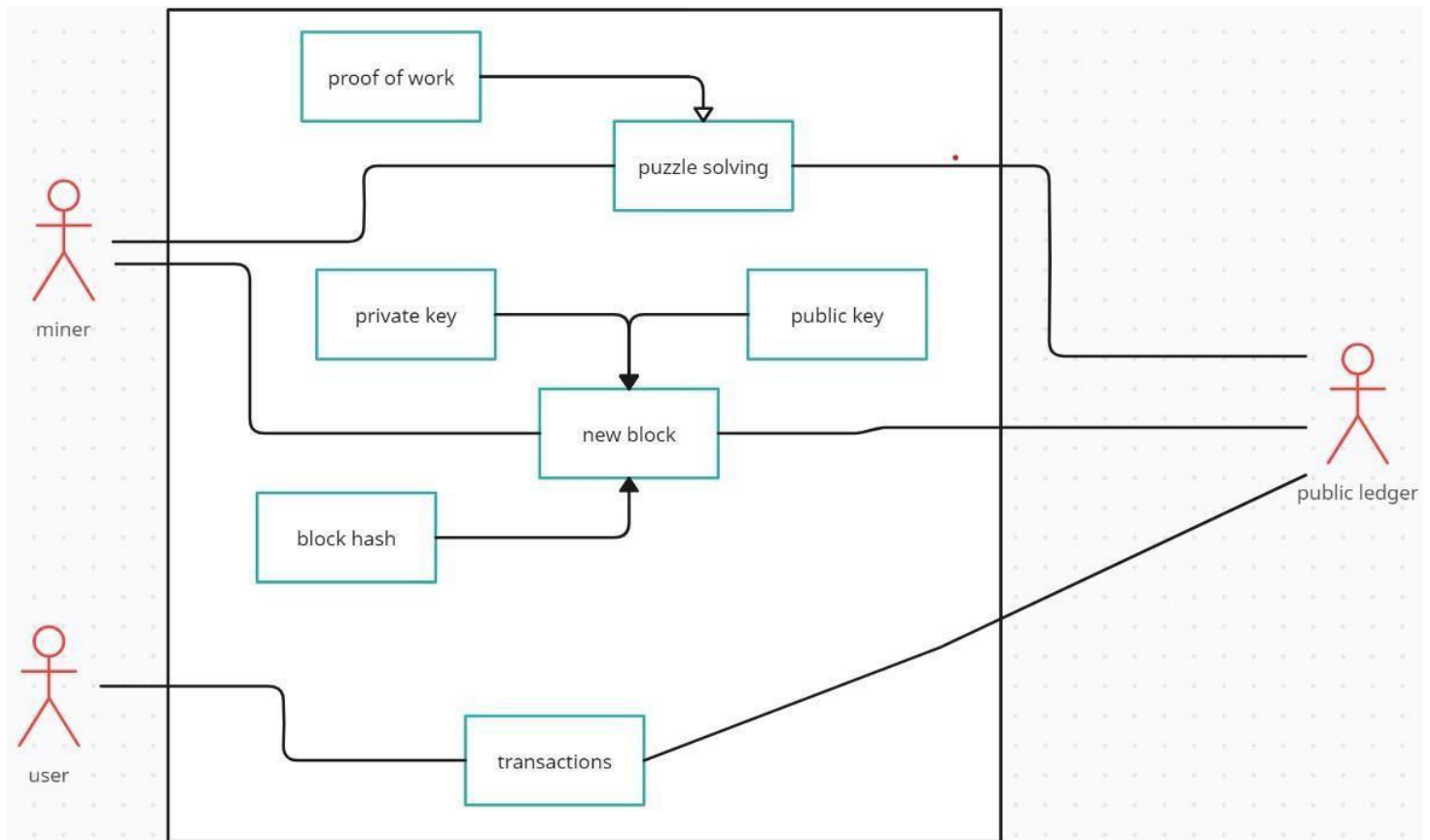


Figure 4.1 Architecture of the proposed system

In the figure 4.1, we can see that there are three users coming into the picture during the process of block hashing. The miner, the user, and the public ledger. The block hashing starts when the miner begins to solve a given puzzle in order to unlock a new block. Many miners take part in solving the puzzle in order to claim ownership of the new block mined. The reason behind the participation of many miners is that, they gain a profit from the amount that was transacted and stored in that block.

To verify that a particular miner has solved a puzzle to mine a new block is done through a document called Proof Of Work (Pow). This document is obtained by the miner after he successfully solves a puzzle. The difficulty rate of

the puzzle varies from block to block. And also keep in mind that, puzzle solving not only needs a good knowledge but also powerful computers so not everyone can become miners.

After mining a new block, the new block now contains the following data and information – private key, public key, block hash, timestamp, merkle root, previous block hash and also a space to occupy the data of that block. As discussed above, block hashing is done using the popular SHA256 algorithm.

The Secure Hash Algorithm 256 bit or the SHA256 is a popular hashing algorithm where the new blocks mined are stored in a hash table and the respective keys are generated. It is said that if a key is obtained from a SHA256 hashed block, then it takes about 120 year to find the original key value of that block. Since the time taken to crack this hashing algorithm is so long, Sha256 is the most reliable algorithm. There is also SHA512, even the this 512 bit hashing is more secure than the 256 bit hashing, the latter is preferred because of the length of the keys. The users tend to forget the keys of a block since it is an alphanumeric key.

The steps involved in the blockchain block hashing in the architecture are as follows:

- **Architecture design:** The first step in blockchain hashing is to design the architecture. The architecture of the blockchain is responsible for defining the structure of the blocks that make up the chain, how they are linked together, and how they are secured. This design will include the specific hashing algorithm that will be used to generate the hash values for each block.
- **Block creation:** Once the architecture has been designed, the next step is to create a new block. The block will contain a header and a body. The header contains metadata about the block, such as its version number, the hash value of the previous block, and the timestamp of when the block was created. The body contains the actual transaction data that is being recorded on the blockchain.

- **Hashing**: The next step in blockchain hashing is to generate the hash value for the block. This is done by running the entire block through the specified hashing algorithm. The resulting hash value is a fixed-length string of characters that uniquely identifies the block.
- **Validation**: Once the hash value has been generated, it is important to validate that it is correct. This is done by comparing the hash value of the current block with the hash value of the previous block. If the two hash values match, then the block is considered valid and can be added to the chain.
- **Consensus**: In order to ensure that the blockchain is secure and accurate, a consensus mechanism must be in place. This mechanism is responsible for ensuring that all nodes on the network agree on the current state of the blockchain. The most commonly used consensus mechanism is proof-of-work, which requires nodes to solve a cryptographic puzzle in order to add a new block to the chain.
- **Block addition**: Once a block has been validated and consensus has been reached, it can be added to the blockchain. The block is added to the end of the chain and linked to the previous block using its hash value. This creates an immutable and transparent record of all transactions that have occurred on the blockchain.

Overall, the process of blockchain hashing involves designing the architecture of the blockchain, creating new blocks, generating hash values, validating blocks, reaching consensus, and adding blocks to the chain. This process is repeated for each new block that is added to the chain, creating a secure and decentralized ledger that can be used for a wide range of applications.

CHAPTER 5

MODULE DESCRIPTION

5.1 PROJECT MODULES

The system mainly constitutes of two modules:

1. Server
 - Node file
 - Transaction files
2. Client
 - Frontend
 - Form for entering the details

5.2 MODULE DESCRIPTION

5.2.1 BLOCK

SHA-256 applies to build a block within a blockchain. We will discuss here in the context of the Bitcoin blockchain and understand how this tie into the role of miners. The minors are in the process of building blocks, and these blocks are added to a blockchain to build out what the Bitcoin blockchain will be.

Block:	# 1
Timestamp:	2023-03-15 19:08:38.230046
	[]
Data:	
Previous:	0
Hash:	ad3b1321d7df3cb1d8a7579ebba9951146i

In the above image, the generated hash would look like *00001acbm010gfh1010xxx*. I would like to point out that this hash has four leading zeros. The four leading zeros describes whether the block is valid or not. For practical purposes, you will see that this hash is corresponding to the nonce, and the block number is corresponding to the available data. Since the hash has four leading zeroes, therefore, it is a valid block.

If we make any change in the data section, it will give the completely different hash that can be shown in the below image.

Block:	# 1
Timestamp:	2023-03-15 19:08:38.230046
Data:	[]
Previous:	0
Hash:	ad3b1321d7df3cb1d8a7579ebba99511467

If the newly generating hash does not have four leading zeroes, then it will not a valid block. To make the block valid, we will do it by using the field called **nonce**.

Nonce stands for **a Number Used Once** in a cryptographic communication such that the block's hash meets a certain criterion. This criterion could be

generated a hash that must have its leading four digits to be zero. Thus, the generated hash would look like *00001acbm010gfh1010xxx*.

A nonce is basically a random number which figures out how you can make this specific block provide you with a valid hash. The way you can do this is by changing the nonce manually. Generally, the miner starts with a Nonce value of 1 and keeps on incrementing it until the generated hash meets the specified criterion. Thus, it may take several iterations until the desired hash with four leading zeros is generated. The expected time for generating a block in the bitcoin system is 10 minutes. Once the miner successfully mines the block, he releases that block in the system and making it the last block in the chain.

In Anders Brownworth Hash Program, when we click mine button as shown in the image, it will give the valid block. This block has a unique nonce with hash leading four zeroes in the beginning.

5.2.2 **BLOCK HASHING**

A blockchain is a concept of storing data digitally. This data comes in blocks. These blocks are chained together and make the data **immutable**. When a block of data is chained with the other blocks, its data can never be changed again. It will be publicly available to anyone who wants to see it ever again and will be available in the same sequence in which it was added to the blockchain. Nobody can change the information once it is added in the blockchain.

In the previous section, we have learned how to build the block and the cryptographic hashing ties into the whole process. Here you can see how blockchain uses all these concepts together to preserve the integrity of a full blockchain.

Imagine a **bunch of blocks** of transaction data as of the following image.

In the above images, you can see that block assembles in **chronological order**. Block number one is followed by block number two, then block number three, then block number four, and you can continue it as long as you want. Here, you will find that there is a **block number field, data field, nonce field, hash**

value field, and **previous field**. The previous field is corresponding to the hash value field of the previous block.

Block:	# 1
Timestamp:	2023-03-15 19:08:38.230046
Data:	[]
Previous:	0
Hash:	ad3b1321d7df3cb1d8a7579ebba9951146i

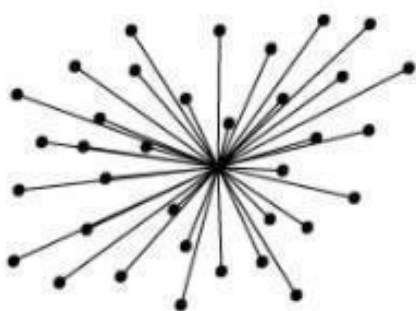
In the above images, you can see that block assembles in **chronological order**. Block number one is followed by block number two, then block number three, then block number four, and you can continue it as long as you want. Here, you will find that there is a **block number field**, **data field**, **nonce field**, **hash value field**, and **previous field**. The previous field is corresponding to the hash value field of the previous block.

We know that every block in a blockchain is cryptographically tied to the next block. In the above example, the previous field in block one is zero because the block one doesn't have a previous hash value, so its value is zero. In block number two, you will find that there is a hash value in the previous field, which references to the previous block hash value. This process is continuing until the last block.

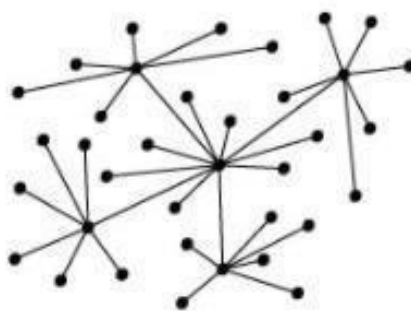
Now imagine if the data in any block is altered. Let us say that we have altered in block number 2, the data in block 2 is now different, and mining the block also gets a new **signature**. The signature that corresponds with this new set of data is no longer chained to other blocks. It just breaks block number 2 because the hash is no longer valid and it also invalidates every single block that comes after it to the end of the chain. It indicates to other users of this blockchain that some data in block 2 has been altered, and because the blockchain should be immutable, they reject this change by shifting back to a previous record of the blockchain where all the blocks are still chained together. This is the main advantage of the blockchain.

5.2.3 BLOCKCHAIN ARCHITECTURE

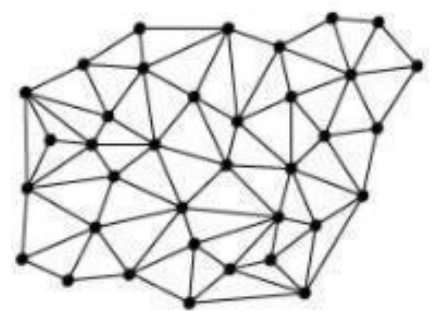
A blockchain is an open financial ledger or record in which every transaction is authenticated and authorized. A blockchain is designed as a decentralized network of millions of computers, commonly referred to as nodes. It's a distributed database architecture in which each node plays the role of a network administrator who voluntarily joins the network. Since there's no centralized information in a blockchain architecture, a blockchain is literally impossible to hack.



centralised



decentralised



distributed

CONCLUSION

In conclusion, designing a blockchain block hashing system requires careful consideration of several key components, including the hash function, consensus mechanism, data structure, security, and scalability. These elements work together to ensure the integrity and immutability of the blockchain, as well as its ability to achieve consensus in a decentralized and trustless manner.

A secure and cryptographically robust hash function, such as SHA-256, is critical to creating a unique digital fingerprint of each block in the chain. The consensus mechanism, such as proof-of-work or proof-of-stake, is responsible for ensuring that all nodes on the network agree on the state of the blockchain, while the data structure used to store the blockchain can impact its performance and scalability.

Ensuring the security of the blockchain is essential to prevent attacks such as collision attacks and pre-image attacks, which could compromise the integrity of the chain. At the same time, scalability is crucial to enable the blockchain to handle large volumes of transactions without incurring high fees or long confirmation times.

A well-designed blockchain block hashing system requires careful consideration of multiple factors to create a secure, efficient, and scalable blockchain that can achieve consensus in a decentralized and trustless manner. By prioritizing these factors, blockchain projects can build a strong foundation for a blockchain network that can provide robust and reliable services to users.

In addition to the technical components, it's also essential to consider the community aspect of blockchain block hashing projects. Blockchain networks are often community-driven, and the success of the project can depend on the level of engagement and participation from the community. This can include developers, miners, and users who contribute to the network in various ways.

Moreover, it's crucial to consider the governance structure of the blockchain network. Governance is essential to ensure that the network can adapt and evolve over time, while also maintaining the principles of decentralization and consensus. A well-designed governance system can allow for decision-

making that aligns with the interests of the community and promotes the long-term sustainability of the network.

Finally, blockchain block hashing projects should prioritize transparency and openness. Blockchain technology was developed to enable transparent and immutable records of transactions, and this principle should be extended to the development and management of the blockchain network itself. Openness and transparency can help build trust among stakeholders and promote collaboration and innovation within the community.

Designing a blockchain block hashing project requires a holistic approach that considers technical components, community engagement, governance structures, and transparency. By prioritizing these factors, blockchain projects can build robust and reliable networks that can provide real-world solutions and applications. Ultimately, a successful blockchain project should strive to make a positive impact on society while embodying the principles of decentralization, trustlessness, and transparency that underpin blockchain technology.

Blockchain hashing is a critical component of the architecture of blockchain systems. Hashing algorithms are used to secure the data stored on the blockchain and to validate transactions. The effectiveness of a hashing algorithm is based on its security, scalability, and efficiency.

Existing blockchain systems such as Bitcoin and Ethereum provide valuable insights into the use of hashing algorithms in blockchain technology. However, as the technology continues to evolve, there will be new challenges and opportunities for improvement.

System analysis of blockchain hashing is an important step in evaluating the effectiveness of existing systems and identifying areas for improvement. By enhancing security, increasing scalability, improving efficiency, promoting interoperability, and ensuring regulatory compliance, blockchain hashing can continue to play a crucial role in the architecture of blockchain systems.

FUTURE WORKS

The future works on the blockchain block hashing project will likely focus on improving the efficiency and security of the hashing process. This could involve developing new algorithms or optimizing existing ones to reduce the time and energy required to complete each hash. Another area of focus may be to enhance the resistance of the hashing process against attacks or attempts to manipulate the blockchain. This could involve exploring new cryptographic techniques or improving the verification process for each block. Additionally, as blockchain technology becomes more widely adopted, there may be a need for standardization in the block hashing process to ensure interoperability between different blockchain platforms. Finally, as the blockchain industry continues to evolve, there may be new use cases and applications for block hashing that have yet to be discovered, and researchers and developers will need to remain agile and adaptable to keep pace with these changes.

Furthermore, the development of quantum computing technology may pose a threat to the security of the hashing process. As quantum computers become more powerful, they may be able to break the cryptographic algorithms currently used to secure the blockchain. As such, future works may focus on developing quantum-resistant hashing algorithms to ensure the continued security of the blockchain.

APPENDIX 1

SAMPLE CODE

CLIENT

Add Transaction Screen

```
import React, { useState } from "react";
import Button from "@mui/material/Button";
import TextField from "@mui/material/TextField";
import Dialog from "@mui/material/Dialog";
import DialogActions from "@mui/material/DialogActions";
import DialogContent from "@mui/material/DialogContent";
import DialogContentText from "@mui/material/DialogContentText";
import DialogTitle from "@mui/material/DialogTitle";
import axios from "axios";

export default function AddTxnForm({ open, handleClose }) {
  const [state, setState] = useState({
    sender: "",
    receiver: "",
    amount: 0,
  });

  const handleSubmit = async () => {
    handleClose();
    console.log(state);
    const response = await axios.post(
      "http://localhost:5000/add_transaction",
      state
    );
    console.log(response.data);
    alert(response.data.message);
  };

  const handleChange = (evt) => {
    setState({
      ...state,
      [evt.target.name]: evt.target.value,
    });
  };
}
```

});

```

};
return (
  <div>
    <Dialog open={open} onClose={handleClose}>
      <DialogTitle>Send Crypto Coins</DialogTitle>
      <DialogContent>
        <DialogContentText>
          Please Enter Information Carefully.
        </DialogContentText>
        <TextField
          autoFocus
          margin="dense"
          name="sender"
          label="Sender Address"
          type="text"
          fullWidth
          variant="outlined"
          value={state.sender}
          onChange={handleChange}
        />
        <TextField
          autoFocus
          margin="dense"
          name="receiver"
          label="Receiver Address"
          type="text"
          fullWidth
          variant="outlined"
          value={state.receiver}
          onChange={handleChange}
        />
        <TextField
          autoFocus
          margin="dense"
          name="amount"
          label="Amount"
          type="number"
          fullWidth
          variant="outlined"
          value={state.amt}
          onChange={handleChange}

```

```

    />
  </DialogContent>
  <DialogActions>
    <Button onClick={handleClose}>Cancel</Button>
    <Button variant="contained" onClick={handleSubmit}>
      Send
    </Button>
  </DialogActions>
</Dialog>
</div>
);
}

```

Blockchain Screen

Block.js

```

import * as React from "react";
import Box from "@mui/material/Box";
import { makeStyles } from "@mui/styles";
import { Card, Grid, Typography } from "@mui/material";

```

```

const useStyle = makeStyles(() => {
  return {
    block: {
      padding: 10,
      background: "#bfaee8",
    },
    row: {
      paddingBottom: 2,
    },
    head: {
      width: "100px",
      display: "inline-block",
    },
  };
});

export default function Block({ block }) {
  const classes = useStyle();
  return (
    <Box sx={{ minWidth: 375, padding: 2 }}>

```

```

<div className={classes.block}>
  <Grid>
    <div className={classes.head}>
      <Typography>Block:</Typography>
    </div>
    <input
      type="text"
      style={{ width: "250px" }}
      value={`# ${block.index}`}
      readOnly
    />
  </Grid>
  <br />
  <Grid>
    <div className={classes.head}>
      <Typography>Timestamp:</Typography>
    </div>
    <input
      type="text"
      style={{ width: "250px" }}
      value={block.timestamp}
      readOnly
    />
  </Grid>
  <br />
  <Grid>
    <div className={classes.head}>
      <Typography>Data:</Typography>
    </div>
    <textarea
      type="text"
      style={{ width: "250px" }}
      rows={8}
      value={JSON.stringify(block.transactions)}
      readOnly
    />
  </Grid>
  <br />
  <Grid>
    <div className={classes.head}>
      <Typography>Previous:</Typography>

```

```

    </div>
    <input
      type="text"
      style={{ width: "250px" }}
      value={block.previous_hash}
      readOnly
    />
  </Grid>
  <br />
  <Grid>
    <div className={classes.head}>
      <Typography>Hash:</Typography>
    </div>
    <input
      type="text"
      style={{ width: "250px" }}
      value={block.hash}
      readOnly
    />
  </Grid>
  <br />
  <br />
</div>
</Box>
);
}

```

Blockchain_Screen.js

```

import { Container, Grid } from "@mui/material";
import { makeStyles } from "@mui/styles";
import React, { useEffect, useState } from "react";
import Header from "../header/header";
import Block from "../block";
import Chain from "../link";
const useStyles = makeStyles((theme) => ({
  root: {},
  blockchain: {
    padding: 5,
  },
}));
function BlockchainScreen() {

```

```

const classes = useStyles();
const [blocks, setblocks] = useState([]);
useEffect(() => {
  const fetchData = async () => {
    const response = await fetch("http://localhost:5000/get_chain");
    const result = await response.json();
    if (result) setblocks(result.chain);
  };
  fetchData();
}, []);

console.log(blocks);

return (
  <>
    <Header />
    <br />
    <br />
    <br />
    <Container>
      <Grid
        className={classes.blockchain}
        container
        spacing={5}
        alignItems={"center"}
        justifyContent={"center"}
      >
        {blocks.map((block) => (
          <>
            <Block key={block.index} block={block} />
            <Chain key={block.hash} />
          </>
        ))}
      </Grid>
    </Container>
  </>
);
}

export default BlockChainScreen;

```

Link.js

```
import React from "react";
import InsertLinkIcon from "@mui/icons-material/InsertLink";
import { makeStyles } from "@mui/styles";
```

```
const useStyle = makeStyles(() => {
  return {
    root: {
      padding: 5,
    },
  };
});
```

```
function Chain() {
  const classes = useStyle();
  return (
    <div className={classes.root}>
      <InsertLinkIcon />
    </div>
  );
}
```

```
export default Chain;
```

Header

```
import * as React from "react";
import AppBar from "@mui/material/AppBar";
import Box from "@mui/material/Box";
import Toolbar from "@mui/material/Toolbar";
import IconButton from "@mui/material/IconButton";
import Typography from "@mui/material/Typography";
import Menu from "@mui/material/Menu";
import MenuItem from "@mui/material/MenuItem";
import Container from "@mui/material/Container";
import Button from "@mui/material/Button";
import AddTxnForm from "../add_transaction_screen/add_txn_screen";
import axios from "axios";
import { useNavigate } from "react-router-dom";
```

```
const Header = () => {
```

```

const [anchorElNav, setAnchorElNav] = React.useState(null);

const handleOpenNavMenu = (event) => {
  setAnchorElNav(event.currentTarget);
};

const handleCloseNavMenu = () => {
  setAnchorElNav(null);
};

const history = useNavigate();
const [open, setOpen] = React.useState(false);

const handleClickOpen = () => {
  setOpen(true);
};

const handleClose = () => {
  setOpen(false);
};

const mineBlock = async () => {
  const response = await axios.get("http://localhost:5000/mine_block");
  console.log(response.data);
  // history("/");
  alert(response.data.message + "Refresh Page to see new blocks!");
};

const pages = [
  {
    page: "Blockchain",
    func: () => {
      history("/");
    },
  },
  {
    page: "Add Transaction",
    func: handleClickOpen,
  },
  {
    page: "Mine New Block",
    func: mineBlock,
  },

```

```

    },
  ];

  return (
    <AppBar position="static">
      <Container maxWidth="xl">
        <Toolbar disableGutters>
          <Typography
            variant="h6"
            noWrap
            component="div"
            sx={{ mr: 2, display: { xs: "none", md: "flex" } }}
          >
            ILLUMINATI
          </Typography>

          <Box sx={{ flexGrow: 1, display: { xs: "flex", md: "none" } }}>
            <IconButton
              size="large"
              aria-label="account of current user"
              aria-controls="menu-appbar"
              aria-haspopup="true"
              onClick={handleOpenNavMenu}
              color="inherit"
            >
              <MenuIcon />
            </IconButton>
            <Menu
              id="menu-appbar"
              anchorEl={anchorElNav}
              anchorOrigin={{
                vertical: "bottom",
                horizontal: "left",
              }}
              keepMounted
              transformOrigin={{
                vertical: "top",
                horizontal: "left",
              }}
              open={Boolean(anchorElNav)}
              onClose={handleCloseNavMenu}
            >

```

```

      sx={{
        display: { xs: "block", md: "none" },
      }}
    >
    {pages.map((page) => (
      <MenuItem key={page.page} onClick={handleCloseNavMenu}>
        <Typography textAlign="center">{page.page}</Typography>
      </MenuItem>
    ))}
  </Menu>
</Box>
<Typography
  variant="h6"
  noWrap
  component="div"
  sx={{ flexGrow: 1, display: { xs: "flex", md: "none" } }}
>
  LOGO
</Typography>
<Box sx={{ flexGrow: 1, display: { xs: "none", md: "flex" } }}>
  {pages.map((page) => (
    <Button
      key={page.page}
      onClick={page.func}
      sx={{ my: 2, color: "white", display: "block" }}
    >
      {page.page}
    </Button>
  ))}
</Box>
</Toolbar>
</Container>
<AddTxnForm open={open} handleClose={handleClose} />
</AppBar>
);
};
export default Header;

```

App.js

```
import logo from './logo.svg';
```

```

import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

```

```
export default App;
```

SERVER

node.py

```

import datetime
import hashlib
import json
from flask import Flask, jsonify, request
import requests
from uuid import uuid4
from urllib.parse import urlparse
from flask_cors import CORS

```

```
# Building Blockchain
```

```

class Blockchain:
    def __init__(self):
        self.chain = []
        self.transactions = []

```

```

self.create_block(proof=1, previous_hash = '0')
self.nodes = set()

def create_block(self, proof, previous_hash):
    block = {
        "index": len(self.chain) + 1,
        "timestamp": str(datetime.datetime.now()),
        "proof": proof,
        "previous_hash": previous_hash,
        "transactions": self.transactions,
    }
    hash = self.hash(block)
    block["hash"] = hash
    self.transactions = []
    self.chain.append(block)
    return block

def get_previous_block(self):
    return self.chain[-1]

def proof_of_work(self, previous_proof):
    new_proof = 1
    check_proof = False
    while check_proof is False:
        hash_operation = hashlib.sha256(str(new_proof**2 -
previous_proof**2).encode()).hexdigest()
        if hash_operation[:5] == '00000':
            check_proof = True
        else:
            new_proof += 1
    return new_proof

def hash(self, block):
    encoded_block = json.dumps(block, sort_keys=True).encode()
    return hashlib.sha256(encoded_block).hexdigest()

def is_chain_valid(self, chain):
    previous_block = chain[0]
    block_index = 1
    while block_index < len(chain):

```

```

        block = chain[block_index]
        if block["previous_hash"] != self.hash(previous_block):
            return False
        previous_proof = previous_block["proof"]
        proof = block["proof"]
        hash_operation = hashlib.sha256(str(proof**2 -
previous_proof**2).encode()).hexdigest()
        if hash_operation[:4] != "0000":
            return False
        previous_block = block
        block_index += 1
    return True

```

```

def add_transactions(self, sender, receiver, amount):
    self.transactions.append({
        "sender": sender,
        "receiver": receiver,
        "amount": amount
    })
    previous_block = self.get_previous_block()
    return previous_block["index"] + 1

```

```

def add_node(self, address):
    paresed_url = urlparse(address)
    self.nodes.add(paresed_url.netloc)

```

```

def replace_chain(self):
    network = self.nodes
    longest_chain = None
    max_length = len(self.chain)
    for node in network:
        response = requests.get(f'http://{node}/get_chain')
        if response.status_code == 200:
            length = response.json()["length"]
            chain = response.json()["chain"]
            if length > max_length and self.is_chain_valid(chain):
                max_length = length
                longest_chain = chain
    if longest_chain:
        self.chain = longest_chain
    return True

```

```

        return False
# Mining Blockchain

app = Flask(__name__)
CORS(app)
node_address = str(uuid4()).replace('-', '')

blockchain = Blockchain()

@app.route("/mine_block", methods=['GET'])
def mine_block():
    previous_block = blockchain.get_previous_block()
    previous_proof = previous_block['proof']
    previous_hash = previous_block["hash"]
    proof = blockchain.proof_of_work(previous_proof)
    blockchain.add_transactions(sender=node_address, receiver="Miner",
amount=10)
    block = blockchain.create_block(proof, previous_hash)

    response = {
        "message": "Congratulation, you just mined a block!",
        "index": block["index"],
        "timestamp": block["timestamp"],
        "proof": block["proof"],
        "previous_hash": block["previous_hash"],
        "transactions": block["transactions"],
    }

    return jsonify(response), 200

@app.route("/get_chain", methods=["GET"])
def get_chain():
    response = {
        "chain": blockchain.chain,
        "active_nodes": list(blockchain.nodes),
        "length": len(blockchain.chain)
    }

    return jsonify(response)

@app.route("/is_valid", methods=["GET"])

```

```

def is_valid():
    is_valid = blockchain.is_chain_valid(blockchain.chain)
    if is_valid:
        response = {'message': "All Good!. The Blockchain is valid."}
    else:
        response = {'message': "We have a problem. The Blockchain is not valid."}
    return jsonify(response), 200

```

```

@app.route("/add_transaction", methods=['POST'])
def add_transactions():
    json = request.json
    transaction_keys = ["sender", "receiver", "amount"]

    if not all(key in json for key in transaction_keys):
        return "Some elements of the transaction are missing", 400

    index = blockchain.add_transactions(json["sender"], json["receiver"],
    json["amount"])
    response = {
        "message": f"This transaction will be added to Block {index}"
    }
    return jsonify(response), 201

```

Decentralisation

```

@app.route("/connect_node", methods=['POST'])
def connect_node():
    json = request.json
    nodes = json.get('nodes')
    if nodes is None:
        return "No Node", 400
    for node in nodes:
        print(node)
        blockchain.add_node(node)
    response = {
        "message": "All the nodes are now connected. The ViCoin Blockchain
now contains the following nodes:",
        "total_nodes": list(blockchain.nodes)
    }
    return jsonify(response), 201

```

```

@app.route("/replace_chain", methods=["GET"])
def replace_chain():
    is_chain_replaced = blockchain.replace_chain()
    if is_chain_replaced:
        response = {'message': "The nodes had different chains so the chain was
replaced by the longest one.", "new_chain": blockchain.chain}
    else:
        response = {'message': "All good. The chain is the largest one.",
"actual_chain": blockchain.chain}
    return jsonify(response), 200

app.run(host='0.0.0.0', port=5000, debug=True)

```

transaction.json

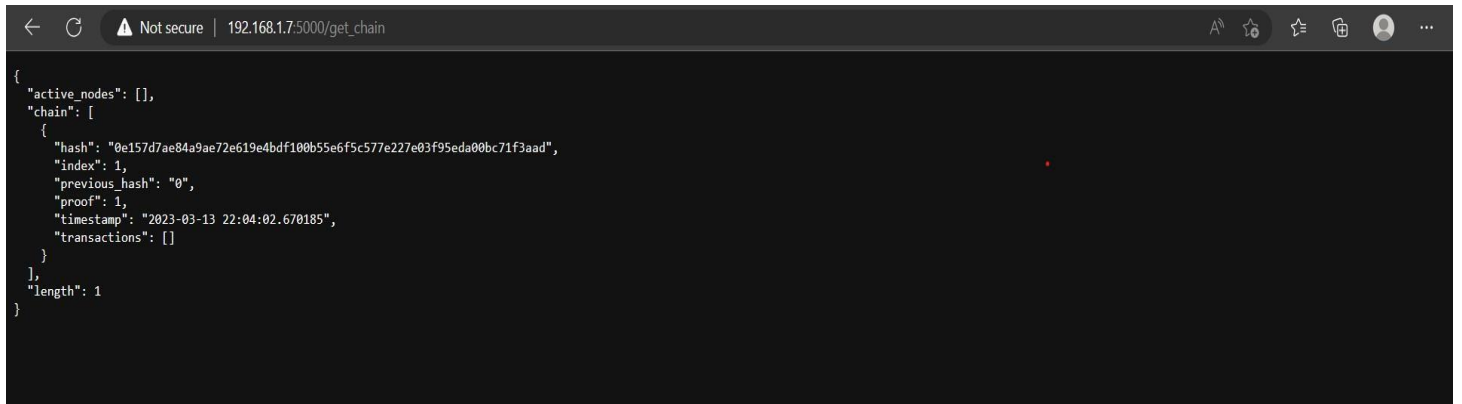
```

{
    "sender": "",
    "receiver": "",
    "amount": 0
}

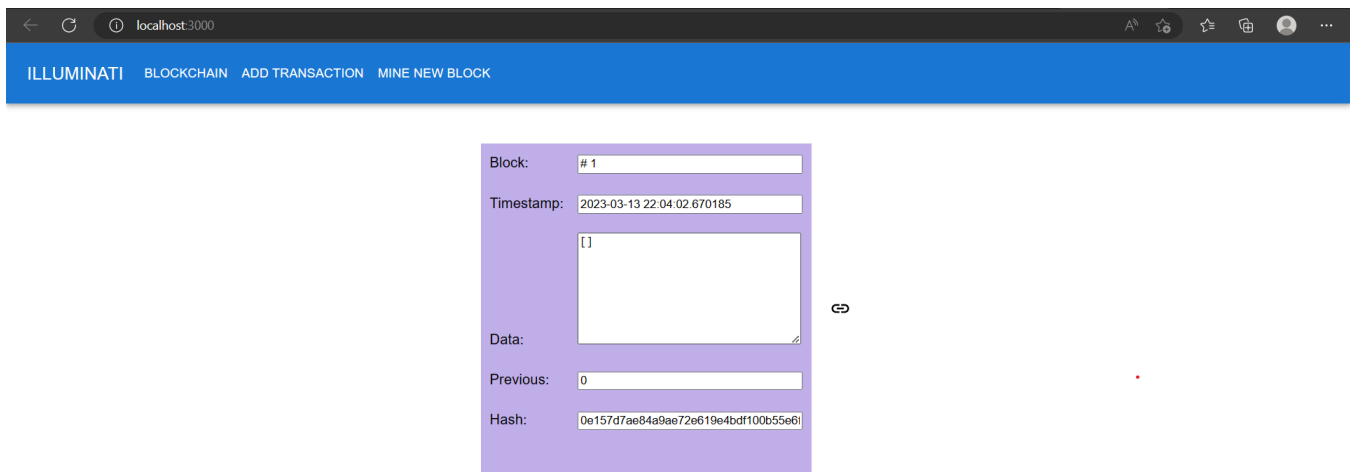
```

APPENDIX 2

SCREENSHOTS



```
{
  "active_nodes": [],
  "chain": [
    {
      "hash": "0e157d7ae84a9ae72e619e4bdf100b55e6f5c577e227e03f95eda00bc71f3aad",
      "index": 1,
      "previous_hash": "0",
      "proof": 1,
      "timestamp": "2023-03-13 22:04:02.670185",
      "transactions": []
    }
  ],
  "length": 1
}
```



ILLUMINATI BLOCKCHAIN ADD TRANSACTION MINE NEW BLOCK

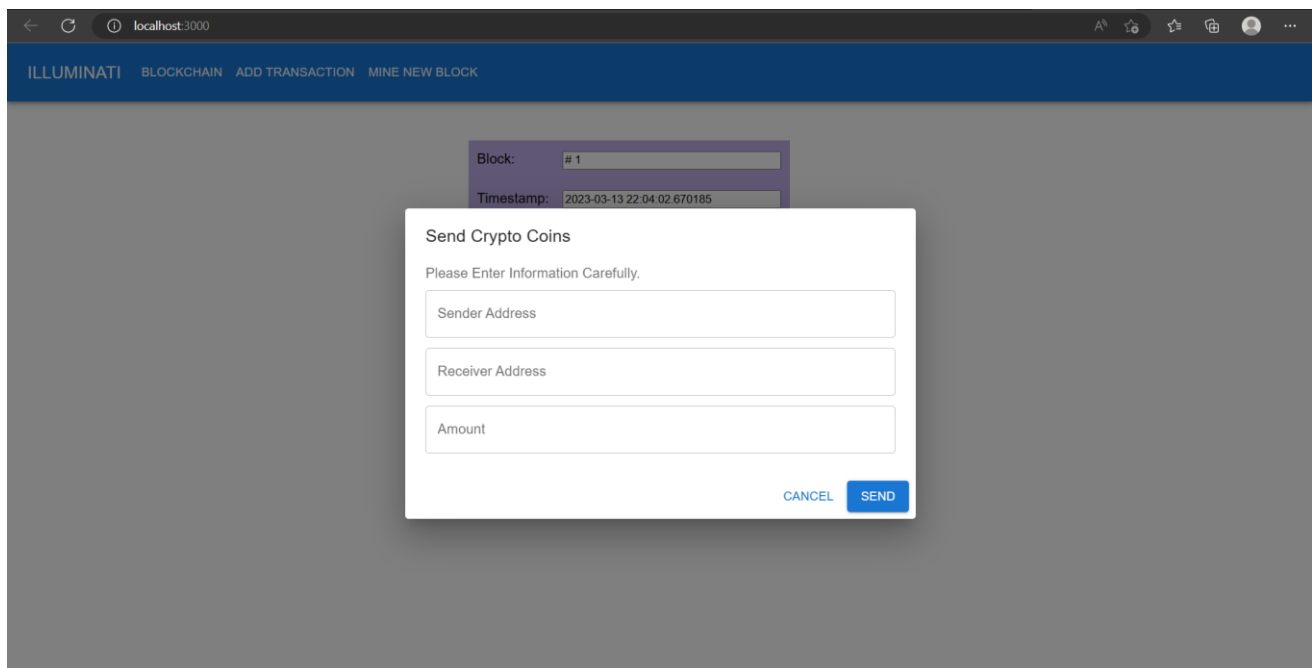
Block:

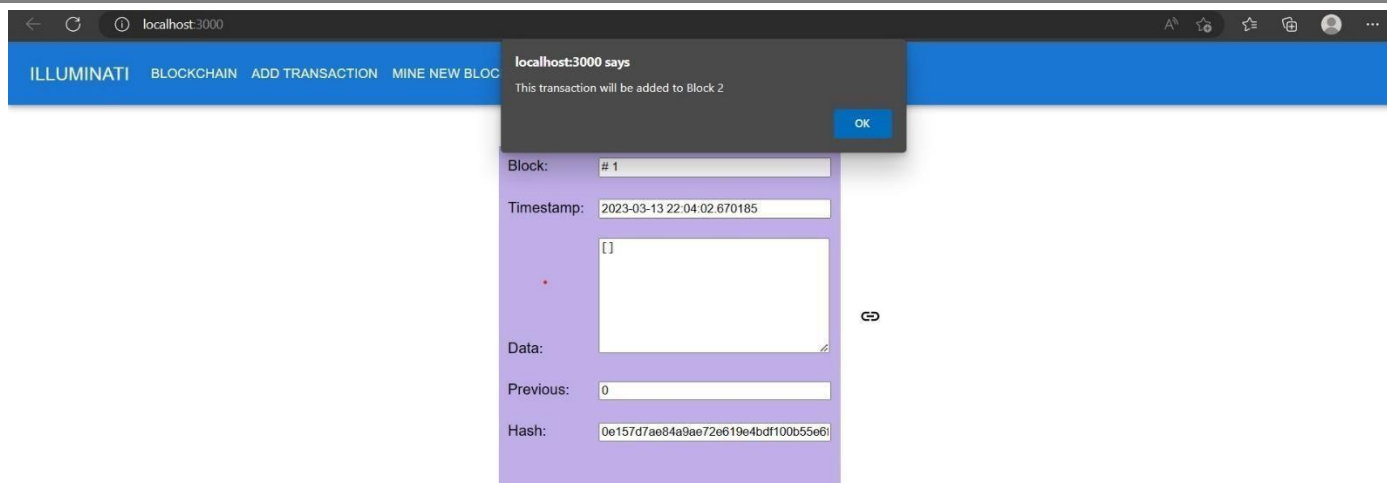
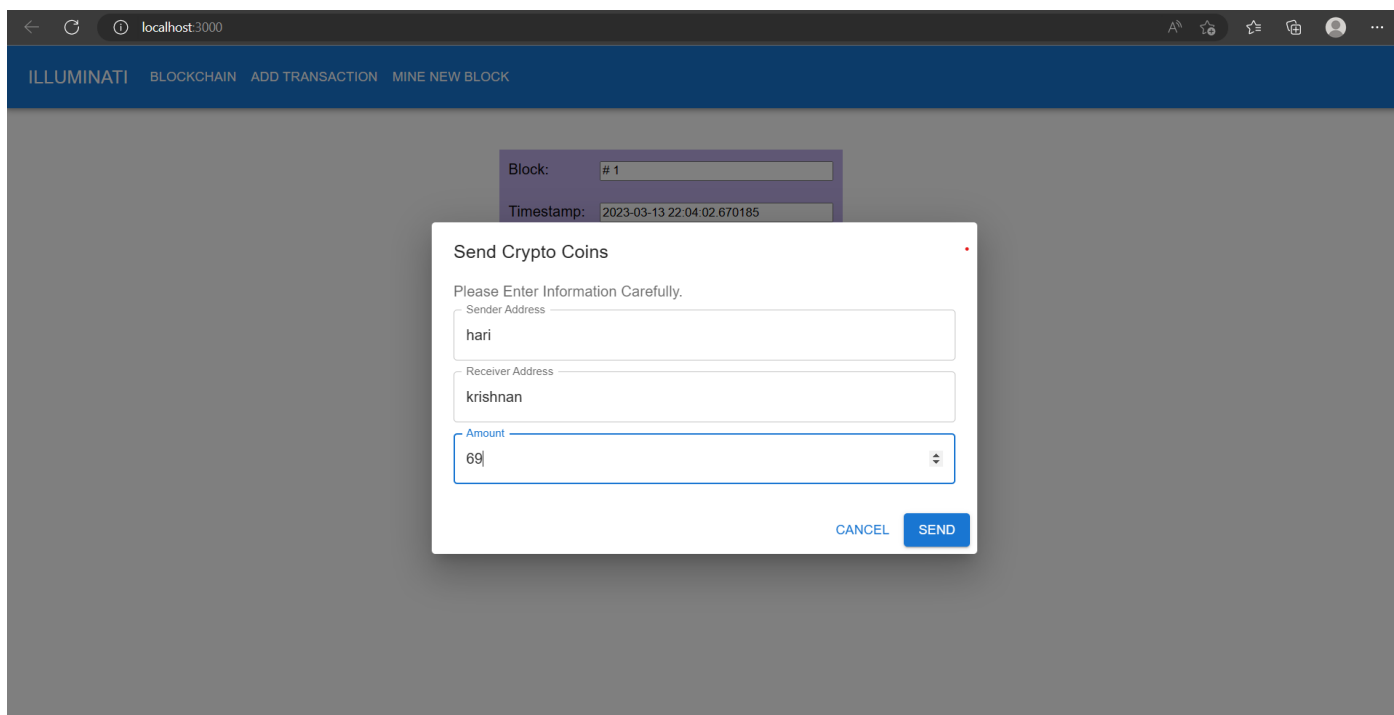
Timestamp:

Data:

Previous:

Hash:





ILLUMINATI

BLOCKCHAIN

ADD TRANSACTION

MINE NEW BLOCK

localhost:3000 says

Congratulation, you just mined a block!Refresh Page to see new blocks!

OK

Block: # 1

Timestamp: 2023-03-13 22:16:54.577107

[]

Data:

Previous: 0

Hash: 19c9126803b28515c5649d6bca56556d96

← ↻ ⓘ localhost:3000

ILLUMINATI

BLOCKCHAIN

ADD TRANSACTION

MINE NEW BLOCK

Block: # 1

Timestamp: 2023-03-13 22:16:54.577107

[]

Data:

Previous: 0

Hash: 19c9126803b28515c5649d6bca56556d96

Block: # 2

Timestamp: 2023-03-13 22:17:48.287035

[{"amount": "69", "receiver": "krishnan", "sender": "hari"}, {"amount": "10", "receiver": "Miner", "sender": "465ae9094f4242109ef853b7748fe039"}]

Data:

Previous: 19c9126803b28515c5649d6bca56556d96

Hash: e58844721ab1daf849463e348f20f402022f

```
← 192.168.1.7:5000/get_chain
{
  "active_nodes": [],
  "chain": [
    {
      "hash": "19c9126803b28515c5649d6bca56556d96c04404942cab4972b723fa7b055d78",
      "index": 1,
      "previous_hash": "0",
      "proof": 1,
      "timestamp": "2023-03-13 22:16:54.577107",
      "transactions": []
    },
    {
      "hash": "e58844721ab1daf849463e348f20f402022fb459ec0c22430012adc6139728df",
      "index": 2,
      "previous_hash": "19c9126803b28515c5649d6bca56556d96c04404942cab4972b723fa7b055d78",
      "proof": 632238,
      "timestamp": "2023-03-13 22:17:48.287035",
      "transactions": [
        {
          "amount": "69",
          "receiver": "krishnan",
          "sender": "hari"
        },
        {
          "amount": "10",
          "receiver": "Miner",
          "sender": "465ae9094f4242109ef853b7748fe039"
        }
      ]
    }
  ],
  "length": 2
}
```

ILLUMINATI BLOCKCHAIN ADD TRANSACTION MINE NEW BLOCK

localhost:3000 says
This transaction will be added to Block 3

OK

Block: # 1

Timestamp: 2023-03-13 22:16:54.577107

Data: []

Previous: 0

Hash: 19c9126803b28515c5649d6bca56556d96

↔

Block: # 2

Timestamp: 2023-03-13 22:17:48.287035

Data: [{"amount": "69", "receiver": "krishnan", "sender": "hari"}, {"amount": "10", "receiver": "Miner", "sender": "465ae9094f4242109ef853b7748fe039"}]

Previous: 19c9126803b28515c5649d6bca56556d96

Hash: e58844721ab1daf849463e348f20f402022fb459ec0c22430012adc6139728df

↔

Block: # 1

Timestamp: 2023-03-13 22:16:54.577107

Data: []

Previous: 0

Hash: 19c9126803b28515c5649d6bca56556d96

↔

Block: # 2

Timestamp: 2023-03-13 22:17:48.287035

Data: [{"amount": "69", "receiver": "krishnan", "sender": "hari"}, {"amount": "10", "receiver": "Miner", "sender": "465ae9094f4242109ef853b7748fe039"}]

Previous: 19c9126803b28515c5649d6bca56556d96

Hash: e58844721ab1daf849463e348f20f04020221

Block: # 1

Timestamp: 2023-03-13 22:16:54.577107

Data: []

Previous: 0

Hash: 19c9126803b28515c5649d6bca56556d96

↔

Block: # 2

Timestamp: 2023-03-13 22:17:48.287035

Data: [{"amount": "69", "receiver": "krishnan", "sender": "hari"}, {"amount": "10", "receiver": "Miner", "sender": "465ae9094f4242109ef853b7748fe039"}]

Previous: 19c9126803b28515c5649d6bca56556d96

Hash: e58844721ab1daf849463e348f20f04020221

↔

Block: # 3

Timestamp: 2023-03-13 22:20:10.433633

Data: [{"amount": "100", "receiver": "tyrion lannister", "sender": "thomas shelby"}, {"amount": "10", "receiver": "Miner", "sender": "465ae9094f4242109ef853b7748fe039"}]

Previous: e58844721ab1daf849463e348f20f04020221

Hash: 8ac4841646868a59eb63e7af5fcb8eccc9b6

```
← ↻ ⚠ Not secure | 192.168.1.7:5000/get_chain 🔍 📌 🏠 👤 ...

{
  "active_nodes": [],
  "chain": [
    {
      "hash": "19c9126883b28515c5649d6bca56556d96c04404942cab4972b723fa7b055d78",
      "index": 1,
      "previous_hash": "0",
      "proof": 1,
      "timestamp": "2023-03-13 22:16:54.577107",
      "transactions": []
    },
    {
      "hash": "e58844721ab1daf849463e348f20f402022fb459ec8c22430012adc6139728df",
      "index": 2,
      "previous_hash": "19c9126883b28515c5649d6bca56556d96c04404942cab4972b723fa7b055d78",
      "proof": 632238,
      "timestamp": "2023-03-13 22:17:48.287035",
      "transactions": [
        {
          "amount": "50",
          "receiver": "krishnan",
          "sender": "hari"
        },
        {
          "amount": "10",
          "receiver": "Miner",
          "sender": "465ae9094f42421109ef853b7748fe039"
        }
      ]
    },
    {
      "hash": "8ac4841646868a59eb63e7af5fcbfbfcc9b039981f13ec1ae827cbcc0cccf58a",
      "index": 3,
      "previous_hash": "e58844721ab1daf849463e348f20f402022fb459ec8c22430012adc6139728df",
      "proof": 403001,
      "timestamp": "2023-03-13 22:20:10.433633",
      "transactions": [
        {
          "amount": "100",
          "receiver": "tyrion lannister",
          "sender": "thomas shelby"
        },
        {
          "amount": "10",
          "receiver": "Miner",
          "sender": "465ae9094f42421109ef853b7748fe039"
        }
      ]
    }
  ],
  "length": 3
}
```


RESEARCH PAPER

BLOCKCHAIN HASHING Prototype of Blockchain Mining and Hashing with the help of Secured Hashing Algorithm of SHA256 bits

line 1: 1st Given Name Surname
line 2: dept. name of organization
(of Affiliation)
line 3: name of organization
(of Affiliation)
line 4: City, Country
line 5: email address or ORCID

line 1: 2nd Given Name Surname
line 2: dept. name of organization
(of Affiliation)
line 3: name of organization
(of Affiliation)
line 4: City, Country
line 5: email address or ORCID

line 1: 5th Given Name Surname
line 2: dept. name of organization
(of Affiliation)
line 3: name of organization
(of Affiliation)
line 4: City, Country
line 5: email address or ORCID

Abstract - This paper presents a Blockchain Hashing prototype that focuses on SHA256 bit secure hashing algorithm implementation for blockchain mining and hashing. The goal is to incorporate a powerful hashing algorithm into the Blockchain system in order to improve security measures. The prototype addresses a number of topics, such as the creation of a safe mining procedure, effective hashing methods, and the incorporation of the SHA256 bits algorithm into the current Blockchain network. The suggested prototype makes use of the cryptographic characteristics of SHA256 bits to guarantee the integrity and immutability of data stored in the Blockchain. It has been found via thorough testing and analysis that the prototype significantly improves the security and functionality of the Blockchain system, making it a workable solution for practical uses. The Blockchain network's integrity is ensured by the additional layer of security provided by the SHA256 bits hashing algorithm.

Keywords: Blockchain Hashing, Mining, Hashing Algorithm, SHA256 bits, Security, Cryptography, Integrity, Immutability, Efficiency.

I. INTRODUCTION

Blockchain technology, which offers a safe and decentralized way to record and verify transactions, has completely changed a number of businesses. The fundamental idea of this technology is hashing, which guarantees the immutability and integrity of data recorded on the blockchain. To that end, a Blockchain Mining and Hashing prototype has been created, utilizing the ultra-secure SHA256 hashing algorithm.

In the blockchain ecosystem, hashing is essential because it serves as a cryptographic function that transforms any size of data into a fixed-length string of letters. Because of its stability and ability to withstand collision assaults, SHA256 (Secure Hash Algorithm 256-bit) is one of the most popular algorithms used for this purpose. It is extremely unlikely that two different inputs will produce the same hash output because it generates a unique 256-bit hash value for each input.

To guarantee the durability and security of the blockchain network, the prototype integrates hashing and blockchain mining. In the context of blockchain technology, mining is the process of resolving challenging mathematical puzzles in order to approve transactions and add them to the ledger. A network of individuals known as miners creates new blocks and confirms transactions through mining.

The SHA256 hashing algorithm is utilized by the prototype during the mining process, which makes it a crucial part of blockchain security. The immutability of the entire chain is ensured by the hash value contained in each block of the blockchain, which represents the data from the previous block. Any modification to the contents would result in a shift in the hash value of the block, hence rendering the chain invalid. SHA256 is therefore essential for spotting attempts at tampering and preserving the blockchain's integrity.

A new block is generated at a constant rate because to the prototype's incorporation of the mining difficulty idea. The network is kept safe from bad actors by periodically adjusting the difficulty level to make it computationally tough to construct new blocks. Miners compete to solve the

challenging mathematical riddles, and the first one to do so wins money. This encourages mining while bolstering the blockchain's security.

Overall, the importance of the SHA256 hashing algorithm in the larger blockchain ecosystem is demonstrated by the Blockchain Mining and Hashing prototype. It is a pillar of data integrity and security due to its resilience to attacks. The prototype demonstrates the core ideas of blockchain technology by fusing hashing and mining, guaranteeing the decentralized, transparent, and unchangeable character of the blockchain network. It is an effective tool for companies and sectors looking to take use of blockchain technology while protecting their information against manipulation and unwanted access.

II. RELATED WORKS

1. The blockchain hashing method based on PRCA is optimized in the paper by Fu et al. [1]. The authors suggested a cutting-edge strategy to improve the blockchain system's effectiveness and security. They conducted in-depth simulations to assess the performance of their suggested algorithm and contrasted it with other hashing algorithms that are widely employed in blockchain systems.

2. Performance analysis of cryptographic hash algorithms appropriate for blockchain application was carried out by Kuznetsov et al. [2]. To determine which hash function would work best for blockchain applications, the authors evaluated the security and efficiency of several hash algorithms. Developers can use their findings to select the right hash algorithm for their blockchain systems.

3. For bitcoin mining, Pham et al. [3] introduced a double SHA-256 hardware architecture with a small message expander. A hardware implementation of the double SHA-256 algorithm, which is commonly used in bitcoin mining, was devised by the authors. Their suggested architecture sought to lower the needed hardware resources in order to increase bitcoin mining efficiency.

4. Manishankar et al.'s study [4] investigated the application of blockchain technology to cloud-based IoT data security. The authors provided a framework for the safe cloud storage and exchange of IoT data, as well as a discussion of the advantages and difficulties of fusing blockchain technology with IoT. Their work uses blockchain technology to improve the security of Internet of Things networks.

5. Virtual Block Group (VBG), a scalable blockchain concept that makes use of distributed hash tables and partial node storage, was proposed by Yu et al. [5]. The authors

proposed an innovative solution to typical blockchain systems' scalability problems. Their method makes it possible to store and retrieve data on blockchain networks in an effective manner.

6. Amanat et al. [6] proposed a blockchain and cloud computing-based method for the safe exchange and archiving of electronic medical records. The authors talked about how important it is to store and share healthcare data securely and suggested a blockchain-based solution that protects data integrity and privacy. The security and interoperability of electronic health records are enhanced by their research.

7. The idea of Imagechain, which makes use of blockchain technology for safe image management and storage, was presented by Koptyra and Ogiela [7]. The authors talked about the difficulties in storing photos and offered a blockchain-based solution that gives images access control, immutability, and traceability. Their findings present a viable use of blockchain technology for picture management.

8. The security of cryptocurrencies in blockchain technology was examined by Ghosh et al. [8]. The writers outlined the most recent methods for securing bitcoins, as well as the difficulties and opportunities that lie ahead in this area. Their research sheds light on the security issues surrounding cryptocurrencies built on blockchain technology.

9. In a healthcare communication system, Haque et al. [9] presented an information security solution for electronic medical records (EMR) based on blockchain technology. The authors provided a blockchain-based solution that guarantees the availability, integrity, and confidentiality of EMR data in addition to discussing the flaws in the current EMR systems. The security and privacy of medical data are improved by their research.

10. Through blockchain radio access networks, Ling et al. [10] presented a revolutionary idea called Hash Access, which provides reliable grant-free IoT access. The authors provided a blockchain-based solution that does away with the requirement for conventional access control systems while also discussing the difficulties associated with IoT access management. Their study provides a fresh viewpoint on decentralized IoT device security.

III. EXISTING SYSTEM

The existing system, which is a prototype for Blockchain Mining and Hashing utilizing the SHA256 algorithm, has several notable drawbacks: High Processing Power Requirement: The SHA256 algorithm used in blockchain

mining demands a significant amount of processing power. This requirement leads to the utilization of specialized hardware, such as ASICs (Application-Specific Integrated Circuits), which can be expensive to acquire and maintain. This results in a concentration of mining power among a small group of individuals or organizations with substantial financial resources. Consequently, the decentralized and democratic nature of the blockchain is compromised, as control becomes centralized among those who can afford the necessary hardware.

Escalating Energy Consumption: Blockchain mining consumes a substantial amount of energy. The energy consumption associated with mining operations is comparable to that of entire nations, which raises concerns about the sustainability of blockchain technology and its impact on the environment. As more miners join the network and the blockchain ecosystem expands, energy consumption is expected to rise further, straining existing power infrastructures and contributing to increased carbon emissions.

Scalability Issues: The current system faces challenges in terms of scalability. As the blockchain continues to grow with the addition of more transactions, all nodes within the network find it increasingly difficult to store, verify, and transmit the entire blockchain. This exponential growth in blockchain size can lead to longer confirmation times for transactions and network congestion, ultimately affecting the efficiency and utility of the blockchain.

Security Vulnerabilities: While the SHA256 algorithm has proven to be secure thus far, there is no guarantee against future advancements in cryptanalysis or the discovery of vulnerabilities. The reliance solely on SHA256 as a hashing technique may pose a risk to the overall integrity and security of the blockchain system. If vulnerabilities are identified, it could jeopardize the trustworthiness of the entire blockchain network.

Privacy and Anonymity Concerns: The current system lacks robust privacy and anonymity features. Transactions recorded on the blockchain are transparent and publicly accessible, raising concerns about the privacy of sensitive data. Although additional layers or privacy-focused protocols can be implemented to enhance privacy, the inherent openness of the blockchain may limit its applicability in sectors or use cases where data privacy is of paramount importance.

In summary, while the prototype for Blockchain Mining and Hashing with SHA256 shows promise, these drawbacks need to be addressed for blockchain technology to achieve widespread adoption and long-term success. Solutions are required to mitigate the concentration of mining power, reduce energy consumption, enhance scalability, strengthen security, and improve privacy features in blockchain

systems. Addressing these challenges is crucial to realizing the full potential of blockchain technology across various industries and applications.

IV. PROPOSED SYSTEM

The goal of the proposed effort is to create a working prototype for Blockchain mining and hashing with the secure SHA256 hashing algorithm. Due to its decentralized and immutable characteristics, blockchain technology has drawn a lot of attention. These qualities make it perfect for a variety of applications, including supply chain management, smart contracts, and cryptocurrencies. To confirm and add new transactions to the Blockchain, one of the most important aspects of Blockchain is the mining process, which entails resolving challenging mathematical riddles. A strong hashing algorithm that guarantees the confidentiality and integrity of the data is necessary for this procedure. Thus, the suggested prototype will use the Secure Hashing Algorithm (SHA256).

A popular cryptographic hash function called SHA256 (Secure Hash Algorithm 256-bit) generates a 256-bit (32-byte) hash value. It is made to withstand many kinds of attacks, such as pre-image, collision, and brute force attacks. The integrity and security of the Blockchain mining process can be guaranteed by the prototype's use of SHA256.

Building a decentralized network of nodes that take part in mining will be the prototype's task. These nodes will compete to use the SHA256 hashing technique to solve challenging mathematical puzzles. Tokens for newly created cryptocurrencies or transaction fees will be awarded to the first node to identify the right answer. Every transaction is confirmed and uploaded to the Blockchain in a transparent and safe manner thanks to the mining and hashing process.

The prototype will also incorporate elements like data encryption, consensus methods, and timestamping to improve the overall security and effectiveness of the Blockchain. The data encryption feature will shield private information from unwanted access, and the timestamping feature will guarantee that transactions are done in the correct chronological order. The implementation of consensus techniques, like Proof of Work or Proof of Stake, will facilitate the verification and validation of transactions by members of the network.

To sum up, the proposed effort is to create a Blockchain mining and hashing prototype utilizing the secure SHA256 hashing algorithm. By enabling safe and effective transaction validation and inclusion to the Blockchain, this prototype will protect the data's validity and integrity.

V. SYSTEM ARCHITECTURE

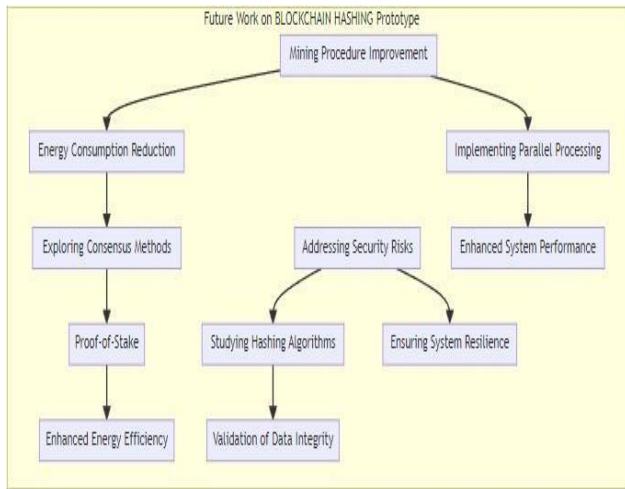


Fig. 1. System Architecture

VI. METHODOLOGY

Module 1: Blockchain Mining - The Foundation of Blockchain Security and Consensus Blockchain mining stands as a fundamental pillar in the world of blockchain technology, underpinning the security, integrity, and decentralization of the entire network. This module is dedicated to unraveling the intricate workings of blockchain mining, shedding light on its significance, and exploring the mechanisms that make it the backbone of modern blockchain systems. At its core, blockchain mining revolves around the critical tasks of adding new blocks to the blockchain and verifying the validity of existing ones. These seemingly simple actions are the bedrock upon which the entire blockchain ecosystem is built.

Mining is a process that calls upon computational power to solve complex mathematical puzzles. These puzzles, often referred to as "proof-of-work" algorithms, serve two primary purposes: they secure the network and validate transactions. Miners across the network compete to be the first to solve these puzzles, which are intentionally designed to be resource-intensive and time-consuming. In the world of blockchain mining, it's a race against time and computational power. Miners engage in a highly competitive environment, vying to find the solution to the proof-of-work puzzle before their peers. The first miner to successfully solve the puzzle gets the privilege of adding a new block to the blockchain. This competitive nature of mining serves a crucial function. It ensures that no single entity or group can easily dominate the network, preventing centralization and preserving the decentralized nature of blockchain systems.

Mining isn't just a selfless act of securing the network; it's also an opportunity for miners to earn rewards. Typically, miners are rewarded with cryptocurrency coins native to the blockchain they are mining. For instance, Bitcoin miners receive newly minted bitcoins as their reward. This incentive system not only motivates miners to participate in maintaining the network but also contributes to the distribution of cryptocurrency tokens, fostering wider adoption and decentralization.

Beyond the rewards, mining plays a pivotal role in maintaining the consistency and integrity of the blockchain ledger across all nodes in the network. Every miner verifies the transactions in a block before adding it to the blockchain. This consensus mechanism ensures that every participant in the network holds an identical copy of the blockchain ledger. In essence, blockchain mining synchronizes the decentralized network, ensuring that all nodes agree on the validity of transactions and the order in which they are added to the blockchain. This consensus mechanism is a cornerstone of blockchain's trustworthiness and immutability.

As blockchain technology continues to evolve, mining methods and algorithms are also subject to innovation. Alternative consensus mechanisms like proof-of-stake and delegated proof-of-stake are gaining popularity due to their energy efficiency and scalability benefits. The industry is witnessing a shift toward more sustainable and eco-friendly mining practices. Module 1 has delved deep into the realm of blockchain mining, revealing its critical role in securing the network, incentivizing participation, and ensuring the synchronized consensus of the blockchain ledger. Understanding mining is not only essential for blockchain enthusiasts but also for anyone looking to grasp the intricate mechanics behind the decentralized future of technology. As the world of blockchain continues to advance, mining remains an ever-evolving and dynamic field that will shape the future of decentralized systems.

SHA-256 hash function

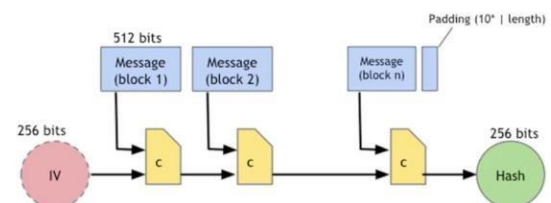


Fig.2. SHA-256 hash function

Module 2: SHA256 Bits Hashing Algorithm. In the ever-evolving landscape of blockchain technology, Module 2 takes center stage, shedding light on the critical role played by the SHA256 Bits Hashing Algorithm. As the foundation upon which our proposed system is built, this algorithm stands as a paragon of cryptographic excellence, offering unparalleled security and data integrity.

At its core, hashing is the process of taking raw data and converting it into a unique, fixed-length string of characters, known as a hash value. This transformation is akin to creating a digital fingerprint for the original data, allowing for quick and efficient verification. However, not all hash algorithms are created equal, and in the realm of blockchain, security is paramount. This is where SHA256 takes the spotlight. The acronym "SHA" stands for Secure Hash Algorithm, and "256" denotes the number of bits in the resultant hash value. SHA256, specifically, has earned its reputation as one of the most robust and secure hashing algorithms in the blockchain domain. Its strength lies in its ability to consistently generate a distinct 256-bit hash value for every unique input, making it extremely difficult for any two different inputs to produce the same hash.

In our system, Module 2 harnesses the power of the SHA256 algorithm to hash not only individual transactions but also the hash of the preceding block in the blockchain. This dual application ensures that every block in the chain has a unique digital fingerprint, derived from both its own data and the data contained in the preceding block. This interlinking of blocks through their hash values forms the basis of the blockchain's immutability and trustworthiness. One of the key advantages of the SHA256 algorithm is its resistance to tampering. Even a minor change in the original data will result in a drastically different hash value. This property creates an effective safeguard against data manipulation and ensures that once a block is added to the blockchain, it becomes practically immutable. This level of security is essential for maintaining the integrity of the blockchain, particularly in applications where trust and accuracy are paramount.

In summary, Module 2 underscores the significance of the SHA256 Bits Hashing Algorithm within our system. It serves as the cornerstone of our blockchain's security and reliability, providing the means to create digital fingerprints for data that are both unique and highly resistant to alterations. With SHA256 as our cryptographic backbone, we are equipped to build a robust and trustworthy blockchain infrastructure capable of meeting the demands of diverse industries and applications. As we delve deeper into the intricacies of our proposed system, the role of SHA256 in ensuring data integrity and security will continue to be a central theme.

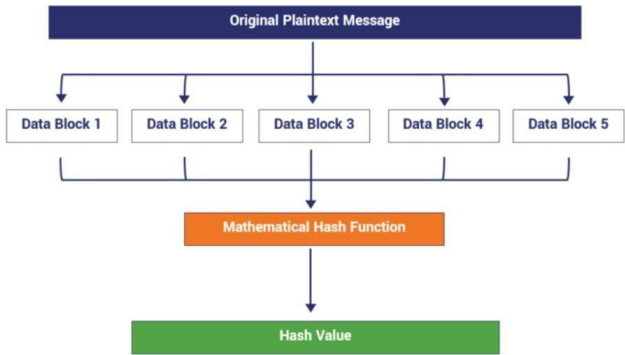


Fig.3. Hashing working process

Module 3: Blockchain Hashing Prototype. The Blockchain Hashing Prototype module represents a pivotal phase in the realization of the proposed system, as it brings to life the core elements of the system's hashing and mining functionalities. In this module, our primary objective is to translate the theoretical framework into a tangible, functional model, shedding light on the intricate workings of a blockchain network. This module encompasses a series of essential activities, each contributing to the development and implementation of the prototype:

The foundation of our prototype begins with the creation of a multi-node blockchain network. This step is critical to mimic the decentralized nature of real-world blockchain ecosystems. Nodes are strategically distributed to establish a network that mirrors the concept of a distributed ledger. Mining is the heartbeat of any blockchain, and in this module, we integrate the mining process with computational power. By doing so, we simulate the process of miners competing to solve complex cryptographic puzzles, thus validating and adding new transactions to the blockchain. The cornerstone of blockchain security lies in its hashing algorithm. Our prototype leverages the SHA256 hashing algorithm, renowned for its cryptographic strength. It is employed in the creation of blocks and the verification of transactions, ensuring the immutability and integrity of the data recorded on the blockchain. We conduct thorough performance assessments to gauge the efficiency of our prototype. These tests measure the speed of transaction processing, block creation times, and overall system responsiveness. By identifying potential bottlenecks or areas for improvement, we aim to enhance the prototype's efficiency. Security is paramount in blockchain technology. Our module scrutinizes the security measures in place, assessing the resilience of the system against potential attacks. We explore ways to fortify the prototype's defenses, protecting it against vulnerabilities and threats. Scalability is a vital aspect of any blockchain network, and our module assesses the system's ability to grow and handle an increasing volume of transactions. This scalability testing ensures that the prototype can adapt to evolving demands. In

essence, the Blockchain Hashing Prototype module serves a dual purpose. Firstly, it acts as a proof-of-concept, demonstrating the viability of the proposed system and its core components. Secondly, it lays a strong foundation for future enhancements and optimizations. As the world of blockchain technology continues to evolve, the insights gained from this module will inform future developments, allowing us to stay at the forefront of this transformative field.

In summary, this module is a crucial stepping stone in our journey towards a fully operational blockchain system, where the principles of decentralization, security, and immutability come together to revolutionize various industries and redefine the way we conduct transactions and share data.

VII. RESULT AND DISCUSSION

The BLOCKCHAIN HASHING prototype is a method that uses SHA256 bits, a secure hashing technique, to combine hashing and mining. The objective of this prototype is to offer a dependable and effective method for cryptographic functions on a blockchain network.

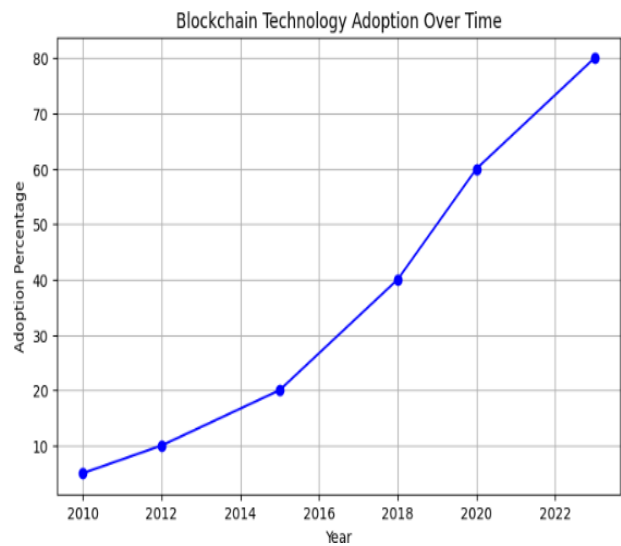


Fig.4.Blockchain Technology Adoption Over Time

In the context of blockchain technology, mining is the process of creating new blocks by resolving challenging mathematical puzzles. The proof-of-work consensus process used by the BLOCKCHAIN HASHING prototype requires miners to discover a hash value that satisfies specific requirements. These hash values are produced using SHA256 bits, a popular hashing method.

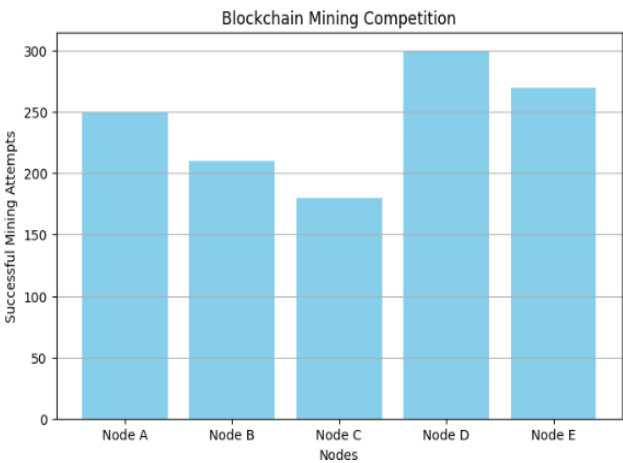


Fig.5.Blockchain Mining Competition

The blockchain network's security and integrity are guaranteed by the SHA256 algorithm. It accepts input data and outputs a fixed-sized result, commonly referred to as the hash value. Reversing the hash from the original data is nearly impossible because even a slight alteration in the input data yields a whole different hash value. This guarantees the blockchain data's integrity and immutability. The BLOCKCHAIN HASHING prototype offers a high level of security against tampering and cyberattacks by employing the SHA256 bits algorithm for hashing. Furthermore, the decentralized and transparent addition of new blocks to the blockchain is guaranteed by the mining mechanism of the prototype. This enhances the blockchain network's overall dependability and efficiency.

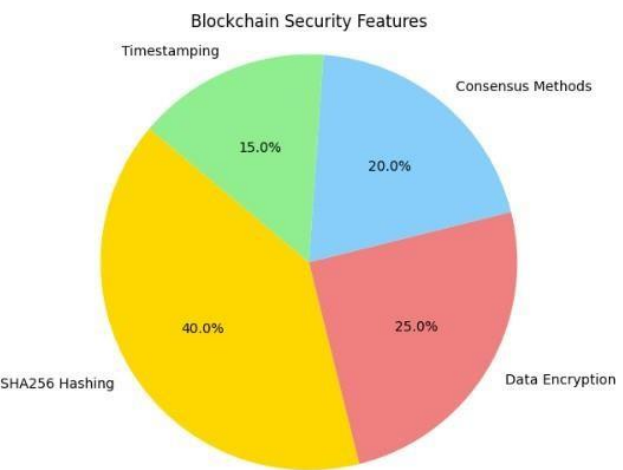


Fig.6 .Blockchain Security Features

In conclusion, the BLOCKCHAIN HASHING prototype

leverages the safe SHA256 hashing algorithm to provide a strong and efficient system for mining and cryptographic operations within a blockchain network. Because of its implementation, which guarantees the blockchain data's integrity, security, and transparency, it is a reliable solution for a variety of businesses that depend on blockchain technology.

VIII. CONCLUSION

In conclusion, the development and implementation of a blockchain hashing prototype, leveraging the power of blockchain mining and the robust security provided by the SHA-256 bit hashing algorithm, represents a significant milestone in the world of digital technology and decentralized ledger systems. The fusion of these two critical components, blockchain mining and SHA-256 hashing, forms the backbone of many of the most well-known cryptocurrencies, such as Bitcoin.

This prototype showcases the immense potential of blockchain technology to revolutionize various industries, ranging from finance and supply chain management to healthcare and beyond. By harnessing the principles of distributed ledger technology, it ensures transparency, immutability, and security of data and transactions, addressing longstanding challenges associated with centralized systems. The utilization of the SHA-256 algorithm, known for its cryptographic strength, adds an extra layer of security to the blockchain network. It guarantees that each block's content remains unaltered, providing integrity to the entire chain. This level of security instills trust in the system, making it suitable for critical applications where data accuracy and immutability are paramount.

Moreover, this prototype serves as a testament to the ongoing innovation within the blockchain ecosystem. As blockchain technology continues to evolve, we can anticipate further advancements in mining algorithms, hashing methods, and consensus mechanisms, all aimed at enhancing the scalability, speed, and energy efficiency of blockchain networks. These developments will open up new possibilities and use cases, driving the widespread adoption of blockchain technology across various sectors. In conclusion, the prototype of blockchain mining and hashing, fortified by the SHA-256 algorithm, exemplifies the continuous progress and transformative potential of blockchain technology. It sets the stage for a future where decentralized, secure, and transparent systems become the norm, reshaping industries and fostering innovation on a global scale.

IX. FUTURE WORK

Future work on the BLOCKCHAIN HASHING prototype will be directed towards exploring and refining the mining and hashing processes, specifically employing secure hashing algorithms such as SHA256 bits. A primary focus will be on enhancing the efficiency and scalability of mining operations. One approach to achieve this is by implementing parallel processing techniques, enabling multiple mining operations to run concurrently. This strategy aims to accelerate the process of block production and transaction verification, thereby improving the overall productivity of the system.

Additionally, addressing the issue of energy consumption associated with blockchain mining is of paramount importance. The energy-intensive nature of current mining methodologies, particularly those based on the proof-of-work consensus mechanism, necessitates a search for more energy-efficient alternatives. Exploring other consensus mechanisms, such as proof-of-stake, which requires significantly less computational power compared to proof-of-work, could be a viable solution to reduce the overall energy footprint of the mining process.

Moreover, ensuring the robustness of the system against potential security threats is crucial. This involves a thorough investigation into the security aspects of hashing algorithms to identify and mitigate any inherent risks or vulnerabilities. The objective is to reinforce the security framework of the blockchain hashing prototype to withstand a wide range of potential security challenges.

The overarching aim of future endeavors is to augment the security, sustainability, and performance of the blockchain hashing prototype. By addressing these key areas, the intention is to foster widespread adoption and application of the prototype across various sectors, leveraging its improved capabilities to meet the diverse needs of different industries. The continuous evolution and enhancement of the system are expected to contribute significantly to the advancement and maturation of blockchain technology as a whole.

REFERENCES

- [1] Fu, J., Qiao, S., Huang, Y., Si, X., Li, B., & Yuan, C. (2020). A study on the optimization of blockchain hashing algorithm based on PRCA. *Security and Communication Networks*, 2020, 1-12.
- [2] Kuznetsov, A., Oleshko, I., Tymchenko, V., Lisitsky, K., Rodinko, M., & Kolhatin, A. (2021). Performance analysis of cryptographic hash functions suitable for use in blockchain. *International Journal of Computer Network & Information Security*, 13(2), 1-15.

- [3] Pham, H. L., Tran, T. H., Phan, T. D., Le, V. T. D., Lam, D. K., & Nakashima, Y. (2020). Double SHA-256 hardware architecture with compact message expander for bitcoin mining. *IEEE Access*, 8, 139634-139646.
- [4] Manishankar, S., Dechamma, T. S., & Anoop, A. (2021, August). Securing IoT Data in the Cloud with Blockchain Technology. In *2021 Asian Conference on Innovation in Technology (ASIANCON)* (pp. 1-8). IEEE.
- [5] Yu, B., Li, X., & Zhao, H. (2020). Virtual block group: A scalable blockchain model with partial node storage and distributed hash table. *The Computer Journal*, 63(10), 1524-1536.
- [6] Amanat, A., Rizwan, M., Maple, C., Zikria, Y. B., Almadhor, A. S., & Kim, S. W. (2022). Blockchain and cloud computing-based secure electronic healthcare records storage and sharing. *Frontiers in Public Health*, 10, 938707.
- [7] Koptyra, K., & Ogiela, M. R. (2020). Imagechain—application of blockchain technology for images. *Sensors*, 21(1), 82.
- [8] Ghosh, A., Gupta, S., Dua, A., & Kumar, N. (2020). Security of Cryptocurrencies in blockchain technology: State-of-art, challenges and future prospects. *Journal of Network and Computer Applications*, 163, 102635.
- [9] Haque, R., Sarwar, H., Kabir, S. R., Forhat, R., Sadeq, M. J., Akhtaruzzaman, M., & Haque, N. (2021). Blockchain-based information security of electronic medical records (EMR) in a healthcare communication system. In *Intelligent Computing and Innovation on Data Science: Proceedings of ICTIDS 2019* (pp. 641-650). Singapore: Springer Singapore.
- [10] Ling, X., Le, Y., Wang, J., & Ding, Z. (2020). Hash access: Trustworthy grant-free IoT access enabled by blockchain radio access networks. *IEEE Network*, 34(1), 54-61.



SATHYABAMA



INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

CATEGORY-1 UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE



www.sathyabama.ac.in

SCHOOL OF COMPUTING

Department of Computer Science and Engineering

Certificate of Presentation

This is to certify that

S. DON INIGO

of

Sathyabama Institute of Science and Technology

has presented a paper entitled

Blockchain Hashing Prototype of Blockchain mining and Hashing Algo of sha 256 bits

in the

International Conference on Cognitive Computing and Artificial Intelligence (ICCCAI-2024) in association with
Taylor's University, Malaysia held on 7th & 8th March 2024.

Dr. Sayan Kumar Ray
Head, School of Computer Science
Taylor's University

Dr. L. Lakshmanan
HOD, CSE Dept
Sathyabama Institute of Science and Technology

Dr. T. Sasikala
Dean, SOC
Sathyabama Institute of Science and Technology



SATHYABAMA



INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

CATEGORY-1 UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC [I2B Status by UGC] Approved by AICTE



www.sathyabama.ac.in

SCHOOL OF COMPUTING

Department of Computer Science and Engineering

Certificate of Presentation

This is to certify that

S. BHUVANESHWARAN

of

Sathyabama Institute of science and Technology

has presented a paper entitled

Blockchain Hashing Prototype of Blockchain mining and Hashing Algo of sha 256 bits
in the

International Conference on Cognitive Computing and Artificial Intelligence (ICCCAI-2024) in association with
Taylor's University, Malaysia held on 7th & 8th March 2024.

Dr. Sayan Kumar Ray
Head, School of Computer Science
Taylor's University

Dr. L. Lakshmanan
HOD, CSE Dept
Sathyabama Institute of Science and Technology

Dr. T. Sasikala
Dean, SOC
Sathyabama Institute of Science and Technology