



Master of Science in Internetworking (MINT)

Project Report

On

*An Analysis of the Security of
Cryptocurrency Implementations and
Proposed Solutions*

By Gurpreet Singh

Under the Guidance of

Michael Spaling

Table of Contents

ABSTRACT	5
INTRODUCTION	6
WHAT IS BLOCKCHAIN?.....	6
BASICS OF BLOCKCHAIN	6
MODERN USE OF BLOCKCHAIN:	8
WHAT ARE CRYPTOCURRENCIES?	13
PROBLEMS OF CRYPTOCURRENCIES (BITCOIN)	13
KEY COMPONENTS OF CRYPTOCURRENCIES.....	14
THE 'CRYPTO' IN CRYPTOCURRENCIES	17
BLOCKCHAIN TECHNOLOGY.....	19
BLOCKCHAIN TECHNOLOGY DEFINITION	19
DECENTRALIZED AND CENTRALIZED NETWORK.....	19
BLOCKCHAIN TECHNOLOGY ARCHITECTURE.....	20
BLOCKCHAIN TECHNOLOGY FEATURES	22
BLOCKCHAIN STORAGE STRUCTURE.....	23
TYPES OF BLOCKCHAIN	24
BLOCKCHAIN TECHNOLOGY SECURITY.....	25
BLOCKCHAIN SECURITY ISSUES	25
BLOCKCHAIN SECURITY VULNERABILITIES	26
BLOCKCHAIN SECURITY ISSUES PER LAYERED ARCHITECTURE.....	27
NETWORK LAYER	30
HASHING	32
PROVIDING HASH VALUES FOR ANY TYPE OF DATA QUICKLY	32
PATTERNS OF HASHING DATA	35
INDEPENDENT HASHING.....	36
APPLICATIONS OF HASHING	39
HASHING RELATION TO BLOCKCHAIN	46
DESIGN OF HASH FUNCTION	47
SECURITY OF CRYPTOGRAPHIC HASH FUNCTIONS	48
CRYPTOGRAPHY	57
THE IDEA OF CRYPTOGRAPHY	58
SYMMETRIC CRYPTOGRAPHY	59
ASYMMETRIC CRYPTOGRAPHY	60
ASYMMETRIC CRYPTOGRAPHY IN THE BLOCKCHAIN	62
DIGITAL SIGNATURES.....	64
HOW DIGITAL SIGNATURES ARE USED IN BLOCKCHAIN.....	68
WEAKNESS OF ASYMMETRIC CRYPTOGRAPHY	69
TWO TYPES OF DIGITAL SIGNATURE	70
MINING AND CONSENSUS.....	72
MINING A BLOCK	72
CRYPTOCURRENCY TRANSACTION VALIDATION PROCESSES	73
DOUBLE SPENDING PROBLEM.....	76

RACE ATTACK.....	76
51% ATTACK.....	77
IMPLICATIONS OF THE ATTACK	77
WHAT ARE THE RISKS WHICH ARE INVOLVED WITH 51% ATTACK?.....	78
WHICH PLATFORMS HAVE FACED A 51% ATTACK.....	79
HOW CAN WE AVOID A 51% ATTACK[].....	79
WHAT ARE ITS CHANCES TO RECUR?.....	80
LAB COMPONENT.....	80
COMPONENTS REQUIRED TO CREATE A CRYPTOCURRENCY	80
LINKING ALL THE COMPONENTS TOGETHER	82
PRIVATE KEY REGENERATION	86
OUTCOME	98
INSECURE CRYPTOGRAPHIC HASH FUNCTIONS:	98
CHOSEN-PREFIX COLLISION ON MD5	100
CHOSEN-PREFIX COLLISION ON SHA1	102
WHAT IS A QUANTUM COMPUTER?.....	104
WHO INVENTED THE QUANTUM COMPUTER?	104
QUANTUM SUPREMACY’S POTENTIAL IMPACT ON CRYPTOCURRENCIES	105
CAN IT BREAK ANY CRYPTOCURRENCY?.....	105
SHOR’S FACTOR ALGORITHM	106
QUANTUM COMPUTERS AND CRYPTOGRAPHY	106
THE IMPACT OF QUANTUM COMPUTING ON CRYPTOGRAPHY	107
GROVER’S ALGORITHM	107
QUANTUM COMPUTER EFFECTS ON HASHING	108
POST-QUANTUM CRYPTOGRAPHY	109
CRYPTOGRAPHY: PRE-QUANTUM, QUANTUM, AND POST-QUANTUM	109
CHALLENGES ASSOCIATED WITH POST-QUANTUM CRYPTOGRAPHY	109
SECURITY IN A POST-QUANTUM WORLD.....	110
CONCLUSION.....	111

Table of Figures

FIGURE 1 CONVENTIONAL CENTRALIZED LEDGER WITH TRUSTED THIRD-PARTIES.....	7
FIGURE 2 A TYPICAL EXAMPLE OF BLOCKCHAIN TECHNOLOGY	7
FIGURE 3 DIFFERENT BLOCKCHAIN APPLICATIONS AND USE CASES.....	8
FIGURE 4 CENTRALIZED NETWORK FIGURE 5 DECENTRALIZED NETWORK	20
FIGURE 6 BLOCKCHAIN STRUCTURE	23
FIGURE 7 HASH VALUE OF STANDARD HASH FUNCTIONS	34
FIGURE 8 HASH VALUE OF STANDARD HASH FUNCTIONS	34
FIGURE 9 SCHEMATIC ILLUSTRATION OF HASHING DIFFERENT DATA INDEPENDENTLY.....	36
FIGURE 10 CALCULATING HASH VALUES REPEATEDLY	36
FIGURE 11 COMBINING DATA AND SUBSEQUENTLY CALCULATING THE HASH VALUE.....	37
FIGURE 12 CALCULATING HASH VALUES SEQUENTIAL	38
FIGURE 13 CALCULATING HASH VALUES HIERARCHICALLY.....	39
FIGURE 14 DATA LINKED TOGETHER IN A CHAIN-LIKE FASHION	42
FIGURE 15 DATA LINKED TOGETHER IN A TREE-LIKE FASHION	42

FIGURE 16 ILLUSTRATION OF A HASH PUZZLE	44
FIGURE 17 ILLUSTRATION OF THE HASH FUNCTION AS A MATHEMATICAL FUNCTION	47
FIGURE 18 ARCHITECTURE OF HASHING ALGORITHM.....	47
FIGURE 19 ILLUSTRATES A COLLISION IN PASSWORD HASHES.....	50
FIGURE 20 PREIMAGE COLLISION	52
FIGURE 21 PREIMAGE ATTACK.....	53
FIGURE 22 SECOND PREIMAGE COLLISION.....	54
FIGURE 23 SECOND PREIMAGE ATTACK.....	55
FIGURE 24 ILLUSTRATION OF THE ESSENTIAL CRYPTOGRAPHIC PROCESS	59
FIGURE 25 SCHEMATIC ILLUSTRATION OF SYMMETRIC CRYPTOGRAPHY	59
FIGURE 26 SCHEMATIC ILLUSTRATION OF ASYMMETRIC CRYPTOGRAPHY	60
FIGURE 27 EXAMPLE OF HANDWRITTEN SIGNATURE USED IN CHEQUE	65
FIGURE 28 CREATION OF A DIGITALLY SIGNED MESSAGE.....	66
FIGURE 29 VERIFYING THE SIGNED MESSAGE.....	67
FIGURE 30 IDENTIFYING FRAUD IN A SIGNED MESSAGE.....	68
FIGURE 31 ILLUSTRATE MAN IN THE MIDDLE ATTACK.....	70
FIGURE 32 CRYPTOCURRENCY MINING PROCESSES[]	73
FIGURE 33 VICTIM'S WALLET BEFORE THE ATTACK.....	89
FIGURE 34 ATTACKERS WALLET BEFORE THE ATTACK.....	90
FIGURE 35 VIEW PUBLIC KEY.....	91
FIGURE 36 GENERATING MODULUS OF PUBLIC KEY	91
FIGURE 37 PROVIDES MODULUS TO CADO-NFS PROGRAM.....	92
FIGURE 38 CADO-NFS RESULTS FACTORED PRIME NUMBERS	92
FIGURE 39 SCRIPT TO GENERATE CONFIG FILES	93
FIGURE 40 VIEW CONFIG FILE.....	94
FIGURE 41 PRIVATE KEY IS GENERATED	95
FIGURE 42 APPLICATION INTERFACE TO CONDUCT A TRANSACTION	95
FIGURE 43 APPLICATION INTERFACE TO MINE TRANSACTIONS	96
FIGURE 44 ATTACKER'S WALLET AFTER THE ATTACK.....	97
FIGURE 45 VICTIM'S WALLET AFTER THE ATTACK.....	97
FIGURE 46 GENERATING FILES WITH IDENTICAL MD5 HASHES	100
FIGURE 47 MINT IS AWESOME.....	101
FIGURE 48 MINT IS NOT AWESOME	101
FIGURE 49 TERMINAL SHOWING IDENTICAL MD5 HASHES.....	102
FIGURE 50 IMAGE FROM SHATTERED.IO DEPICTING IDENTICAL HASHES FOR TWO PDFS.....	103
FIGURE 51 QUANTUM ARCHITECTURE.....	105

Abstract

Cryptocurrency has emerged as one of the widely used applications based on Blockchain. This report intends to provide an overview of Cryptocurrency, Blockchain, Cryptographic hashing, Public key encryption and Quantum computer subjects. The report will touch upon various technologies that play a significant role in the creation and security of Cryptocurrencies.

Blockchain and Cryptocurrency have captured the interest of many individuals and enterprises in the industry. Many countries and institutions are starting to understand the significance of Cryptocurrency and Blockchain in their business models, but how secure are these technologies? Bitcoin paper [1] has ushered Blockchain and Cryptocurrency into a new era, an era where individuals can manage their assets in a distributed trust model. The security of these systems heavily relies on cryptography. In this report, we will also analyze the security of cryptocurrency implementation of Blockchain and learn how cryptography is evolving to maintain the working of Cryptocurrency.

Introduction

What Is Blockchain?

Blockchain, also known as distributed ledger (a book in which a company or a bank records the money it has paid and received) technology, is a distributed ledger system that stores distinct transactions/operations in a chain of blocks without the requirement for a trusted third-party. Through a pair of public and private keys, Blockchain has shown to be irreversible, assisting with and responsibility for behaviour, as well as, to a lesser extent, (keeping private information private). Following the success of Bitcoin, Blockchain has gotten much attention. Attempts have been made to leverage Blockchain's main features for various applications and use cases.

Basics Of Blockchain

The underlying mechanism for cryptocurrencies like Bitcoin is blockchain technology. Bitcoin is the First Cryptocurrency that was introduced in 2009. The launch of Bitcoin has introduced a slew of cryptocurrencies with market capitalizations in the billions of dollars. Satoshi Nakamoto, a pseudonym for individual or group, first introduced Blockchain in 2008 and deployed it as the Bitcoin infrastructure in 2009. Blockchain is a sequential chain of blocks, each of which can be thought of as a page in a ledger. The chain continues to grow as miners (who generate cryptocurrencies using various methods) find new blocks to add to the existing Blockchain. Each transaction is broadcast throughout the network using encrypted communication, and miners attempt to collect as many transactions as possible, validate them using "proof-of-work," and generate a new block. To make such blocks, miners would compete with one another. A fresh copy of the winning Block is broadcast to the whole network once it is attached to the Blockchain, resulting in a decentralized public ledger. As represented in Figure 1, traditional ledger technology requires a trusted third party, such as a bank. However, Blockchain-based technology operates on a peer-to-peer network, as shown in Figure 2, eliminating the requirement for a centralized trusted third party to manage transactions. This system does not require an operator, a centrally trusty third party, because the miner's agreement handles concerns such as double-spending. When someone changes a blockchain network and inserts a

special one to reclaim a coin, this is known as double-spending. Double-spending is possible, but it is more likely that a coin is taken from an unprotected and safe wallet. [2]

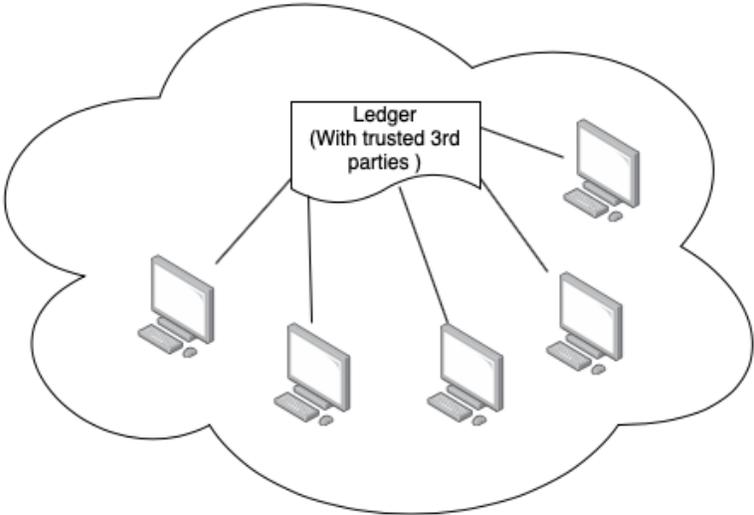


Figure 1 Conventional centralized ledger with trusted third-parties

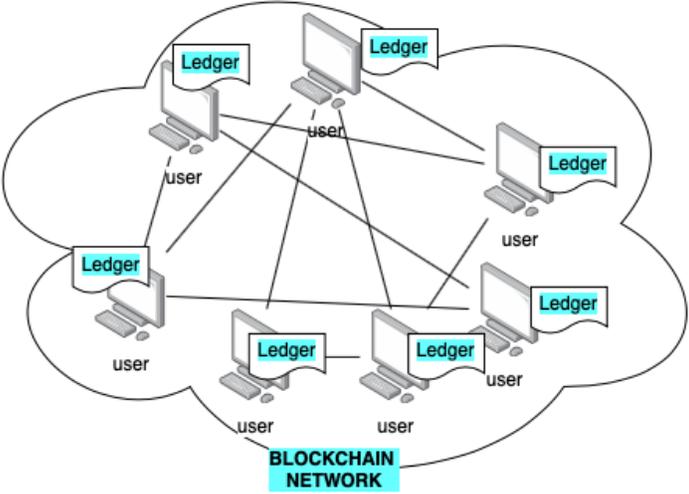


Figure 2 A typical example of Blockchain technology

Although Blockchain is most frequently associated with cryptocurrencies like Bitcoin, it can also be used for other purposes. As demonstrated in Figure 2, Blockchain technology provides

financial services without the involvement of financial institutions such as banks or other intermediaries. It can be used to undertake online payment, digital asset management, and remittance services. De-centralization, immutability, integrity, and anonymity are key characteristics of Blockchain technology, making it applicable to non-financial domains such as smart contracts, the Internet of Things, reputation systems, security services, wireless network virtualization, and other applications.

Modern Use Of Blockchain:

After the successful implementation of Blockchain in Bitcoin because of its most essential characteristics, Blockchain emerged to be used in various other applications and use cases, as shown in Figure 3. a brief overview of each domain is mentioned in the following section.

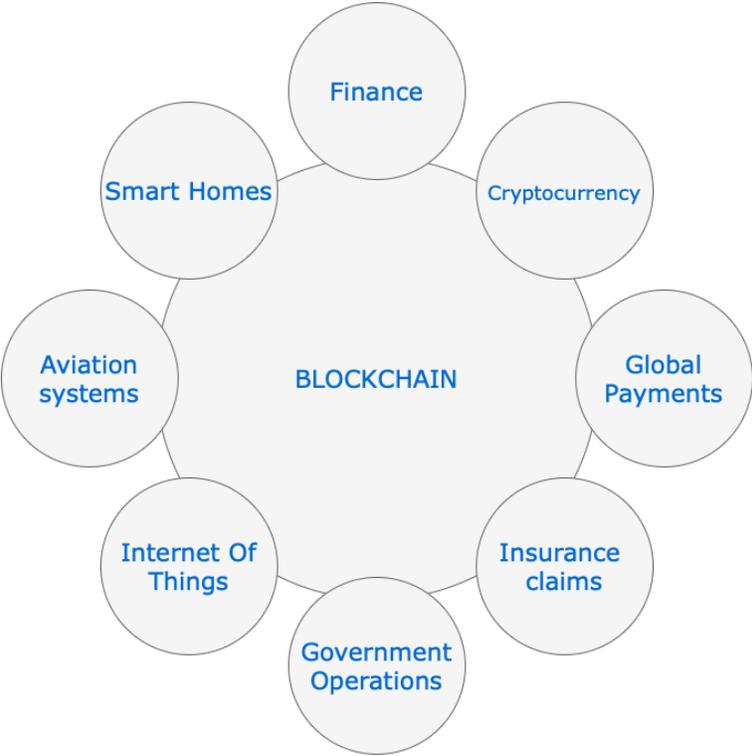


Figure 3 Different Blockchain applications and use cases

Let us explore some applications and use cases of Blockchain in different sectors: [3]

Finance

Financial transactions are verified and processed by an intermediary, such as a bank. Having such a centralized system places much work in the hands of intermediaries, and the transactions are prone to errors because several uncoordinated parties must keep track of and change the records. As a result, the entire procedure is time-consuming and expensive. With the creation of a distributed public ledger, the Blockchain reduces the complexities involved in the financial services by allowing miners to verify transactions. There is transparency about transactions because each node in the Blockchain network has a copy of the updated Blockchain. Because the blocks are structured chronologically, once a block with a validated transaction is added to the Blockchain, the entire Blockchain cannot change. As a result, once a transaction is registered in the system, attackers cannot modify it. Miners always choose the longest chain in the event of a conflicting Blockchain where branching may occur, as the longest chain is more reliable. It creates an excellent system to improve our existing financial services with a secure communication protocol and sturdy verification technique.

2. Cryptocurrency

Blockchain enabled the creation of Cryptocurrency, which now has a market capitalization in the billions of dollars. Bitcoin is based on cryptographic technology that allows the users to securely send and receive money without the necessity for a trusted third party, such as a bank or a corporation like PayPal. A new block is formed by executing a consensus method such as Proof-of-Work on the Bitcoin network, which is based on a Blockchain—a distributed transaction public ledger points out that getting someone's private key from their public key is practically impossible, which protects users from impersonation attacks. The Bitcoin client software uses a mathematical process to combine the recipient's public key and the sender's (i.e., your own) private key and the number of Bitcoins you want to pay/send to complete a transaction. The transaction is then sent out to the Bitcoin network's dispersed clients for verification by Bitcoin software other than the sender and recipient; all Bitcoin users who are online check whether a valid owner sent the money by utilizing the mathematical relationship between the public and

private keys; and the public transaction log stored on each Bitcoin user's computer to ensure that the sender has the Bitcoins to spend.

3. Global Payments (traditional Currencies)

Because there are so many middlemen engaged in verifying transactions, global payments become complicated and time-consuming. The entire procedure can be costly and prone to errors. These challenges arise primarily as a result of the centralization of monetary transactions, in which entities such as banks and other financial firms prescribe processes and are accountable for transaction verification. By establishing a decentralized public ledger and a rigorous verification technique to verify transactions, Blockchain technology lowers the complexities involved in Global payments. As a result, Global payments are faster, verifiable, immutable, and secure in this peer-to-peer network. Several transfer businesses are already employing Blockchain technology for remittance services.

4. Insurance Claims and Processing

Several fraudulent claims have been filed with the insurance company. Furthermore, in order to correctly process an insurance claim, which is difficult to handle using traditional methods, updated policies and data must be attached with each claim. The process may be handled swiftly and securely using Blockchain (distributed ledger technology). Similarly, because several participants/miners must agree on the legality of each transaction, any false claims/transactions may be identified and dropped with high assurance. This ensures that the insurers satisfy their claim in a timely and efficient manner.

5. Block chain

Government Different government organizations and units can use Blockchain technology to develop trustworthy and successful government operations through collaborative and transparent networks. The key qualities of blockchain technology will aid accountability, transparency, and confidence among stakeholders such as citizens, leaders, and government officials, as well as their various activities. To address the accountability of its bodies, the government must make its operations open. To do so, the government may have to make a large amount of data available to

the public. According to a McKinsey analysis, open data made publicly available on the Internet can benefit people in the billions of dollars. Open data can be used by a variety of organizations to expose illegal activities.

6. Internet of Things (IoT)

Every year, the number of electronic devices connected to the Internet grows dramatically. The Internet-of-Things is created when a large number of devices are connected to one another (IoT). The Internet of Things is projected to improve people's lives, making notions like smart houses a reality. While having a large number of heterogeneous devices connected to the Internet is expected to make life easier, it also raises serious concerns about cyber security and privacy. Blockchain has the potential to be a significant technology for securing IoT. With millions of devices connected and communicating, it's critical to ensure that the data flowing through IoT is secure and that participants are held accountable.

7. Blockchain in Aviation Systems

In the aviation business, Blockchain can enable strong collaborative collaborations between service and product suppliers to supply travel services and products in a distributed and secure manner. Smart contracts have the potential to simplify interactions between organizations and within businesses.

8. Supply Chain Systems/Sensors

Smart sensors can assist businesses in gathering information about their supply chains as they travel throughout the world. Smart sensors are said to be used by several major supply chain organizations to track commodities. As a result, the number of these sensors is predicted to increase in the near future quickly. With such a large number of sensors, there will be a vast amount of data to collect and evaluate. Blockchain technology can disrupt supply chains and networks, making them more efficient and secure.

9. Smart Homes

In the perspective of smart homes with IoT devices, Blockchain aid in ensuring secure and dependable operations. However, Blockchain requires high resources for proof-of-work, considerable storage capacity, fast data, and high scalability. Implementing Blockchain in resource-restricted IoT devices is a challenge in itself.

10. Cars and Phones

Authentication keys are primarily used to protect personal devices like phones and cars, therefore, they are exclusively available to their owners via smart keys. Although cryptography permits this type of technology, such systems can fail if the authentication key is stolen, copied, or transferred. Users can replace and recreate lost credentials on the Blockchain ledger, which can resolve such situations.

What Are Cryptocurrencies?

Cryptocurrency is one of the most prominent use cases of Blockchain technologies. It is an Internet-based exchange medium to conduct financial transactions. Cryptocurrencies utilize blockchain technology to achieve decentralization, transparency, and immutability. The main advantage of Cryptocurrency is that a single central authority does not control it. The history of all the transactions is openly available for anybody to see and verify, making it highly transparent. Blockchain technology is immutable, meaning the blocks once added to the chain cannot be easily changed, making Cryptocurrency immutable. Using cryptocurrency technology, funds can be directly exchanged between two parties using private and public keys. The transactions can be done using minimal processing fees, saving users from high transfer fees charged by traditional financial institutions. [4]

The term cryptocurrency is coined from the cryptographic processes that make Cryptocurrency safe against frauds. The fraud of spending the same currency more than once is called double-spending. The traditional fiat currency is not susceptible to double spending as it is either used as physical entities or controlled by trusted centralized authorities. The digital currency is, however, vulnerable to double-spending. This problem is addressed by cryptography and blockchain technologies, making it nearly impossible to counterfeit or double-spend[5]. The word 'Crypto' in Cryptocurrency refers to the cryptographic techniques that make it secure; some are encryption, hashing, digital signatures. These will be further explored in the following chapters of this report.

Problems of Cryptocurrencies (Bitcoin)

In this section, we will discuss what problems are associated with the usage of Cryptocurrency. Since Bitcoin is the most widely used Cryptocurrency, we can view the problems linked to the usage of Bitcoin. The bitcoin is not regarded as money, at least not legally. Contrary to many peoples' beliefs, Bitcoin is not considered a legal tender. To be a legal tender, the money serves three functions – a medium of exchange, a unit of account and a store of value. [6]

Bitcoin or other cryptocurrencies are not universally accepted as a mode of payment, so it fails to be a universal medium of exchange. The Bitcoin itself is priced in USD or other fiat currencies, and to some individuals, it can be seen as another item (that can be purchased). Cryptocurrencies do not have a stable value; their value drastically varies within hours. For anything to be used as a store of value, its value must be somewhat stable as well as should be universally accepted.

The process of verifying the transactions is called mining and offers a reward in return. This reward is nothing but newly created tokens of Cryptocurrency itself. The bitcoin limits the total quantity in circulation, which is 21 million bitcoins. At writing this report, Bitcoin has reached 90% of its circulation limit, i.e., 19 million. This supply limitation makes cryptocurrencies unsuitable as legal tender because the static 'money supply' will prevent the central authority from conducting countercyclical policy. Additionally, the majority of the world's central bank's policies do not have the provision of issuing digital currencies. So, using cryptocurrencies as a legal tender will require changes to the existing laws.

Key Components Of Cryptocurrencies

Blockchain

A blockchain is a decentralized database that is shared between nodes on a computer network. Like any database, the Blockchain also stores information electronically in a digital format. Blockchain is one of the key components for cryptocurrencies such as bitcoin for securely maintaining the record of transactions. A blockchain stores information by collecting it in groups called Blocks. Each Block has a predefined size limit; it is linked to the previously filled Block when filled. This linking creates a chain among all the blocks and is called Blockchain.

The Blockchain guarantees the reliability and security of records, and it generates trust without the requirement of a trusted third party. [7]

Public ledgers

A ledger is a book or collection of accounts in which accounts' transactions are recorded. Every account has an opening balance and carries a forward balance. A record of each credit or debit transaction is entered in separate columns along with the closing balance. [8] It is very similar to the bank passbook that your bank issues to you. Since it is issued by your bank to you, only you can view it, and only your bank can update it. A public ledger is also like a bank ledger, but instead of a single authority, it can be by more than one authority and is openly available to view by anyone. The Blockchain is used as a public ledger for cryptocurrencies.

Wallets

A cryptocurrency wallet stores cryptographic keys. Cryptocurrencies use a pair of cryptographic keys called private and public keys; these keys are used to send and receive Cryptocurrency. Both these keys are linked to the owner via this wallet; the public key is the address to send currency to the wallet, while the private key provides the authorization to send currency from the wallet. [9]

Transactions

A transaction is an agreement between a seller and a buyer in order to exchange currency in return for goods or services. Cryptocurrency is also a mode of payment; the same principle of a transaction also applies to it. A transaction in Cryptocurrency is the record of the transfer of cryptographic funds among its users. These transactions are stored on a public ledger called Blockchain. The transactions on Blockchain are irreversible, meaning if a transaction on a blockchain is confirmed, it cannot be revoked at any point later in time. Transactions in bitcoin follow a specific procedure when a user sends funds to another user. The remainder of funds are sent back to the user as part of the transaction, e.g., if Bob has 10 coins and sends 2 coins to Alice, the bitcoin system will deduct 10 coins. It will send 2 coins to Alice and the remaining 8 coins back to Bob. This is all recorded as part of the transaction. [10]

Peer-to-Peer network

A peer-to-peer(P2P) network is formed when two or more computers are connected and share information or resources without relying upon a separate server computer. The P2P network enables the Blockchain to operate without any central server. The P2P is maintained by a

distributed network of computers called nodes. Each node runs the bitcoin software and contains a copy of Blockchain. Anybody can participate in the P2P network of bitcoin. [11]

Mining

Mining refers to the process of adding blocks to the Blockchain after verifying and validating the transactions in those blocks. The nodes that participate in the blockchain P2P network and contribute to the process of validating and verifying the transactions are called miners. The mining involves a process of solving a computational puzzle and submitting the result to the network as a reward Miner is awarded cryptocurrency funds themselves.

Consensus algorithm

Since there is no central authority controlling the blockchain network, a protocol is required to add verified blocks to the Blockchain. The consensus algorithm is that protocol through which all the nodes of the blockchain network come to a common agreement about the current state of the public ledger. A consensus algorithm is required to achieve reliability in the network and establish trust between unknown peers. The Consensus Algorithm ensures that the new Block added to the Blockchain is verified and agreed upon by all the nodes in the network. [12]

Here is a list of various consensus algorithms used in cryptocurrencies:

- Proof of Work
- Proof of Stake
- Delegated Proof of Stake
- Byzantine Fault Tolerance
- Proof of Burn
- Proof of Authority

Proof of work

The proof of work is the popular consensus algorithm used in Bitcoin. Under this algorithm, miners compete with each other to solve a complex mathematical computation. The miner who first solves this puzzle gets the reward. The puzzle is based on cryptographic hashing (will be discussed in later chapters), which is hard to solve but very easy to verify. When the miner solves the puzzle and submits the solution to the blockchain network, other miners receive the update and verify the solution submitted by the miner. Miners have to contribute a significant amount of computational power to solve these hashing puzzles, which makes this proof of work an expensive consensus algorithm. It is due to proof of work mechanism the cryptocurrency transactions are immutable as changing the previous blocks in the Blockchain require exponentially large computational power. [13]

The 'Crypto' in Cryptocurrencies

Cryptocurrency is formed by words crypto and currency, from which currency means a medium for the exchange of goods and services. The 'Crypto' in Cryptocurrency comes from cryptography, meaning 'secret- writing'[14]. Cryptography is used as means to exchange messages in such a way that only the intended recipient can read them. Cryptography ensures the security of Cryptocurrency transactions and the involved participants. Cryptography also helps to achieve decentralization as well as protection from issues such as double-spending.

Cryptography is used for multiple purposes-

- To provide security to the transactions on the network
- To control the generation of new currency units
- To verify the transfer of ownership of funds

Public key cryptography

Public key cryptography is based on Asymmetric Encryption Cryptography, which uses two separate keys – public and private keys.[15] These keys are used to encrypt and decrypt data, encryption refers to converting the plaintext into a secret text called ciphertext, and decryption

refers to converting back the ciphertext to plain text. The public key can be shared publicly, just as an address of the fund receiver, whereas the private key is known to the owner only, just as an ATM pin. By this method, a person can encrypt a message before sending it using the receiver's public key, but it can only be decrypted by the owner using its private key. The message remains unreadable in transit.

Cryptographic Hashing

Cryptographic Hashing is the process of converting any data into a fixed-length unique string of text. Hashing is based on hashing function that always returns a fixed-length text no matter the type or size of input data. In cryptocurrencies or Blockchain, hashing is used to link the blocks of Blockchain together. The hash of a block containing the transaction is calculated, and its reference is embedded into the next Block. This reference to the previous blocks creates a chain; hence a so-called blockchain is created. Due to this linkage, it is nearly impossible to change or tamper in the blocks. If any tries to change anything within a block, he must change all blocks from there forward to apply that change which becomes enormously expensive. [16]

Digital Signatures

Digital signatures can be seen as the digital equivalent of handwritten signatures. Just as handwritten signatures are signed to show an agreement and/or to authorize a transaction, digital signatures are also used in the same way to show agreement and authorize the transfer of cryptocurrency funds. Digital signatures are the extension of public-key cryptography discussed above; they are used for two purposes in Cryptocurrency. Firstly, to authorize the transfer of funds from your wallet and secondly to verify that nobody else has tempered the transaction since you signed it. [17]

Blockchain Technology

Blockchain Technology Definition

Blockchain is essentially a database used to store information digitally. However, unlike other databases that require a central authority to manage and update, the Blockchain is updated and managed by the participants of the distributed network. The primary purpose of the Blockchain is to provide a structure for sharing information on the distributed network and preventing changes to that information. In 1991, Stuart Haber and W. Scott Stornetta first outlined the idea of Blockchain in their paper “How to timestamp a digital document.” This paper wanted to implement a system that would prevent tampering with document timestamps. However, this system was not implemented until 2009, when the Bitcoin project was launched. [18] [19]

Decentralized and centralized network

The network consists of two or more computers connected to send and receive communication. Each computer in the network is called a node. One node acts as a server in a centralized network, while other nodes act as clients. All the data is stored on the server node, so whenever the client node needs to access or update the data, it will request the server node. This type of network always has a central authority within the server node to authorize the changes in the data. On the other hand, in a decentralized network, each node can act as both a client and a server. This implies that every node in a decentralized network is independent and possesses the identical copy of data as other nodes of the network. This decentralized network forms a decentralized database, which does not require a trusted central authority for updating the database. The traditional financial systems use a centralized network architecture, whereas Cryptocurrencies are built on the decentralized network of Blockchain. [20]

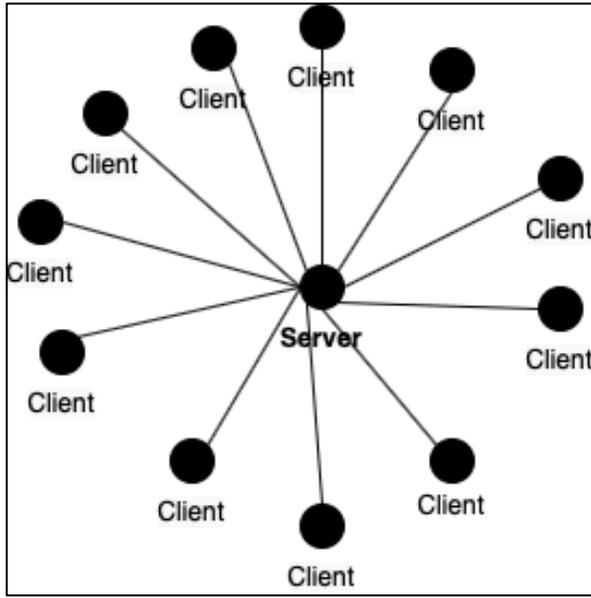


Figure 4 Centralized Network

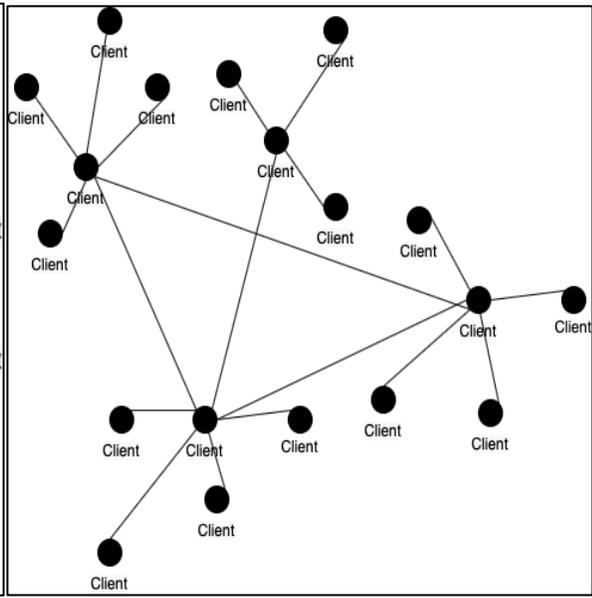


Figure 5 Decentralized network

Blockchain Technology Architecture

The Architecture of Blockchain technology consists of many layers. The number of layers is also dependent on its use case as well as each application has different requirements. To relate it more closely with cryptocurrency, we can divide the Blockchain into four core layers: Application, Consensus, Network, and Data. To discuss the Blockchain Architecture on the application of cryptocurrency, we can focus on how the Bitcoin framework classifies on these four layers.

Application Layer

This is the topmost level layer of blockchain technology architecture, and as the name suggests, it relates to the application that can interact directly with the end-users.

Consensus Layer

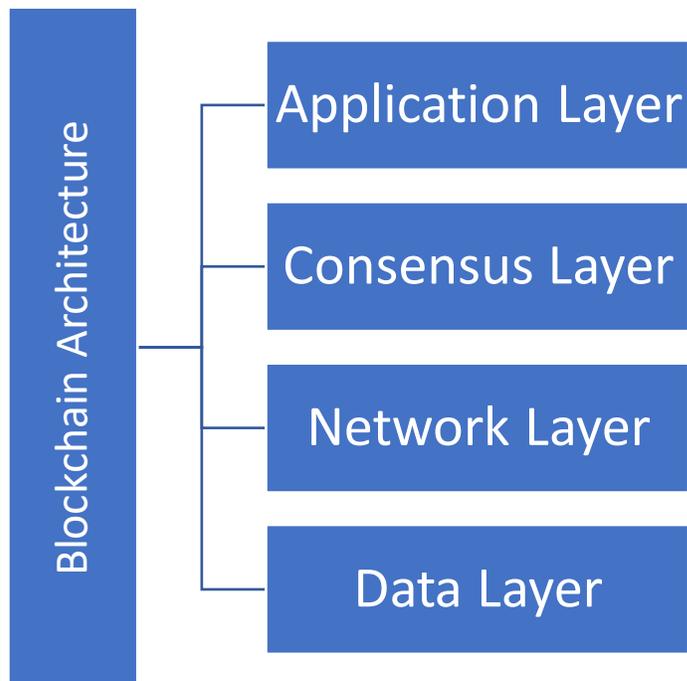
This layer constitutes various consensus algorithms used in the Blockchain to make decisions. Most cryptocurrencies use the proof of work consensus Algorithm explained later in this report.

Network Layer

This layer is responsible for communication and synchronization among the peer-to-peer network nodes. This layer ensures that only valid transactions are shared on the network.

Data Layer

This layer is responsible for arranging the blocks in a decentralized network. The blocks contain various transactions comprised of sender and recipients account information.



Blockchain Technology Features

Blockchain technology consists of different technologies such as algorithms and cryptography, mathematics and P2P networks. The consensus algorithm is at the heart of this technology. This is because the active nodes rely on a consensus algorithm for making decisions.

Blockchain technologies are composed of six key features:

Decentralization

The traditional financial systems require centralized institutions such as central banks to accept and monitor transactions. However, Blockchain eliminates the need for dependence on these centralized institutions. A consensus algorithm maintains the credibility of transactions.

Immutable

It is very hard or nearly impossible to change data after it is added to the Blockchain. The whole network has a copy of the digital ledger, and every transaction is verified against it before adding to the Blockchain.

Transparency

Because of decentralization, every participant can view and verify the transactions. The history of all the transactions is publicly available, making Blockchain highly transparent.

Anonymity

Blockchain technology supports anonymity; participants of the network are not required to provide identification credentials to contribute or even transact on the network. One only needs the recipients' payment address to send the cryptocurrency funds.

Distributed ledgers

This is an important feature that allows participants to verify ownership. Whenever a participant adds a block to the Blockchain, other network participants will have to verify and approve it, allowing fair participation.

Consensus

Consensus is one of the essential features of the Blockchain that provides trustlessness in the network. The consensus is achieved through an algorithm, and it is a kind of a voting system where the majority makes a decision.

Blockchain Storage Structure

All the valid transactions in the Blockchain are collected and stored in the groups called blocks. A block consists of several transactions along with their proof of work and the hash of the previous block. The average size of the bitcoin block is approximately 1 MB and can contain more than 500 transactions. Every node of the network validates each transaction before adding to the block, and once the block is filled, it is chained to the Blockchain.

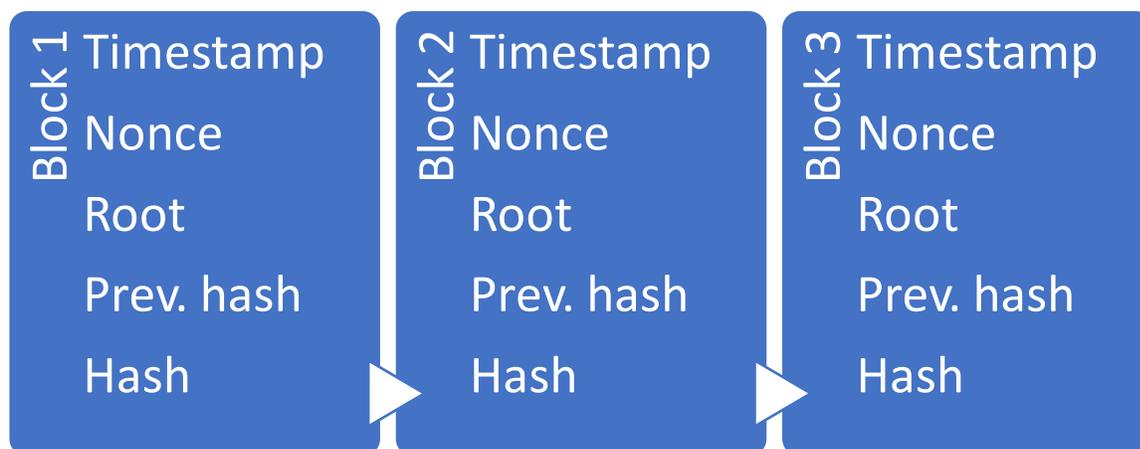


Figure 6 Blockchain Structure

A Block is formed by two components, a block header and the body. The block's body contains the list of all transactions associated with that block. The header of the block is again divided into five components. 1) *Nonce* stands for 'Number only used once'; it is the magic number that solves the hashing puzzle (discussed later). 2) *Root of the Merkle tree*, all the transactions in the body part are combined hierarchically to get a combined hash. 3) *Previous Hash*, the immediate predecessor block's hash, is added to create a cryptographically secure linkage. This linkage is why blockchain

data is very to edit after being added to the chain. 4) *Time in seconds*, the current timestamp at which the block was mined. 5) *Difficulty level*, the block's hash needs to meet the specific requirement; the difficulty refers to the number of leading zeros required in the hash.

Types of Blockchain

There are at least four types of Blockchain [21]

Public Blockchain

A public blockchain has no restrictions to access the Blockchain from the open Internet. Anyone on the Internet can send transactions and even become a validator on the network. A public blockchain network is also called Permissionless. The Bitcoin and Ethereum blockchain are the two largest known public blockchain networks.

Private blockchains

Unlike public Blockchain networks, private blockchain networks have restricted access and are called Permissioned. The participant can only contribute to the private network if the network administrators invite them.

Hybrid blockchains

The Hybrid blockchain network is a combination of private and public blockchains. The exact features of the Hybrid Blockchain depend on its usage and application. Some hybrid blockchains are open to join as a validator but restricted for open transactions.

Sidechains

The Sidechain is a designated chain for primary Blockchain that runs in parallel to it. The funds or tokens from the primary chain can be transferred to the Sidechain and used independently and vice-e-versa. The Sidechain Blockchain is itself a fully functional Blockchain.

Blockchain Technology Security

Blockchain is famous for its great privacy and security features, Blockchain has become a trendy topic in recent years. Security is a topmost priority and consideration in today's world, as the number of people and businesses who use technology is rapidly expanding. When data needs to be stored & distributed securely and efficiently, blockchain technology plays an essential role. They have private data passed back and forth between systems and applications. To secure this information from numerous security concerns, precautions must be taken. The blockchain consensus algorithm is a system that allows all network nodes to manage and agree on a single source of truth; a Suggested proof-of-work system allows network participants to coordinate equally.[22] Furthermore, the operation of Blockchain enforces network members to act honestly talk about how expensive it is to carry out fraudulent transactions just on the bitcoin network.[23] Many Miners are hungry to hack the system and might do so if they could pool their computer power to equal or exceed the aggregate processing power of the rest of the network; however, this would require hundreds of millions of dollars. Even though various experts have demonstrated and documented how difficult it is to destroy a blockchain, users are still apprehensive about its security. Blockchain is still vulnerable to security flaws and attacks. Since the dawn of the Internet, blockchain systems have used a wide range of technology, including computers and electronic databases, and security difficulties have plagued all these technologies.

Blockchain Security Issues

The blockchain architecture is composed of many layers. This section, however, will focus on four layers, as indicated in the previous section. The first is the application layer, the second is the consensus layer, network layer, and data layer are the four layers. Data blocks, chain structure, timestamp, hash functions and digital signature comprise the data layer.[24] In comparison to other layers, the network and data layers are more vulnerable to security attacks. Time hacking attacks, selfishness mining, 51% attack, and double-spending attacks occur on the data layer. Other layers, such as the application layer, may be vulnerable to other attacks, such as threats on the wallet software. Chen et al. (2019)[25] describe financial losses caused by a DDOS attack that targets the network layer to disrupt the system by introducing malicious attacks that consume most of the system's processing capacity. In 2016, a DAO attack on Ethereum resulted in the theft of around

US\$60 million. Despite the potential advantages that this Blockchain provides society and how it is transforming people's lives in various ways, it still has to be improved in terms of security.

Blockchain Security Vulnerabilities

Although blockchain technology is just a recent innovation that promises secure computing in a distributed system without centralized authority, its layered architecture is vulnerable to security threats. Mauro Conti[26] and colleagues describe the security flaws that emerged while implementing the bitcoin system. Chen et al.^{iv} (2019) investigated security flaws and attacks on Ethereum blockchain systems, applying to any blockchain system. Li et al.(2020)[27] reviewed an actual attack on major blockchain systems from 2009 to May 2017 and examined the vulnerabilities. Both bitcoin and other cryptocurrencies have vulnerabilities that arise within the blockchain operating mechanism. Furthermore, several risks are associated with the Ethereum blockchain, which arises from the creation, implementation, and execution of smart contracts. Each layer of the Blockchain is linked to a specific set of security flaws.

Vulnerabilities in the application layer affect the user interface when engaging with the blockchain system using programs like Google Chrome. This covers risks such as when the visibility of a function is wrongly configured, allowing unauthorized access. As a result, there is insufficient validation and external reliance. Vulnerabilities in the data layer are linked to the database configuration error due to insufficient transaction information. 51% consensus layer Vulnerability is unavoidable when a group of miners pool their processing power to exceed 50%, allowing them to take control of the Blockchain—performing double-spending via reversing transactions by altering blocks. We will be discussing more double-spending and 51% attacks in the coming chapter. Network layer Insecure API design, improper configuration, Insufficient authentication and other common attacks that attack via internet infrastructure.

Vulnerabilities in Blockchain, in general, are caused by configuration errors during the design of the database implementation, which can lead to flaws that allow unauthorized access to data,

inadequate human functionality of applications, as well as internal attackers who attack the consensus algorithm mechanism by pooling their computing power to work against it.

Blockchain Security Issues per layered architecture

Blockchain technology has the potential to make a significant impact, particularly in the financial sector, by improving business operations while also addressing issues that have been identified in the traditional financial system, such as security concerns due to the cryptographic mechanism that underpins it. However, vulnerabilities may enter the system at several phases of blockchain implementation, such as the development stage, user interface, configuration error, and so on.

Each of the layers described in the preceding section must fulfil its role in the architecture for a blockchain application to work. Furthermore, each layer has some weaknesses that cause security difficulties in the system, resulting in financial losses for individuals or institutions who use it.

Application Layer

The application layer is the layer that allows the end-user to interact with the system. It enables the user to engage with the blockchain system. On this layer, apps like blockchain dApps (distributed applications) and smart contracts run. Users can use blockchain wallets to store, transfer, and manage bitcoin and ether. A user must have an account in the wallet in order for it to work. Clients in the blockchain system are unable to communicate with the API directly. As a result, an application layer functions similarly to a web browser, providing a user interface for users who are not developers. For clients to be able to access the service, a distributed application is provided. For example, a decentralized application (dApp) is a system terminal that serves as a user interface for clients to communicate with smart contracts. The security concerns in the system emerge from the user's engagement with the system. Each tier of the blockchain architecture is vulnerable to a different type of attack. There are, however, security dangers that can attack many layers. DDoS attacks, for example, are a type of distributed denial of service attack (DDoS). DDoS attacks can occur at the network layer or at the application layer.

Attacks against wallet software: In Blockchain, a wallet is required for a customer to make transactions. Software wallets are programs that users can download from the Internet and install on their computers, such as smartphones, desktops, and laptops. These wallets are also referred to as online wallets because they are used to store private keys locally.[28] Instead of being saved on a local computer, private keys are stored in the cloud. The most common reason for wallet software attacks is a vulnerable online wallet. The Blockchain's authentication method relies heavily on the private key. The Elliptic Curve Digital Signature Algorithm (ECDSA) is used by bitcoin to sign and validate transactions. There are several types of research showing ECDSA method generates insufficient signatures, resulting in the compromising of the private key. Also, Inadequate control over address creation: When sending or receiving bitcoin, payers can name a trusted party that attests to the payee's identity, and they can compel the payee to utilize a verified bitcoin address. This gives the attacker the ability to modify the payee's address to the attacker's address. Issues and malware: There are still bugs in the client software, such as in configuration, GUI design, security, and other areas. These flaws are used to compromise the blockchain technology. Weaknesses in the wallet due to flaws in key generation and configuration issues in the implementation of ECDSA lead to the exposure of private keys ^{vii}.

Criminal activity

Because bitcoin users can have multiple bitcoin addresses and the procedure is anonymous, it is difficult to determine who is doing what with the currency. One type of criminal conduct involving bitcoin is ransomware. WannaCry ransomware affected roughly 230,000 people in 150 countries in just two days in May 2017. It used a flaw in the Windows operating system to encrypt victim's files and then demand a Bitcoin ransom. ^{vi}

The DAO assault is a type of cyber-attack that targets Ethereum specifically. DAO is a smart contract that was released on Ethereum on May 28, 2016, and implements a crowd-funding platform. DAO is a smart contract that was released on Ethereum on May 28, 2016, and implements a crowd-funding platform. An attacker can make many calls to the smart contract using the intermediate state. By publishing a malicious smart contract with a withdraw () function call to DAO in its callback function, this attack takes use of the reentrancy issue. The attacker can then steal from DAO. ^{iv}

Data Layer

The content, data format, and operation of the blockchain data are all part of this layer. Blockchain is a decentralized technology that allows each node in the network to submit transactions to blocks. The sender, recipient, amount, and hash value are all necessary values that must be contained in a block. Because it involves network transactions, which are the main components of the blockchain system, this layer is vulnerable to security concerns.

Transaction privacy leakage

Normally, private keys are used to safeguard transactions so that an attacker cannot determine whether the funds in separate transactions are acquired by the same user. When initiating a transaction in Monero (a digital currency), however, users add chaff coins so the attacker wouldn't infer the actual coins being sent. However, Blockchain's privacy protection is still lacking. Because all transactions do not contain chaff coins, the attacker may be able to deduce the actual coins in the transaction, resulting in privacy leakage.^{vi}

Private Key Security

In the blockchain system, the private key is utilized for user identification and is regarded as a security credential. ECDSA (Elliptic Curve Digital Signing Algorithm), which is used in Blockchain to produce private keys for users, has a flaw in that it does not generate enough unpredictability, making it difficult to guess the signature process, resulting in privacy leakage.^{vi}

Consensus Layer

In the blockchain architecture, the consensus layer is the most important component. It's the layer in charge of enforcing the rules that network participants should follow in order to achieve a consensus on the transactions that have been published. Because there is no central authority to ensure the system's reliability and consistency, blockchain systems use a consensus process to do this. This layer is in charge of verifying and validating the blocks and ensuring that all network members agree on what is going on. Furthermore, it is concerned with the transaction's order. This layer is the most important in the system, and it is vulnerable to a variety of attacks that can result in significant security vulnerabilities.[29]

51% attack

A group of miners can pool their processing resources to achieve a hash rate of higher than 50%. Attackers can gain control of the entire Blockchain by modifying and reversing the transactions they initiated, allowing them to engage in double-spending behaviour. It is also referred to as the majority attack, and it is capable of preventing validation. As a result, transactional denial of service occurs. 51% attack is further discussed in a later chapter.

Pool hopping attack

The information about the quantity of contributed shares in the mining pool is used in this attack to do selfish mine. The member of the bitcoin network who declares the valid proof of work receives a share. In order to discover a new block, the attacker continuously analyses the number of shares provided by other miners. After conducting this analysis, the attacker can profit from the shares by switching to another pool.[30]

Fork problems

When it is time to upgrade to a new edition of blockchain software, this issue arises. This results in a new consensus rule agreement. The nodes that had the new rules from the new version could not communicate with the nodes still running the old software. As a result of the system's incompatibility, because new nodes' computing capacity is greater than old nodes, the block mined by the old nodes will never be accepted by the new nodes, resulting in unequal income.[31]

Selfish mining

This attack is carried out by a group of miners who collaborate to waste the computational power of honest miners. Dishonest miners aim to establish a long private chain while honest miners continue to mine on the public chain, which will not be broadcasted until the dishonest miners' fresh blocks are exposed. This gives the attacker first priority while mining the following block.^{vi}

Network Layer

Blockchain works on peer-to-peer network infrastructure. The data is shared across the nodes that make up the network. Data representation and network services planes are two components of the network layer. The data representation plane is concerned with data storage, encoding, and protection, whereas the network service plane is concerned with communication, routing, addressing, and naming services. Given the diverse underlying technologies that this layer

encounters, it is also very vulnerable to various security risks. Because the network layer facilitates communication between nodes in a peer-to-peer design, the majority of risks on this layer come from man-in-the-middle attacks.

Eclipse attack

The attacker deliberately seizes a node's connections to its peers in order to gain control over all traffic sent and also received by that node. As a result, the attacker has the potential to cause major security problems in the system, such as selfish mining and double-spending.^{viii}

Timejacking attack

When attaching to a node, the attacker publishes an incorrect time. When the node's time counter is modified, the susceptible node may accept a different blockchain. This could exacerbate the problem of double-spending (discussed later).^{vii}

Sybil attack

This attack creates bogus identities & assigns them to the peers of the target node. The attacker will then force the user to select blocks that are only under the attacker's control. ^{vii}

Balance attack

The attacker stutters connections between nodes that have the same amount of mining power. The main purpose of this attacker is to do double-spending issues in order to disrupt communication between these nodes.[32]

DNS attack

The attacker modifies the DNS data by poisoning the DNS cache. When a user requests the IP addresses of the peer's nodes from the server, the user is forwarded to the attacker's network. As a result, the attacker gets complete control over the target node.^{viii}

Distributed Denial of service

Because Blockchain is a peer-to-peer network, this attack seeks to restrict service between network nodes by injecting malicious traffic that consumes bandwidth and breaks connectivity, resulting in service denial.[33]

Hashing

Fingerprints are the impressions left by friction ridges of human fingers. There are very finely detailed and nearly unique, very difficult to alter and do not change over time; hence are very suitable to be used as long-term markers of human identity. A hash value is a concept of cryptography and can be considered a digital equivalent to human fingerprints. As every human can have a nearly unique fingerprint, every set of data can also have a nearly unique hash value.

Hash functions are small computer programs that convert any type of data to a string of fixed length numbers, regardless of the size of the input data. Hash functions accept only one piece of data as input and generate a hash value based on the bits and bytes that make up the data. Hash values can have leading zeros to provide the required length. Many different hash functions differ in the length of the hash value they generate. An important group of hash functions are called cryptographic hash functions, which generate digital fingerprints equivalent for each type of data. Cryptographic hash functions should have the following properties [34] [35]

- Providing hash values for any type of data quickly
- Being deterministic
- Being pseudorandom
- Being one-way functions
- Being collision-resistant

Providing hash values for any type of data quickly

This property is a combination of these two qualities. First, hash functions should compute hash values for all types of data, and the hash function should perform the calculations quickly. These features are essential because we do not want the hash function to return useless things like error messages, or it may take a long to return the result.

The hash functions should compute the hash value of data of significant sizes in significantly little time. Then only the application can be used to generate and compare the large volume of hashes for their trueness. [36]

Being deterministic

Deterministic means that the hash function should return the same hash value for the same input data. This means that any observed differences in the hash values are due solely to differences in the input data and not to internal causes of the hash function. The function should not be dependent on any operating system environments or implementation methods. When hashed with a specific hashing function, the same data should always return the same hash value without any exception. [37]

Being pseudorandom

Pseudo-random means that the hash value returned by the hash function should change unexpectedly when the input data changes. Even small changes to the input data will unexpectedly change the resulting hash value. In other words, the hash value of changed data should always come as a surprise. The hash value must not be predictable from the input data. This property can be associated with human twins. Even in a case where twins are very similar to each other, they are never the same, and hence they individually have distinct fingerprints. [38]

Below figures 8,9 depict the hash value of strings 'MINT' and 'MINTS' using various hash functions such as MD5, SHA1, SHA256, SHA512. If you look closely, the respective hash value of each hash function is bizarrely different even if we have only added a single character to the data string. This property is called pseudorandom.

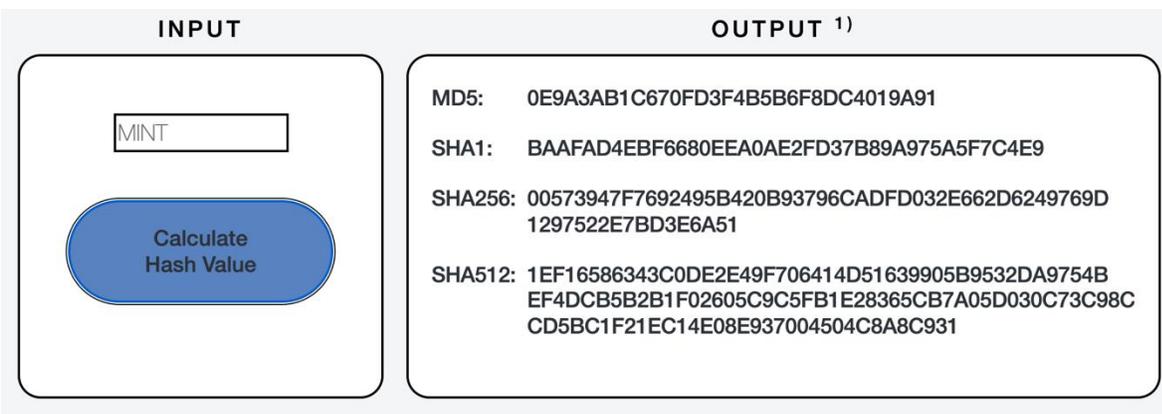


Figure 7 Hash value of standard hash functions

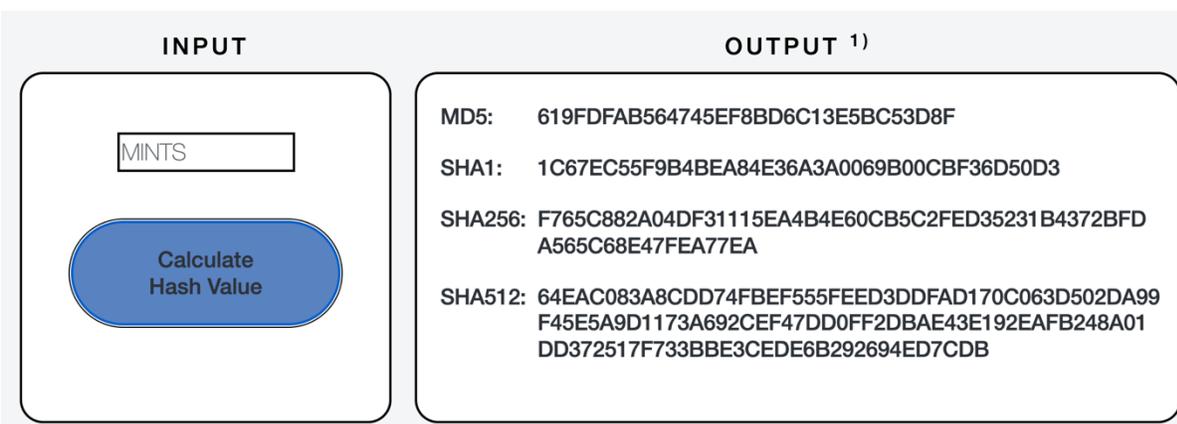


Figure 8 Hash value of standard hash functions

Being one-way functions

One-way functions do not provide a way to track input values through output. Since it is a one-way function, it cannot be used in reverse. In other words, it should be impossible to retrieve the original input data based on the hash value. In the same way, that an isolated fingerprint tells you nothing about the identity of the human, a hash value should tell you nothing about the content of the input data. One-way functionality is also said to be non-transferable.

Being collision-resistant

A hash function is called collision-resistant when it has very little probability to generate the same hash value of two or more distinct pieces of data. Alternatively, if there is a significant possibility of obtaining the same hash value for two or more data sets, the function is not collision-resistant. In this case, the Hash value generated by the Hash function cannot be unique and can create ambiguity when comparing data. If you receive the same Hash value for other data fragments, you will be dealing with the Hash conflict. Hash accident is like two people with the same fingerprint. To use hash values as digital fingerprints, collision protection is essential.

Patterns of Hashing Data

we have now learned that a piece of data can be used as input to a hash function. The hash function, in turn, provides a hash value of that data. This means that each independent piece of data has a unique cryptographic hash value. However, what will we do if we have to provide a hash value of multiple independent pieces of data? The hash function accepts only one piece of data at a time. No hash function processes independent datasets simultaneously, but in practice, many datasets often require hash values. The Blockchain deals with multiple transaction data at once, requiring a hash value.

we can utilize one of the following patterns in applying hash functions to data [39]

- Independent hashing
- Repeated hashing
- Combined hashing
- Sequential hashing
- Hierarchical hashing

Now we can elaborate on each of these techniques and observe how hashing can be performed when we are dealing with multiple independent datasets.

Independent Hashing

Independent hashing means applying a hash function to each piece of data individually.

Fig 3 illustrates this concept by calculating the shortened hash value of two unique words separately.

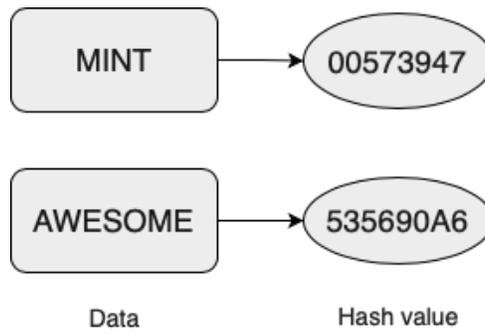


Figure 9 Schematic illustration of hashing different data independently

The white box and the gray circle represent the data and corresponding hash value. An arrow pointing to a circle in the box indicates a change of the data to hash value. As shown in Figure 10, different words have different hash values.

Repeated Hashing

We have learned that hash functions convert arbitrary data into hash values. The hash value itself can be seen as a piece of data. Therefore, it should also be possible to provide a hash value as input to a hash function and calculate the hash value again.



Figure 10 Calculating hash values repeatedly

Repeated hashing is a process of performing a hash operation on the hash value of its own result. Figure 11 illustrates the concept by iteratively computing a shortened hash value of text MINT. It gives a hash value of 00573947, and repeated hashing on this hash value yields the hash value of 535690A6.

Combined Hashing

The goal of combined hashing is to get a single hash value of more than one piece of data in a single attempt. One way to achieve this is to combine all the independent data into a single data item and then calculate the hash value. This is particularly useful if, at some point, we want to generate a single hash value for a set of available data. Consolidating data requires processing power, time, and storage space, so aggregate hashing should only be used when individual data items are small. Another disadvantage of combined hashing is that you cannot hash values for individual data since only the aggregated data is passed to the hash function.

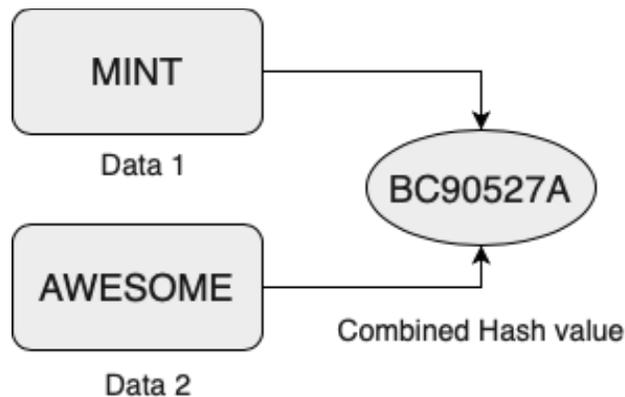


Figure 11 Combining data and subsequently calculating the hash value

Figure 12 depicts the process that the data in the individual white boxes are combined into a single data, usually by joining them with a space. So, 'MINT' and 'AWESOME' would be combined into 'MINT AWESOME', and the hash value of this combined value is BC90527A. The hash value is dependent on how the data is combined. Some other common ways to combine data are joining with plus sign(+) or hashtag(#) sign.

Sequential Hashing

The purpose of sequential hashes is to update the hash value incrementally as new data comes in. This is achieved by using combined and repeated hashes at the same time. Combine the old hash value with the new data and pass it to the hash function to get the updated hash value. Sequential hashes are especially useful if we want to keep hash values for long periods of time and update them when new data arrives. One of the advantages of this type of hash is that there is a hash value at any time, and its evolution can be traced as to the new data.

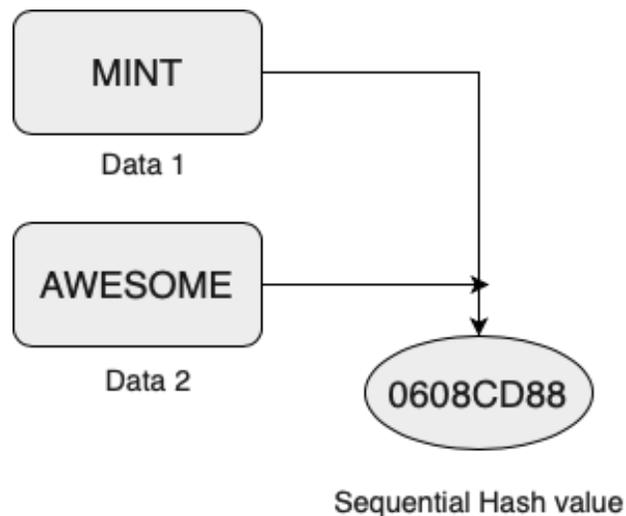


Figure 12 Calculating hash values sequential

Figure 13 illustrates the concept of sequential hashing, which begins by individually hashing the word "MINT" and creating the abbreviated hash value "00573947". When the new data word represented by "AWESOME" arrives, it is appended to the existing hash value and provided as input to the hash function. "AWESOME 00573947" is generated, and then its hash value is calculated as 0608CD88.

Hierarchical Hashing

Generating a hash combined with a pair of hash values forms a small hierarchy of hash values with a single value at the right. Similar to combined hashes, the idea of hierarchical hashes is to create a single hash value from a collection of data. Hierarchical hashes are more efficient because they combine hash values of a fixed size rather than the original arbitrary size data.

Moreover, hierarchical hashes only combine two hash values in each step, while combined hashes combine the amount of data provided in one attempt.

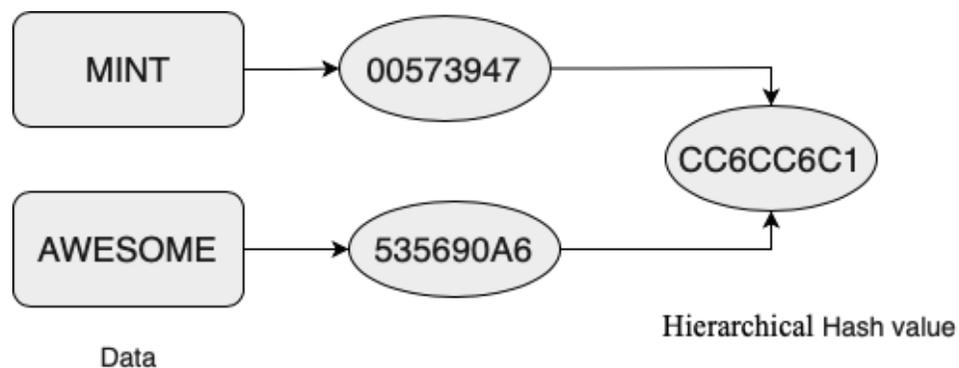


Figure 13 Calculating hash values hierarchically

Applications of Hashing

Now that we have a fair idea about the hashing functions and their properties. We can look into the real-world usage of hashing, and we can further explore how these properties of hashing play a crucial role in the development of blockchain systems and cryptocurrencies

Hash values have many applications and can be used:

- To compare two or more data
- To detect any change in the data
- To have a reference to data in a change-sensitive manner
- To detect changes in a collection of data
- To create computationally expensive jobs

To compare two or more data

The most obvious use case of hash values would be comparing the data based on their hash value. Instead of comparing two files piece by piece, we can simply compare their hash value. If the hash value matches, that means the files are identical. Moreover, since hashing is quick, it is often very little time to calculate the hash of data sets and compare hash values instead of

comparing both the data sets. Data comparison works because of the collision resistance property of cryptographic hash functions when comparing cryptographic hash values of two same data set will always be same and when comparing the cryptographic hash values of different data will be highly unlikely to be same. [40]

To detect any change in the data

This idea of comparing data can easily be extended to the case of detecting a change in the data. So the goal is to check if the data remains true and is not altered upon sending it to some other system or verifying it at a later point in time. We can achieve this by calculating the cryptographic hash value of the data set before storing or sending and then again calculating the cryptographic hash value using the same hash function. If these two hash values are identical, the data is accurate and is not changed over time or in transit. This idea works due to being a deterministic property of hash functions. Since Hash functions always produce the same hash of the same data set, and if the hash value of the same data set is different, that signifies that data has been tampered with and is not true.

To have a reference to data in a change-sensitive manner

The primary use case of hash values could be comparing data and detecting changes. More advanced use of hash values is hash references. Hash reference can be seen as a reference to the data stored somewhere else, such as a hard disk or database. This reference can next be used to determine if the data is unchanged. In order to achieve this, the cryptographic hash value of the data being stored is combined with the information about the place where the data are located. So, If the data changes, the hash reference becomes invalid because the two pieces of information don't match. The whole idea of a reference hash is to protect its users from retrieving data that has been accidentally altered due to technical errors or intentionally altered by someone else without telling you. Therefore, hash references are used where data should remain unchanged after being generated.

This idea of using cryptographic hash values as references to the data is very crucial in understanding the working of Blockchain as the hash value changes if any change is detected in the data and if the hash value is used as a reference somewhere else, it provides a clear indication that data is changed. Consequently, a broken hash reference is an evidence that the data has been changed since the creation of the hash reference.

For example, suppose you and your spouse share the same car. So, every time you return home, you take a picture of the odometer. Then again, before using the car, you take a picture of the odometer and compare both the picture. If you see a change in the odometer reading, you immediately know that somebody had driven the car since you parked. The hash reference works similarly. It captures the change in the data if the data is altered since you stored it earlier.

To detect changes in a collection of data

The idea of referring to data based on hash values can be further extended. A development of this idea is to store data in a change-sensitive manner. The goal is to store large datasets of information in such a way that they are change-sensitive to each other, and this change in data can be suspected quickly and easily.

There are two patterns of storing data in a change-sensitive manner using hash references:

- The Chain
- The tree

The Chain:

The Chain of linked data is also called a linked list. It is formed when each piece of data also includes the reference of the previous piece of data. Every time new data is added to the Chain, the reference of previous data is combined with new data before adding. This way very tightly coupled Chain is produced. [41]



Figure 14 Data linked together in a chain-like fashion

In figure 15, R1 represents the reference of Data 1 in 1st white block. R2 represents the reference of R1 and Data 2 combined in the 2nd white block. Similarly, R3 represents the reference of R2 and Data 3 combined in 3rd white block. The R3 reference is also known as the chain header because it refers to the most recently added data.

The Tree:

The Chain is very useful when new data is needed to be added while creating a link with previous data. However, a data structure Tree is more useful when we already have many distinct pieces of data that are available at the same time and to make them accessible via a single hash reference.

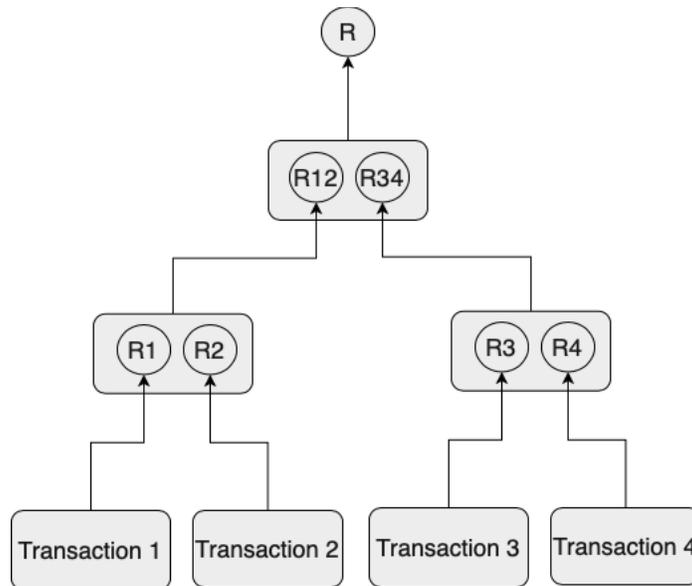


Figure 15 Data linked together in a tree-like fashion

This structure is also called the Merkle tree, named after a computer scientist. This structure looks like an inverted tree, with branches converging to a common root, called the root of the Merkle tree (R).[42]

The distinct hash values of two datasets named transaction 1 and transaction 2 are combined to get a new leaf (node R1 R2). In a similar fashion, node (R3 R4) is calculated from Transaction 3 and Transaction 4. The same process is followed until the Merkle root is converged from the datasets.

Both Chain and Tree can be considered as change-sensitive data structures. If the referenced data changes after the references are created, those references will be broken. Therefore, observing a broken reference in such a structure is evidence that some data has changed since the structure was created. Otherwise, we can conclude that the overall structure has not changed since its creation.

To create computationally expensive tasks

Hash values are used as change-sensitive data structures, but they have more value to provide in Blockchain systems. We can also use hash values to challenge the computer to other computers with complex puzzles. This usage of hash values is one of the essential concepts of the Blockchain.

We can use the analogy of combinations lock to explain this concept. A combination lock is a special lock that requires a unique sequence of numbers as the unlock pattern. If someone does not know the unlock pattern, they may regularly try all the possible combinations to guess the unique unlock pattern finally. This approach is known as brute force, it works, but it is time-consuming. Opening the combination lock using brute force requires no sheer intelligence but time and hard work. So, similarly, Hash puzzles are computer puzzles that can be thought of as a digital version that solves a combination lock by trial and error.

Hash Puzzle has 4 elements:[43]

- Data
- Nonce

- Hash function
- Difficulty

Blockchain and relationships among these elements:

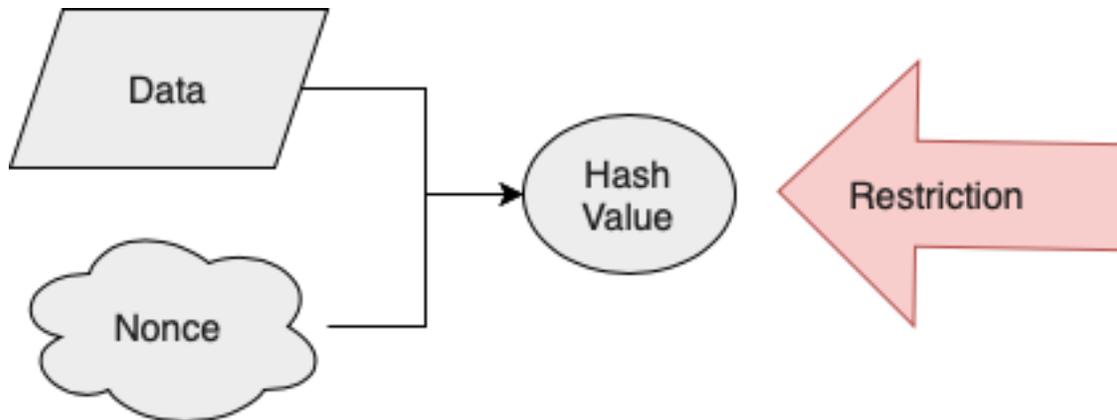
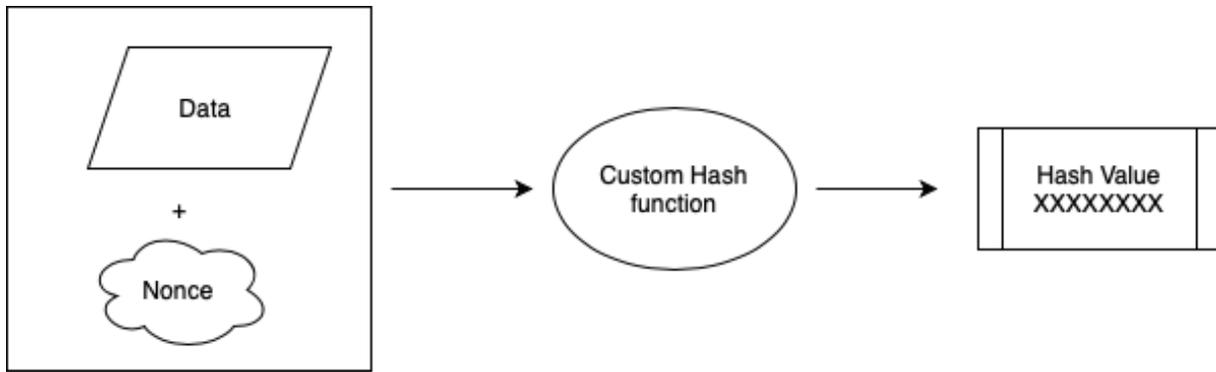


Figure 16 illustration of a hash puzzle

A hash function is always deterministic means it will always provide the same hash value for the same *data* irrespective of any other factors. Now suppose if we want to have a specific pattern to the hash value, we must alter the *data* so that the altered data could produce a different hash that matches the said pattern. This alternation of data is done by combining the data with other data. This other data is called *Nonce*. The condition that hash value meets a specific pattern can be seen as *Restrictions*.

An Illustrative Example

Consider we have a custom hash function that takes the text and returns a fixed-length hexadecimal number as the hash value.



We can use ‘Hello world’ as data and Nonce as whole numbers in this example. We can restrict the hash puzzle to have exactly 3 leading zeros; this restriction is also called Difficulty level. So, first, we combine data with Nonce and pass this hashed data as value to a hash function. This hash function returns fixed-length hexadecimal as a hash value. If the hash value does not fulfil the restriction of leading with exactly 3 zeros, the value of Nonce is incremented and again passed to the hash function. Observe in table 1 how each corresponding hash value is unexpectedly different from the others because of being pseudorandom property. As soon as the hash value of hashed data fulfils the restriction, the hash puzzle is deemed solved. In this case, combined Data with Nonce as 614 produces a hash value of 00068A3C, which has exactly three zeros. This process of finding the solution of the hash puzzle requires computation power hence costs energy.

Table 1 Nonces for Solving Hash Puzzles

Data	Nonce	Hashed Data	Hash Value
Hello world	0	Hello world 0	4EE4B774
Hello world	1	Hello world 1	3345B9A3
Hello world	2	Hello world 3	02307D5F
...
Hello world	614	Hello world 614	00068A3C

The Difficulty level

The requirement of fulfilling the restriction is the core property of the Hash puzzle. This restriction used in the Hash puzzle is consistent to challenge other computers with hash puzzles. This restriction is also referred to as the difficulty or difficulty level of the hash puzzle. The difficulty level is represented as natural numbers and refers to the number of leading zeros to the calculated hash value. Therefore, difficulty 1 means the hash value should have at least 1 zero in front. The more the leading zeros, the higher the difficulty, and as a result, higher computational power is needed to solve it

Hashing relation to Blockchain

Since hash functions are one way only, it is impossible to solve the hash puzzle by reverse engineering and hash value that fulfils the restriction. The only way to solve these hash problems is to try the trial and errors method, which requires a lot of computational power and energy. Also, hash functions are very deterministic and fast, so once the solution is found, it is straightforward to verify it.

In the blockchain context, the Hash puzzle refers to the amount of work someone has done to find the solution and is called *proof-of-work*.

Within Blockchain, hashing is also used

- To store transactions in a change-sensitive manner.
- As a digital fingerprint of transactional data.
- To have a proof-of-work process.

Design of Hash Function

At its core, the hash function is a mathematical function that works on two fixed-size data blocks to produce a hash code. This mathematical function is an integral part of the hashing algorithm.

The whole input is divided into multiple message blocks. The size of each block can vary from algorithm to algorithm, but its typical size is 128 bits to 512 bits.



Figure 17 Illustration of the hash function as a mathematical function

The whole hashing algorithm can have multiple iterations of the above hash function as a block cipher. Each iteration uses the previous iteration's output known as a seed value, and different message blocks produce the following hash code.

This process is continuously repeated until the hash code consumes all message blocks.

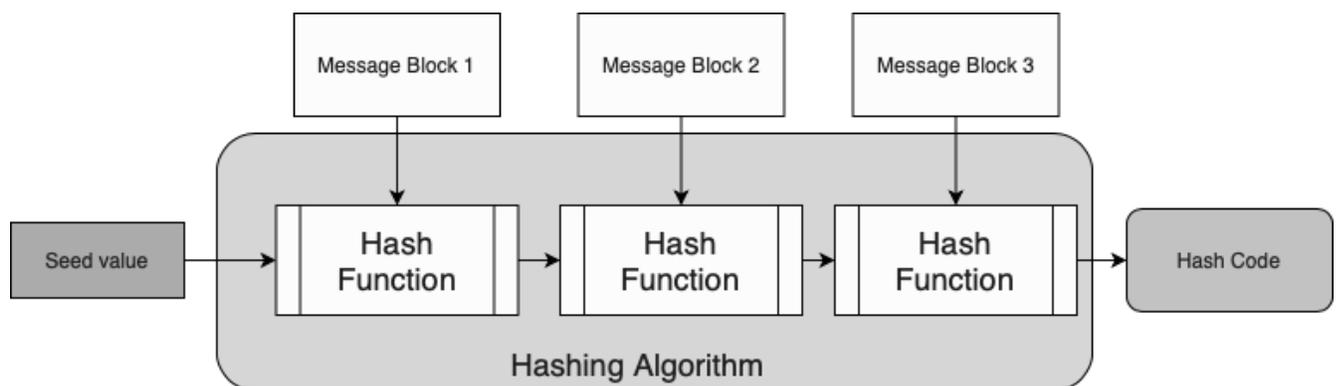


Figure 18 Architecture of hashing algorithm

Because the hash value in the first message block is used as an input to the second operation, this alters the output of the third operation and so on. This is known as the avalanche effect in hashing.

The Avalanche effect produces significantly different hash values when two messages differ even by a single bit.

The hashing algorithm and the hash function are two different aspects. The hash function creates a hashcode by using two blocks of binary data with fixed lengths. Creating a hashcode for a message using the hash function is described in the hashing algorithm. It specifies how the message will be broken down and how results from previous messages are grouped. [44]

Security of cryptographic hash functions

Cryptography equipped in the Bitcoin blockchain is remarkably secure and has stood the test of time. In other blockchains, comparable security techniques are used and are also considered very secure. However, specific security issues such as the possibility of generation and usage of duplicate signature nonces in digital signature schemes (leading to private key recovery attack), chosen prefix collisions in hash functions, and risk of quantum attacks that may break the principal cryptographic algorithms remain a fascinating area of research. [45]

Collisions in hashing

An ideal hash function should be free of **collisions**. Collision is an instance in which two inputs result in the same output. Collisions are considered to weaken a hashing algorithm, as there is a possibility to get the expected result with the different input. As hash functions are used in the digital signatures of certificates, storage of passwords and blockchain signing, a hash function susceptible to collisions could permit a malicious hacker to retrieve passwords from password hashes. A weak hashing algorithm, full of collisions, could be abused in a man-in-the-middle attack, potentially allowing a hacker to spoof a Secure Sockets Layer (SSL) certificate. [46]

MD5, the algorithm used in the Lab component, is regarded as inadequate for cryptographic hashing. Cryptocurrency implementation of Blockchains such as Bitcoin uses more secure hash functions, such as SHA-256 and RIPEMD-160.

Let us understand the concept of collisions using the below example. [47]

As we understand by now, Hashing is the process of converting a set of characters into a fixed-length value. Let us assume that we only want to convert these characters to numbers. This fixed-length number value representing the original set of characters is called a hash value. A hash function will get the hash value from the original set of characters. Various algorithms can be used to design the hash function. For simplicity, we can use the relatively simple function $h(k)$ as our hash function. It takes the sum of ASCII values of the first 3 characters of the input and returns the remainder of the sum when divided by 5.

$$h(k) = \sum_{i=0}^n (\text{ord}(k[i])) \% 5, \text{ where } n=2$$

'ord()' return the ASCII value of character

Observe the highlighted rows in the above table; the hash value of Input, India and Sweden are identical; this is called a collision. It can be represented as

$$h(k)=h(k')$$

here `h` is the hash function, `k` is input as `India` and `k'` is input as *Sweden*.

Input	Hash Value			
Israel	$(\text{ord}("I") + \text{ord}("S") + \text{ord}("R")) \% 5$	$(73 + 83 + 82) \% 5$	$238 \% 5$	3
Peru	$(\text{ord}("P") + \text{ord}("E") + \text{ord}("R")) \% 5$	$(80 + 69 + 82) \% 5$	$231 \% 5$	1
India	$(\text{ord}("I") + \text{ord}("N") + \text{ord}("D")) \% 5$	$(73 + 78 + 68) \% 5$	$219 \% 5$	4
Fiji	$(\text{ord}("F") + \text{ord}("J") + \text{ord}("I")) \% 5$	$(70 + 74 + 73) \% 5$	$217 \% 5$	2
Canada	$(\text{ord}("C") + \text{ord}("A") + \text{ord}("N")) \% 5$	$(67 + 65 + 78) \% 5$	$210 \% 5$	0
Sweden	$(\text{ord}("S") + \text{ord}("W") + \text{ord}("E")) \% 5$	$(83 + 87 + 69) \% 5$	$239 \% 5$	4

In modern applications, the password's hash is stored in the database instead of the password itself, and then the hash of the entered password is compared with the stored hash. Suppose that if this application uses the hash function above-mentioned, the user tries to log in by entering 'India' or 'Sweden' as password, and the application will validate them as both have the same hash as number 4.

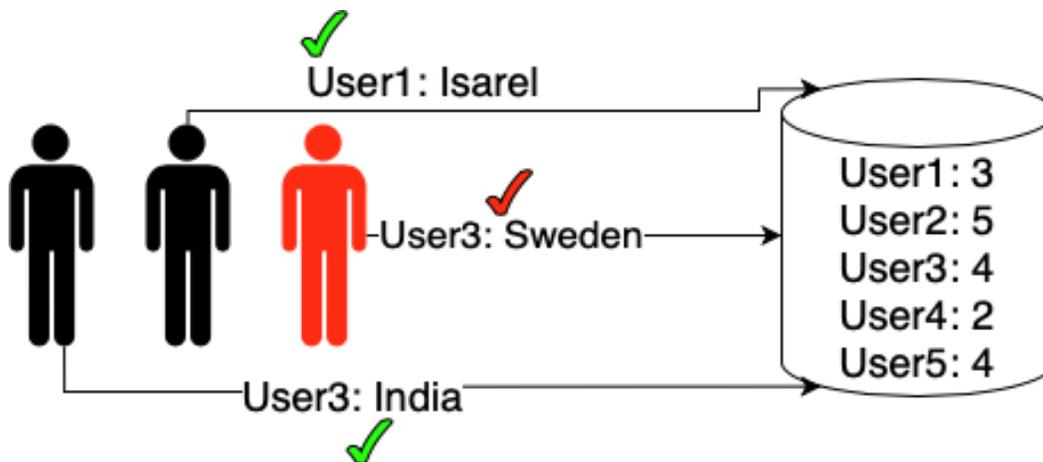


Figure 19 illustrates a collision in password hashes

This collision is highly dependent on the used hash function and size of the input set. Since we want to use the hash functions for any input, the hash function algorithm needs to have some characteristics to decrease the chances of collisions

A good hash function should have four key characteristics:[48]

- 1) The data being hashed is the only thing that determines the hash value.
- 2) The hash function uses all input data.
- 3) The hash function "uniformly" distributes the data across all possible hash values.
- 4) The hash function generates unexpectedly different hash values for even similar or dissimilar strings.

Now, let us dig more about some possible attacks on hashing functions that could affect Blockchain[49]

1. Preimage attack
2. Second preimage attack
3. Collision attack
4. Chosen-prefix collision attack

Preimage Attack

The Preimage attack on cryptographic hash function is referred to as an event when the attacker can find any input k from the hash value H computed using function $h(k)$ such that

$$h(k)=H$$

Here the output of the hash function, H is referred to as Image and the input of the hash function, k is referred to as preimage. [50]

A good hashing function should be resistant to preimage attack means it should be computationally difficult to find any input that hashes to that output. In other words, we can say

the good hash function should act as a one-way function. However, any hash function cannot be purely one way because hash functions produce fixed-length hash values for a larger domain. So, every hashing function would be prone to brute-force attack meaning the attacker can try every combination possible with the hash function and check if any of the input produced the same hash 'H'

So, we can deduce a more precise definition by accounting for the above fact. A hash function will be considered preimage resistant if a brute-force attack is the fastest way to compute a preimage.

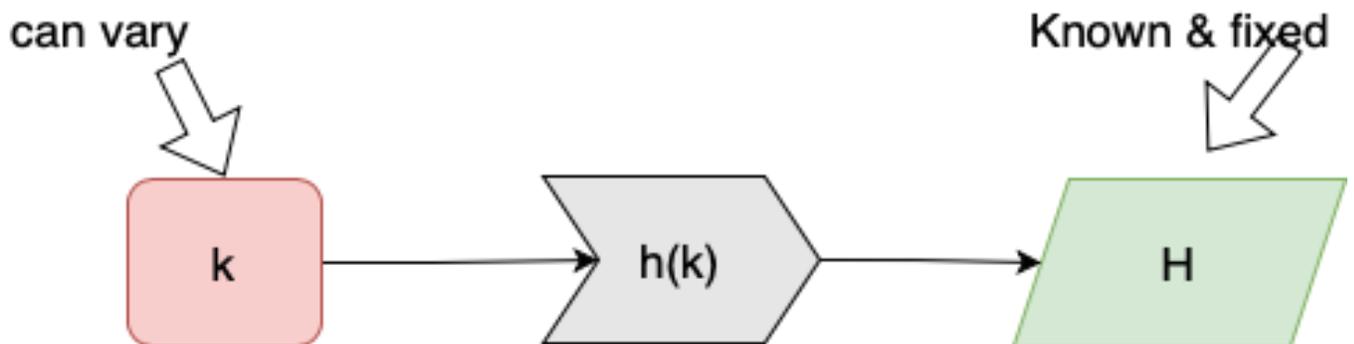


Figure 20 preimage collision

In the above illustration red box means an attacker can vary its value, and the green represents a fixed value that cannot be changed.

Preimage attack effect on Blockchain

The Blockchain uses Hashing for creating a hashing puzzle where computer nodes in the network compete to solve the hashing puzzle. This concept is known as proof-of-work, and nodes find the hash of the transactional data, which has a specific pattern based on the difficulty level. The nodes add a nonce to the transactional data in order to find this hash. However, what if an attacker can work the other way around. The attacker can potentially abuse the hashing function by using a preimage attack to calculate the nonce based on the pattern of hash they are looking for. Due to this, the attacker can create a hash of the transactional data easier than other participating nodes working in the system and will have the advantage of getting the reward.

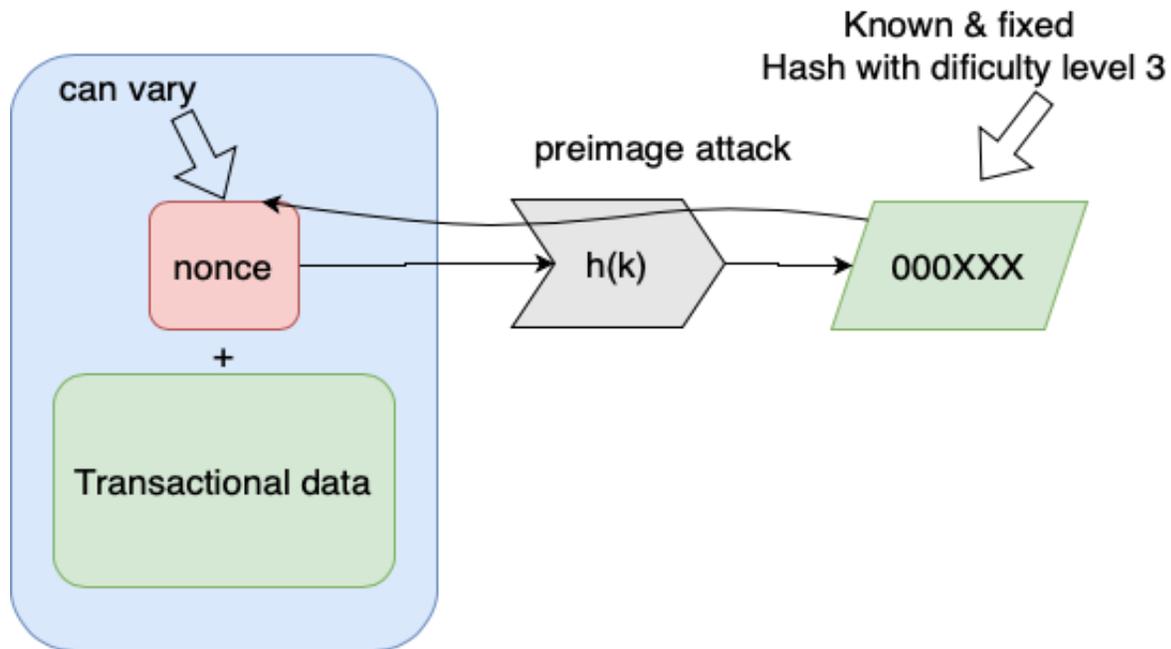


Figure 21 preimage attack

Second Preimage attack

The second-preimage attack on the cryptographic hash function is referred to an event when the attacker can find an input k' that has a hash value H computed using function $h(k')$ such that there exist an input k has same the hash H computed using function $h(k)$.

$$h(k) = H \qquad h(k') = H$$

Here the output of the hash function, H is referred to as Image and the input of the hash function, k is referred to as preImage and k' is referred to as Second-preImage.

A good hashing function should be resistant to second-preimage attack means it should be computationally difficult to find another input that produces the same hash value. In other words, it allows an attacker who has the desired message H_1 to find another message H_2 that has the same hash value. Often, if a hash function is vulnerable to a preimage attack, it is also vulnerable to a second-preimage attack. [51]

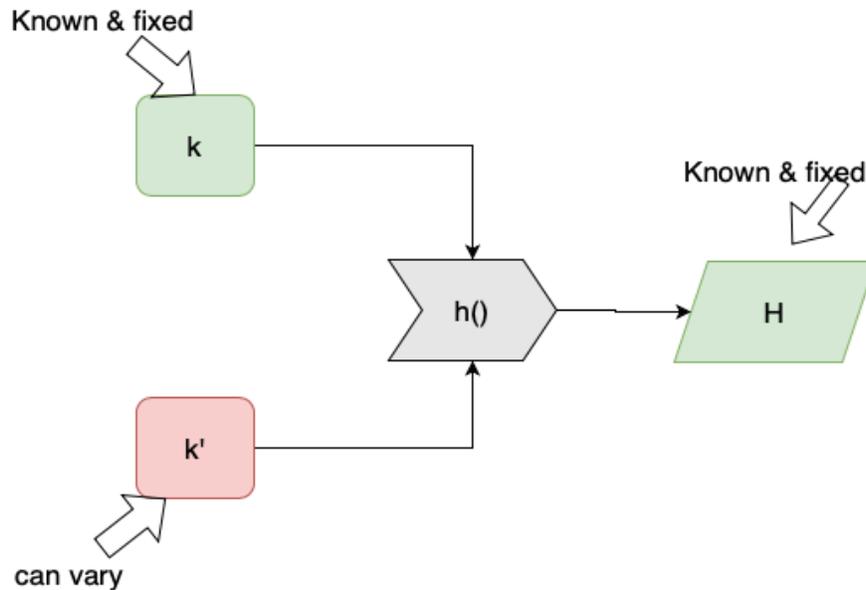


Figure 22 Second preimage collision

In the above illustration red box means an attacker can vary its value, and the green represents a fixed value that cannot be changed.

Second-preImage attack effect on Blockchain

This type of attack, if feasible, can have a significant effect on the core of blockchain technology. At its core, the transactional data and other vital data are combined in a block. The hash of this block is calculated and becomes part of the next block combined with new transactional data. This process is repeated to form a tightly coupled chain with all previous blocks. So, if the data of anyone block is altered, the hashes of all the blocks will be invalidated, and any attempt to change data will be identified.

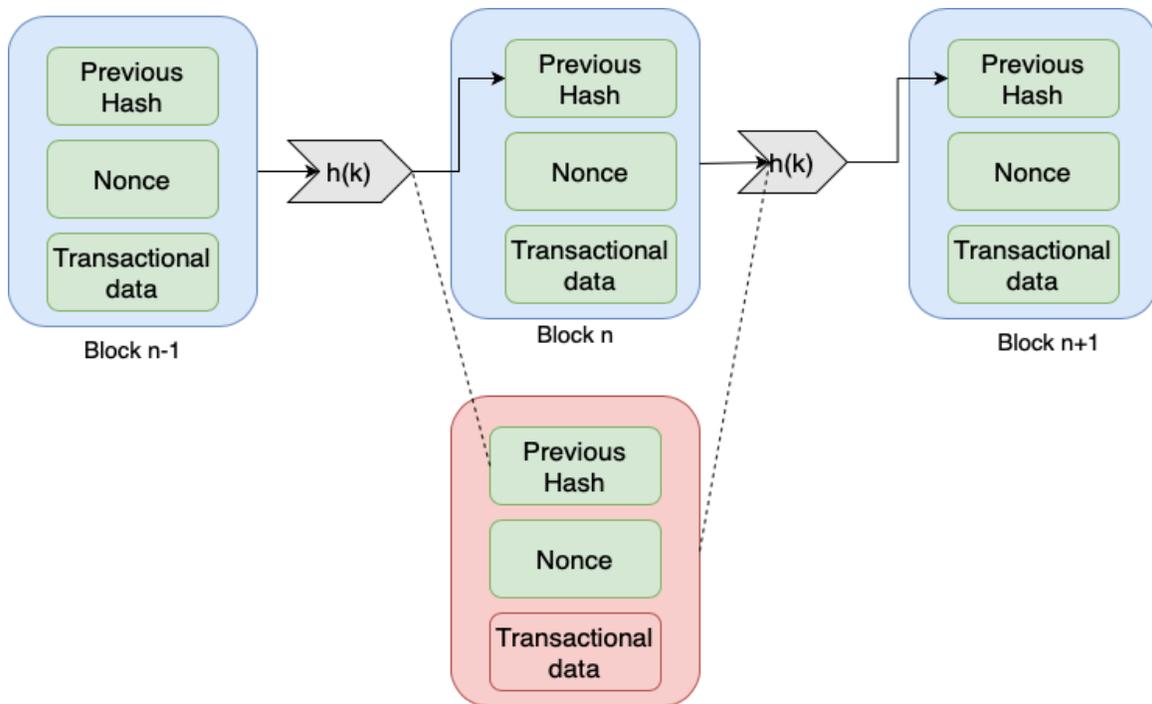


Figure 23 Second preimage attack

If the hashing function used in the Blockchain is abusable to a second-preimage attack, then the attacker can potentially change the transactional data without altering its hash value. Hence the attacker may silently steal cryptocurrency funds without causing any change to the chain of the Blockchain.

Collision Attack

A collision attack in hashing can be seen as just an extension of a Second-preimage attack, and it can be defined as an event if an attacker can find any two inputs k and k' such that they have the same hash value using the hash function $h()$. [52]

$$h(k)=h(k')$$

The attacker is only looking for random two or more inputs of hash function that will result in the same hash. Identifying such inputs in significantly less time than the brute force search time will only categorize the hash function as not collision-resistant.

Collision attack effect on Blockchain

The collision attack would not significantly affect the security of cryptocurrency implementations. Bitcoin hashes the private key of the wallet to add some extra security. In the worst case possible, if one of the randomly chosen inputs from a set of inputs of collision attack is being used as a hash of an active private key, then the attacker can potentially get hold of the funds linked to the affected private key.

Chosen-prefix collision attack

The Chosen prefix collision attack can be viewed as an extension to a collision attack. In this attack, the attacker starts with two different files and append some calculated values in the files to get the same hash value of both files. This attack is considered much more powerful than a collision attack. It can be represented as

$$\mathbf{h(k + m) = h(k' + m')}$$

Here k and k' are entirely different inputs, m and m' are calculated inputs when concatenated with k and k' respectively produces the same hash value by the hash function $h()$.

Though this attack can heavily undermine the security where cryptographic hashes are used, it does not have any implications in Blockchain. For the attack to be considered successful, the attacker must control the input to the hash function. [53]

Cryptography

Apart from Hashing, Blockchain also relies on another technology called Asymmetric Cryptography. It is the base for identifying users and protecting their ownership in the Blockchain. Cryptography is often regarded as complex and challenging to understand.

However, we can use examples to understand in simpler terms.

If you have ever sent or received a postcard, you would have observed that anybody can send the mail to you, and it reaches your mailbox. Although, the sender needs to have your address to send mail to you. However, mailboxes are designed so that anyone can drop mails to your mailbox, but only you're with access to your mailbox key can open the mailbox and read the mails.

The security concept, in this case, is based upon separating two types of information. First, public information that serves as an address for a trapdoor-like container (in this case, your mailbox); second, private information serves as the key to unlock the box and gain access to the contents. When protecting private data, the Blockchain uses the same principle. This example may help you to learn more about cryptography.

In Blockchain, it is vital to be able to identify property owners and ensure that only lawful owners have access to their property.

Blockchain is a peer-to-peer system that is open to everyone. Anyone can connect to the Blockchain and submit transaction data or computational resources. It should not be possible for everyone to have access to the property assigned to accounts using the Blockchain. Private property must have exclusiveness characteristics. The owner of the account that has the ownership should be the only one who can transfer ownership to another account. The Blockchain's challenge is to protect the property that has been assigned to accounts while allowing for the open architecture of the distributed network.

Accounts can be thought of as mailboxes. Anyone can transfer property to them, but only the account owner can use the items that have been stored inside. A mailbox's location is known. Anyone can place something inside, but only the owner can access it with a key. Public-private key encryption is the digital-world equivalent of public mailbox address and private mailbox key. Public keys can identify the account to transfer ownership, while private keys restrict anybody else from using the possessions.

To understand public-private encryption, firstly, we can understand some basics of cryptography[54]

- The idea of cryptography
- Terminology
- Symmetric cryptography
- Asymmetric cryptography

The idea of cryptography

Cryptography's primary purpose is to prevent unauthorized people from accessing data. It acts as a digital version of bank safes or door locks that protect their contents from unauthorized access. Cryptography uses keys to secure data, just like locks and keys in the real world.[55]

Terminology

Encryption is the digital equivalent of closing a lock. Decryption is the digital equivalent of opening a locked door. When talking about cryptography data protection, we use encryption and decryption to protect and unprotect data. Encrypted data is called ciphertext. The ciphertext is a collection of letters and numbers that looks like gibberish to anyone who does not know how to decrypt. Cypher text can be useful, but only those with the key to decrypt it and can use it. The decrypted ciphertext is identical to encrypted data before encryption. The whole journey through cryptography can be summarized as follows: Start with some data, create ciphertext by encrypting it with a cryptographic key, then preserve or send the cyphertext to someone. Finally, recover the

original data by decrypting it with a cryptographic key. Figure 124 illustrates the fundamental functioning of cryptography.[56]

If somebody tries to decrypt the cypher text is an incorrect key, the result would be again gibberish and will not reveal any information about the original text.



Figure 24 Illustration of the essential cryptographic process

Symmetric Cryptography

Cryptography can be utilized in a way that the same key could be used for both encryption and decryption of data. Anyone who could encrypt data using such a key can also decrypt the ciphertext created with that key. This is symmetric cryptography, as the same key can be used in both methods. Figure 25 illustrates the fundamental functioning of symmetric cryptography, where the same key is used to encrypt or decrypt a message.



Figure 25 Schematic illustration of symmetric cryptography

While Symmetric cryptography is used to solve numerous modern problems, Blockchain uses a similar concept but slightly different approach known as Asymmetric cryptography.

Asymmetric Cryptography

Asymmetric cryptography always uses two complementary keys. There is a catch to this rule: Ciphertext encrypted with one of these two keys can only be decrypted using the other key and vice versa.[57]

Figure 36 illustrates asymmetric cryptography. From top to bottom, the figure below depicts the round trip from encryption to decryption using two keys. The black and gray keys can be respectively regarded as private and public keys. On the left side, if the original text is encrypted with a private key, it converts it to ciphertext, and then only a public key can be used to decrypt this ciphertext. While on the other hand, if a public key is used to encrypt, a different ciphertext is generated while it can only be decrypted back to the original text by a complimentary private key.

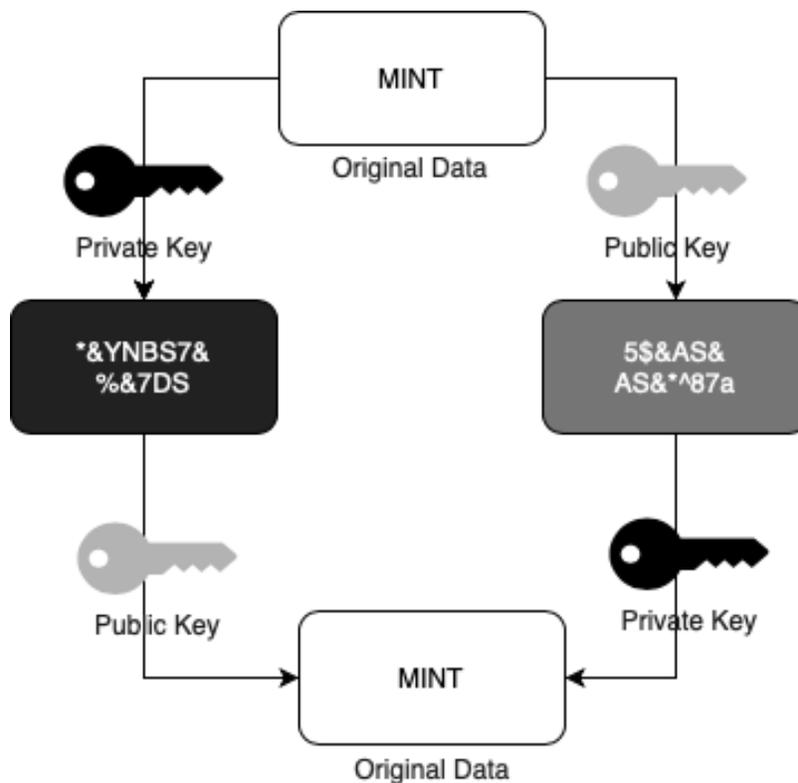


Figure 26 Schematic illustration of asymmetric cryptography

Asymmetric cryptography works because you cannot decrypt ciphertext with the same key used to create it. It is left to the user to decide which key should be used for encryption and which one to decrypt. For every piece of data you wish to encrypt, you can change the roles of the keys, but you must keep both keys to do both encryption and decryption. Your power is limited if you only have one of the keys. Although you can create ciphertext by applying your key data to it, the complementary key is required for decryption. You can, however, decrypt the cypher text created using the complementary key if you already have the ciphertext encrypted with another complementary key. The asymmetric distribution of their cryptographic power allows you to distinguish between those who can create cyphertext and those who can decrypt it.

Usage of Asymmetric Cryptography

Creating and Distributing the Keys

In order to emphasize each key's role in asymmetric cryptography, you will give them specific names when using it in real life. These keys are usually called the public key and private key. Asymmetric cryptography is also known as public-private key cryptography. However, one can use each key to encrypt and decrypt data. These keys can be referred to according to the role they play. Regardless of their trustworthiness, everyone can have a copy of the public key. However, the private key is generally kept private and safe with the owner. [58]

So, these steps are considered while creating and distributing keys:

1. Use cryptographic software to create a pair of complementary keys
2. Name one key as public key
3. Name the other key private key
4. keep the private key confidential
5. Share your public key with everyone

Asymmetric keys can be utilized in two ways, and each way has a different application :

Public to Private

In this approach, the information flows from encryption by a public key and then decryption using the private key. The two complementary keys are used in a similar way to a mailbox, which allows anyone to put in letters, but only the owner can access them. Because it is the simple usage of asymmetric encryption, it is compatible with our intuition about privacy. Our mailbox and address are public, but the contents are private.

This is how asymmetric cryptography works. It allows you to send information securely to the owner. This works because anyone can create cyphertext with the public key, but only the owner can decrypt it and read the message.

Private to Public

In the approach, the information flows from encryption by a private key and then decryption using the public key. This is similar to a public notice board or public news board that allows everyone with a copy of the key to read messages, but only the owner can create messages. This is how asymmetric cryptography can be used to prove authorship. Because using a public key, everyone can decrypt the encrypted cypher text created by the owner using the private key. The phenomenon that only the encrypted message created by the owner can be decrypted by the public key act is proof that only the owner has the private key and the message is from only him.

Asymmetric Cryptography in the Blockchain

The Blockchain has two applications for usage of Asymmetric Cryptography

- Identification of accounts
- Authorization of transactions

Identification of accounts

The Blockchain must identify users to maintain the mapping between property and owner. To identify user accounts and transfer ownership between them, the Blockchain uses the public to the private approach of asymmetric encryption. The blockchain accounts numbers are public

cryptographic keys. Transaction data uses the public cryptographic key to identify the accounts involved in the ownership transfer. The Blockchain treats user accounts in this way as mailboxes. They are publicly identifiable and can be sent messages by anyone.

Authorization of transactions

Transactional data contains the properties of a valid transaction to take places such as sender address, receiver address, amount, timestamp, and a bunch more.

Transaction data must always include data that proves that the account owner who has given up ownership agrees to the transfer of ownership. This agreement implies that information flows from the account owner to all who can view the transaction data. This information flow is similar to the private-to public use case for asymmetric cryptography. The account owner who gives away ownership creates a ciphertext using one's private keys. Furthermore, when everybody else receives it, they can use the public key to verify that indeed the owner has created this transaction. The details of this process are discussed in the next section, which is called digital signature

Digital Signatures

Up until now, we have an idea of Asymmetric cryptography, why it is used and how it can be used in Blockchain. It can also be noted that the Blockchain uses the public cryptographic key as account numbers and employs the public-to-private approach of asymmetric encryption to transfer ownership between accounts. Authorization is defined as having the authority to perform a given task. Blockchain must ensure that only the legal owner can transfer property to other accounts. Here is where authorization can be added. This chapter describes how authorization is done within the Blockchain using asymmetric cryptography. This step focuses on digital signatures. These digital signatures use the private-to-public approach of asymmetric cryptography.[59]

The whole goal of cryptocurrency systems is to be able to securely send and receive funds without the central body of the trust. So, there must be a system to create a trust that verifies the transitions and prevent them from being tampered with on the way as Blockchain uses the peer-to-peer system. Everyone can create transactions and submit them to the peer-to-peer system. Transaction data is the basis of clarifying and describing ownership. Only the legal owner of an account should have the right to transfer property to another account. Blockchain's commitment to preserve its openness and have the ability to transfer ownership to the lawful owner is a challenge in itself

To better understand the problem, let take an analogy of digital signatures to handwritten signatures. Handwritten signatures have an essential function: they provide the agreement to a document or cheque's contents and agree to its execution or encashment. We can agree that handwritten signatures prove agreement because of the uniqueness and character of each individual's handwriting. Each person has a unique way of writing their name. When a name is written in a particular manner, it means that that person has signed the document. This is how electronic ledger transactions can be deemed to have been agreed upon using digital signatures. It is similar to handwritten signatures. This concept is essential for the security and integrity of individual transactions on the Blockchain. The goal of the digital signature is to make sure that

only the account owner can transfer its property to other accounts. Any attempt to access an account or its property by anyone other than the legal owner should be rejected. [60]

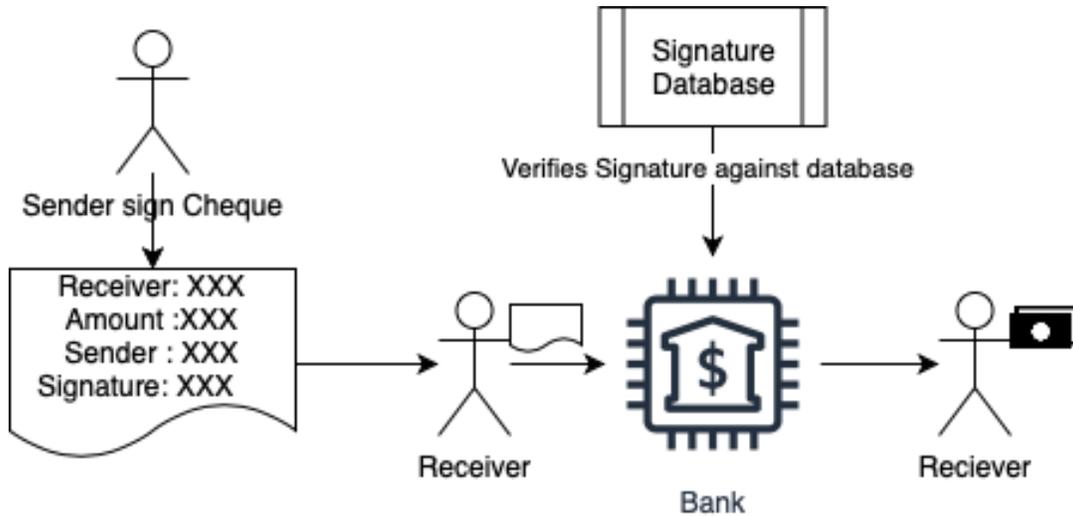


Figure 27 Example of handwritten signature used in cheque

Usage of Digital Signatures

Digital signatures can be used as digital handwritten signatures in Blockchain. They use cryptographic hashing and the private-to public information flow of asymmetric encryption. This is example explains the three main elements of digital signatures.

- Creating a signature
- Verifying data by using the signature
- Identifying fraud by using the signature

Creating a signature

Let us consider that I want to send a message saying ‘MINT IS AWESOME’ to the whole world in an authorized way. I will use asymmetric cryptography to create a public and private key. However, first, I will use hashing (explained earlier) to get the hash value of the message. Then I will use the private key to encrypt this hash to the ciphertext and share the ciphertext plus the original message and my public key to the open world. The process starts with a white box on the left with the message ‘MINT IS AWESOME’, its hash is created, which is 03BC71E6 represented in the circle. Then the hash value is encrypted with a private key to get the digital signature, and it represents by a tape symbol in figure 28 below. Both message and digital

signature combined in the grey box is shared with the world and is known as a digitally signed message.

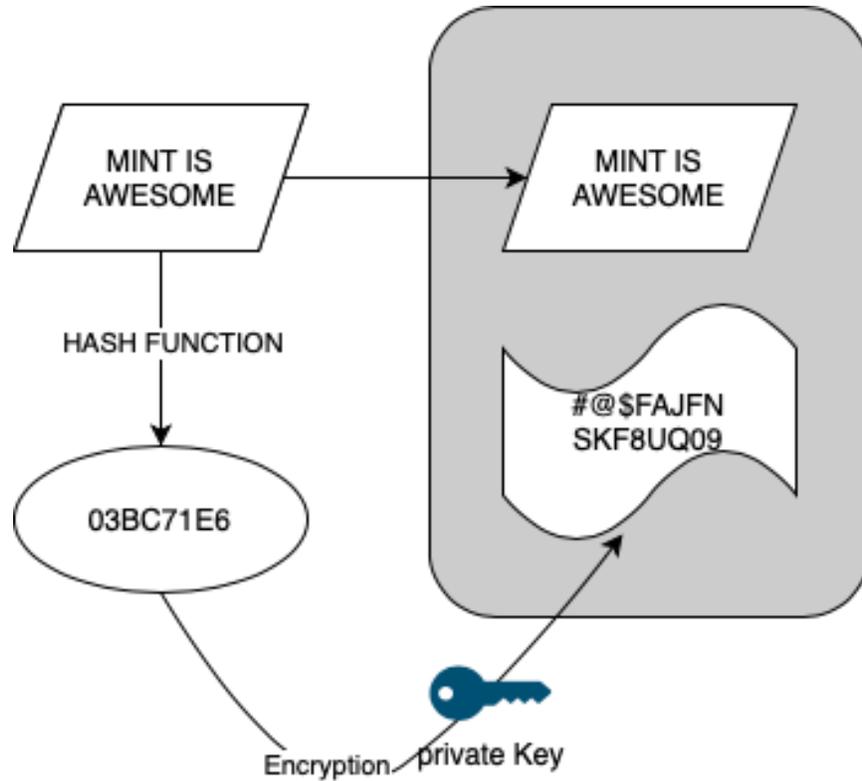


Figure 28 creation of a digitally signed message

Verifying Data by Using the Signature

My message, along with my digital signature, has been sent to the entire world. Anyone can confirm that I have authorized this message by using my public key. This process of verifying the message using the digital signature is illustrated in Figure 29. The recipient of the digitally signed message first calculates the hash of the message itself. This returns 03BC71E6. Next, the recipient of my message decrypts the attached cyphertext (the digital signature) using my public key. This will give him the value 03BC71E6. This is the hash value for the message I intended to send the world. The verification is completed by comparing both hash values. Both hash values are identical, so the recipient concludes that I signed the message. He was able to decrypt the signature using my public key. Second, the recipient believes that the text in the message is the one I sent originally and not

altered in between. The decrypted Cypher text matches the hash value for the message in the signed message.[61]

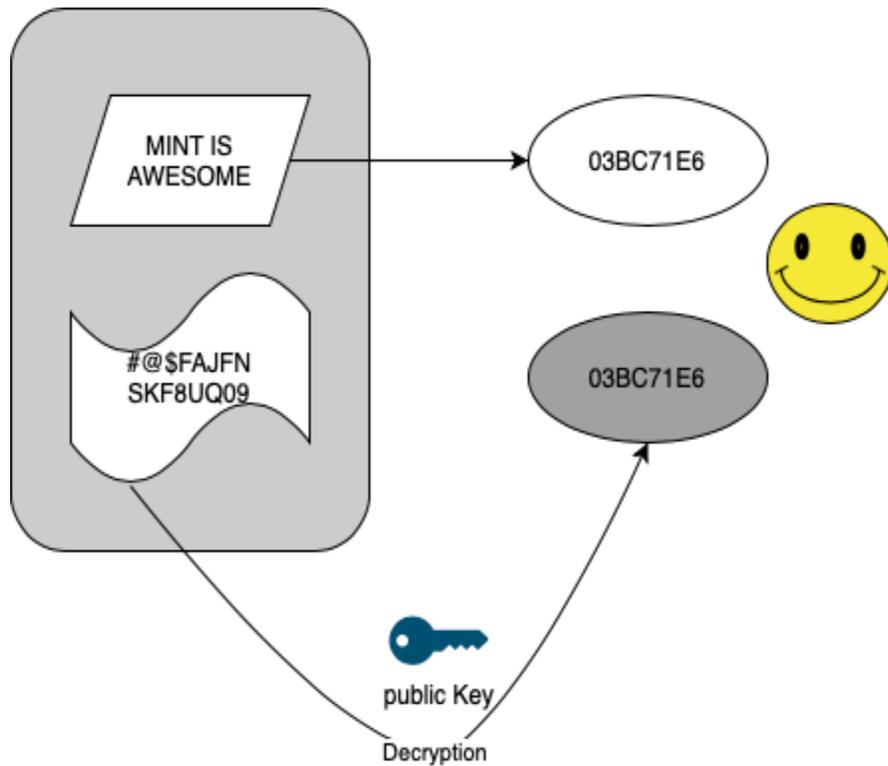


Figure 29 Verifying the signed message

Identifying Fraud by Using the Signature

Figure 30 depicts the message that arrived at my friend's mailbox. Notice the alteration in the message text. A hacker added a 'NOT' in between, changing the message's meaning. This is not how I think of the MINT program. The digital signature will alert everyone that my message was altered.

The first step is for the receiver to create the hash value of the received message, which will give him the value 993B7D13. Next, the recipient of my message will decrypt the digital signature using my public key. This will provide him with 03BC71E6, the hash of the message I intended to send out to the world. The hash values of both messages are different. This shows that the message in my digitally signed message was not what I intended to send to the rest of the

world. Everyone concludes that I did not authorize the message, and therefore no one can hold me responsible for its contents.[62]

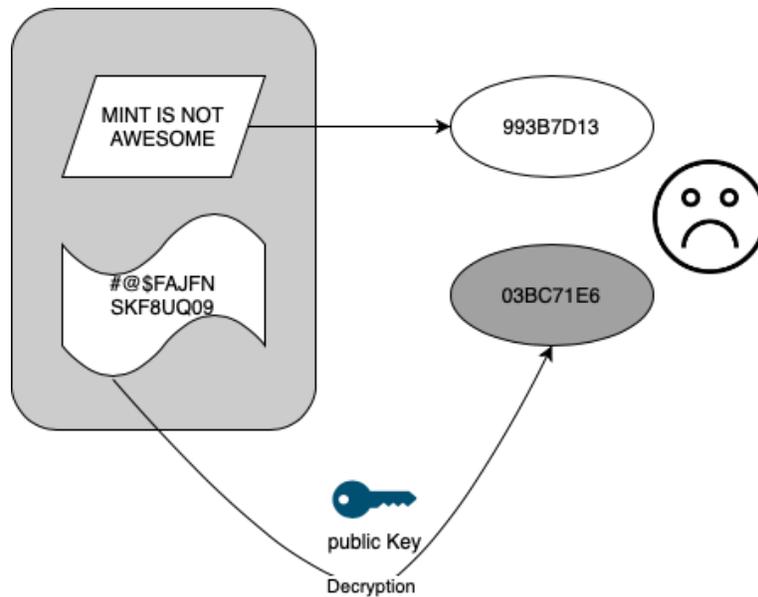


Figure 30 Identifying fraud in a signed message.

How Digital Signatures are used in Blockchain

Digital signatures provide very useful tools to the Blockchain. First is the ability to have a one-to-one relationship; only the property owner, in this case, who has the private key, can sign the signature while others can verify it but cannot forge it. The other most important aspect of verifying the truthiness of the agreement is that if the contents of the agreement are changed at any later point in time, the digital signature will invalidate the agreement. Furthermore, all this is very easy to verify by others.

So digital signatures have two use cases in Blockchain

- Signing a transaction
- Verifying a transaction

Signing a transaction

To create a digital signature to authorize a transaction, the account owner must perform the following steps:

1. Details of the transaction, including all information, such as sender and receiver address and the amount transferred. A signature is not required as it is not available.
2. Compute the cryptographic hash value of the transaction data.
3. Encrypt the hash value of the transaction using the private key of the sender.
4. Add the ciphertext generated from step 3 to the digital message.

Verifying a transaction

The following step must be performed in order to verify the transaction:

1. Compute the cryptographic hash value of the transaction data except for the digital signature.
2. Decrypt the transaction's hash value using the sender's public key, which is the sender's account number.
3. Compare the hash values from steps 1 and 2. If they are identical, it implies the sender authorized the transaction; otherwise, it is regarded as invalid.

Weakness of Asymmetric Cryptography

As Asymmetric cryptography is one of the core technologies used in Cryptocurrency systems, it is crucial to identify any potential weaknesses associated with its usage. [63]

Algorithms

In theory, all public-key schemes are somehow vulnerable to brute-force key search attack[64], an attack carried out by guessing every possible key value. However, these attacks are impractical if the cost of attacking is more than the benefit itself, or it would take an eternity to do it. The computational power needed to carry out such an attack is termed as the "work factor" by Claude Shannon. The work factor can be easily increased by increasing the key size. Even some algorithms effectively resist brute force attacks irrespective of key sizes, while others are more prone to attacks based on their implementations. The RSA algorithm discussed in the Lab

component of this report is vulnerable to integer factorization. By factoring in the large prime number that makes up the RSA keys, the private key can be generated from the public key if smaller key sizes are used.

Man in the middle attack

A communication using asymmetric keys is potentially vulnerable to a ‘Man-in-the-middle attack’. This attack is carried out by wiretapping the network between two parties and successfully placing themselves between them. A man-in-the-middle attack has become challenging to implement, thanks to the complexities of modern security protocols. However, public networks or wireless connections are more vulnerable to a man-in-the-middle attack. If successful, the attacker can sniff the private data shared among the two parties without even detection and worse if the attacker can also change the active communication data. Figure 31 illustrates how an attacker can use a set of public-private keys to spoof a secure connection. [65]

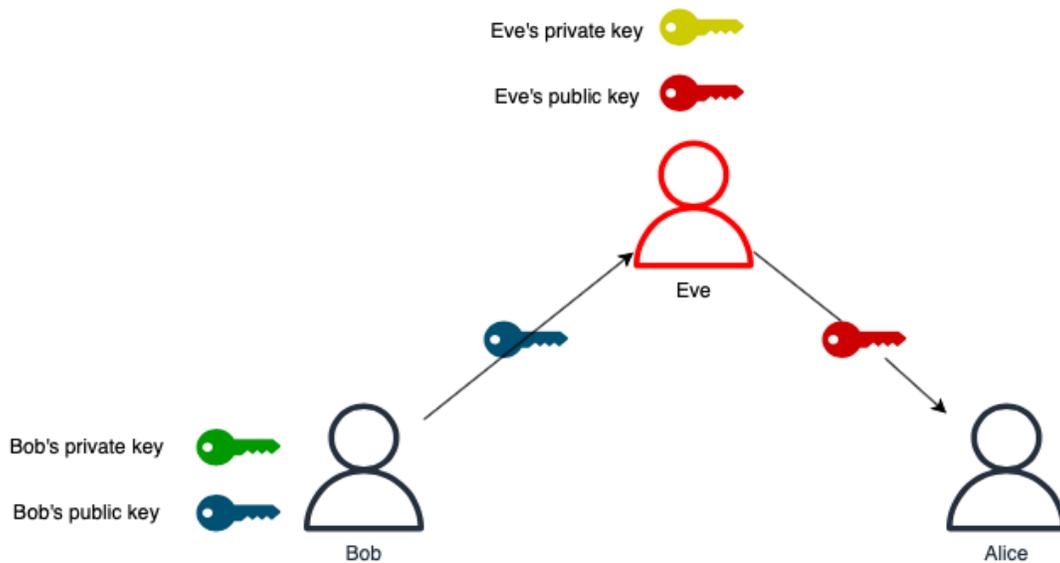


Figure 31 Illustrate man in the middle attack

Two types of Digital signature

There are two major public-key algorithms used for digital signatures, namely RSA and ECDSA. We can further explore these in the next section.

RSA

RSA is one of the earliest implementations of public-private cryptography. It is based upon the principle of prime factorization to generate the key pair. The factorization of large prime numbers acts as a trap door function.

The public and private key pairs are generated based on two large prime numbers. The public key is openly available to everybody, while the private key is kept secret. The prime numbers are kept secret as well. As long as the decided prime numbers are pretty large, it is considered impossible to compute the private key from the public one. The RSA algorithm is based on the problem of integer factorization, which defines that the difficulty of prime factorization is directly proportional to the size of the prime numbers.[66]

We will learn more about RSA in the Lab component section of this report.

ECC

ECC is short for Elliptic-curve cryptography and is popularly used in digital signatures. It is based on the addition of points on the elliptic curve to generate public-private key pairs. ECC requires comparatively smaller key sizes than other asymmetric key algorithms, such as RSA.

ECC provides the equivalent level of security as RSA but uses a relatively smaller key size. A 224-bit ECC key is comparable to a 2048-bit RSA key and so on. ECC has a significant advantage over RSA as it requires fewer computations to generate, store and verify digital signatures.

Due to this reason, ECC is popularly used in major cryptocurrency addressing systems for transaction signing procedures. It is also prevalent in other blockchain applications and in SSH, SSL/TLS, iMessage and Tor. [67]

Mining and consensus

Mining in cryptocurrency is critical for achieving consensus in a decentralized blockchain network. Virtual any node in the cryptocurrency network can perform the mining operations, and in return, these nodes are rewarded for their contribution to mining. Though people see that they do mining just for rewards, the reward is just one part of mining. Mining is the heart of the whole blockchain network; it helps to achieve the consensus among all the nodes and build a decentralized network that is trusted by all the participating nodes.

In almost every cryptocurrency, the miner validates the transaction and initially store them in its local memory pool. After enough transactions are stored in the pool, the miner combines them to create a block. The creation of the block requires a solution of the hash puzzle to include the block in the distributed ledger. The miner gets two types of awards for mining the block, one is the transaction fee of each transaction, and the other is r the newly created cryptocurrency in each block. The transaction fees are paid by the creator of the transaction for processing the transaction through the network. Each block has a reward transaction that creates new cryptocurrency funds and awards them to the creator of the new block. This reward transaction is called a Coinbase transaction.

There is a limit on the amount of currency in circulation for cryptocurrencies such as Bitcoin. When the limit is reached, the minting of new currency funds will be stopped, and there will be only transaction fees as a reward to the miners. [68]

Mining a block

Any node can be a miner by creating a new block in the blockchain network. A participant only has to run the full blockchain software to be a miner. Participants can even use a minimum hardware specification computer to run this software, but the cryptocurrency mining itself is no longer considered profitable using standard computers. Due to high competition among participants, the mining difficulty of the Bitcoin system has drastically increased. As a result, miners are now using a specialized computer with Application-Specific Integrated Circuits (ASICs). This gives the miner a better chance of solving the hash puzzle and creating the next block in the Blockchain. The miner has to perform specific tasks to mine a block successfully.

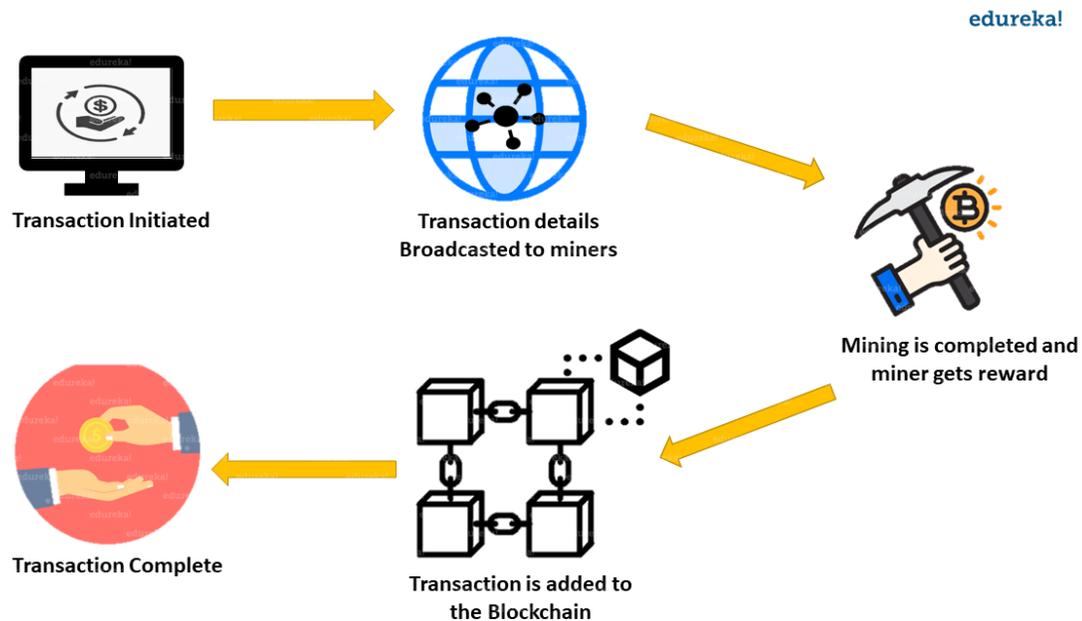


Figure 32 Cryptocurrency Mining processes[69]

Cryptocurrency transaction validation processes

We know that public Blockchain is entirely open to everyone, which means that anyone can join the blockchain network and contribute to the validation process. The goal of the validation process is to ensure that only valid transactions are added to the Blockchain. To achieve this, all the nodes in the network can act as validators and supervisors for other validators. To motivate the supervision, nodes are rewarded for finding errors in the other's work, and a reward is provided for adding valid and authorized transactions. As a result, the system has an incentive to validate and supervise fellow nodes. [70] [71]

How does it work?

The Blockchain system follows a set of instructions that govern the process of adding new transactions and blocks.

Validation

For Cryptocurrencies, the Blockchain has two types of data, the blocks themselves and the transactional data within those blocks. Each of these groups has its own distinct validation rules. The transactional data is validated for the structure of each transaction against transaction rules. These rules may differ from application to application. The validation for blocks checks for the formal and semantic correctness of the block headers. The core element of checking the block header is to verify the hash puzzle or proof of work. Only the validated blocks containing the hash puzzle's correct solution are added to the Blockchain, and the rest are discarded.

Reward

Creating blocks costs energy, time and money as blocks are only added after a computationally expensive hash puzzle is solved. The hash puzzle is an integral part of the proof of work algorithm, which makes the Blockchain immutable. Therefore, solving the hash puzzle is a necessity for the working of the whole system. A reward is offered to motivate the miners to solve the hash puzzle. We can see this reward as the cost of maintaining the integrity of the whole system.

Competition

The presence of reward entails the presence of competition in the system. The competition in Blockchain can be sub-divided into two phases, and the first one is the speed competition. Only the first miner who correctly solves the hash puzzle and submits the solution gets the reward after verification. The other phase is competition for validation, and other miners can still win the reward if the submitted solution is not valid. Every miner first competes to calculate and submit the block and also competes to validate the submitted blocks by others. This creates a competitive equilibrium for the submission and validation of blocks in the blockchain system.

At any given point in time, all the nodes in the network are either evaluating a newly submitted block or working on the puzzle to submit a new block themselves.

The following rules are followed by all the nodes of the blockchain network:

1. New transactions and new blocks are constantly forwarded to all the nodes.
2. When new transaction data is received, each node adds the transaction to its transactional pool and selects it for processing.
3. When a new block is received, nodes will verify it with top priority.
4. The newly received transactions are validated for authentication and semantics correctness.
5. The valid transactions are added into the Merkle tree, and nodes solve the hash puzzle to create the new block.
6. If a node solves the hash puzzle, it immediately creates and shares the block in the network
7. Each node that receives this new block starts verifying the hash puzzle solution and transaction data.
8. If the received block is found invalid, it is discarded, and the node continues to solve its version of the hash puzzle.
9. If the received block is valid, it is added to the node's copy of Blockchain and verified transactions are deleted from its transaction pool.
10. If the received block is deemed invalid at a later point in time. The block and subsequent blocks are removed from the Blockchain, and their transactions are again added back to the transaction pool.
11. The node whose block is accepted in the chain receives the reward and all the transaction fees.
12. If the block is removed at a later point in time, the reward is also revoked from the node.

Double spending problem

A double-spending problem refers to a situation when the same fund amount can be spent more than once. This is very hard in a physical currency such as paper or coins because to double-spend a physical currency, the Attacker needs to create a copy of cash or coins, which is expensive. However, digital currency can be replicated as it is virtually just bits and bytes, unlike physical currency. The double-spending problem is avoidable if a centralized authority can manage and approve transactions, but it is a significant issue when the digital currency is being managed by a decentralized system such as Blockchain. Bitcoin is the first cryptocurrency that provided the practical solution to avoid this problem. However, there are still some vulnerabilities that can be attacked to perform double-spending attacks on cryptocurrencies. One such attack is race attack, which can be exploited to double spend in a decentralized network. [72]

Race attack

All Blockchain is vulnerable to consensus attacks, and race attack is one of the simplest consensus attacks that can be executed in a decentralized network. It is also known as a zero-confirmation attack as it can only be performed before the transaction is added to a block. In this attack, the Attacker will create two conflicting transactions, one transaction is to pay the victim, and the other transaction is to pay the same fund back to the Attacker itself. The Attacker will send the first transaction only to the victim while broadcasting the other one to everyone else in the network. So, this way, the second transaction has better chances of being added to a block before the first transaction and hence gets verified first in the verification race. This attack can affect the merchants who do not wait for confirmations after the transaction, as in the case of Bitcoin, the first confirmation does not come before 10 minutes. Cryptocurrencies such as Bitcoin that take a long time to provide confirmations are more vulnerable to race attacks. [73]

51% attack

The other such consensus attack is the 51% attack, and the 51% attack is a type of attack in which attackers take control of the majority of cryptocurrency's total computing power. If any participant or group in the network can control at least 50% of the hashing power to create blocks, they can manipulate the block creation mechanism. Generally, all consensus algorithms are susceptible to 51% attack; however attackers prominently attack Proof of Work (PoW) based blockchain systems. ^[74]

Implications of the attack

When a participant controls the majority of the computation power in a network, it has more chances of creating new blocks than the rest of the participants. When the attackers have more chances of creating blocks, they have some control over ordering, inclusion or exclusion of transactions. Some people think that the Attacker can add invalid transactions in the blocks, but that is not possible because the blocks will still be verified by the rest of the network and discarded if blocks contain any invalid transactions. Instead, the Attacker would be able to create below scenarios:

- Delay the confirmation of some or all valid transactions
- Prevent other participants from mining the valid blocks

However, an attacker would not be able to:

- Steal or alter transactions from other accounts
- Delay any valid transactions from broadcasting

The Attacker cannot access the information of others even if it takes control of the majority of the network. The only way the Attacker can profit from this attack is by manipulating their own transactions by spending them more than once, the double-spending. As discussed earlier, by in a double-spending attack, the Attacker can spend the funds twice and then reverse the transaction on the Blockchain. As the Attacker already controls the block creation, they can easily reverse their own transactions.

Cryptocurrencies with a high hash power are regarded as more secure because initiating an attack costs a considerable sum of money. As a result, attackers focus on cryptocurrency with low hash power. If a 51% attack is successful, attackers may be able to cancel verified transactions, double-spend the same cryptocurrency, manipulate the coin's price, or even disrupt the blockchain network. Despite the high attack cost, the profitability is also quite large, which drives attackers more than other blockchain attacks. The 51% attack necessitates an enemy developing their chain in secret. According to the longest chain rule, attackers can effectively exploit vulnerabilities if they can establish a chain that is longer than the existing chain.^[75]

What are the risks which are involved with 51% attack?

Users of cryptocurrency may lose digital assets or even cash due to a 51% attack. This raises fundamental questions regarding a blockchain's dependability, security, and integrity. The trust of its consumers and miners has been severely shaken. Young and inexperienced users can be duped into validating and confirming transactions that they can subsequently invalidate. This is because unconfirmed blocks & transactions in a blockchain can be tampered with by attackers.

Furthermore, the attackers may not confirm or reverse the victims' transactions. As a result, consumers begin to distrust the Blockchain, lowering its value. As a result of these attacks, some cryptocurrencies may be delisted owing to security concerns.

51% Attack Real-World Examples

In August 2016, attacks on two Ethereum-based blockchains, Krypton and Shift, totalled 51%. Bitcoin Gold, the 26th-largest cryptocurrency at the moment, was hit with a 51% attack in May of 2018. The attackers had such control over Bitcoin Gold's hash power that they were able to double-spend for several days despite Bitcoin Gold's repeated attempts to raise the exchange thresholds, finally stealing more than \$18 million worth of Bitcoin Gold. In the year 2020, Bitcoin Gold was hit once more. The Bitcoin SV (BSV) network was recently attacked in August 2021.^[76]

Which platforms have faced a 51% attack

Large blockchain platforms like Bitcoin and Ethereum are thought to be secure against 51% of attacks. Compared to smaller initiatives, they were unlikely to face a 51% attack. On the other hand, various smaller projects are vulnerable to this type of attack.

The following blockchain platforms have been subjected to a 51% attack:

Gold Bitcoin

For the first time in 2018, the BTG blockchain was subjected to a 51% attack, resulting in enormous losses. Unlike its fork Bitcoin system, which utilizes the SHA256 consensus method, The BTG blockchain's creators aimed to achieve decentralization by mining with GPUs rather than ASIC equipment. However, the attack was launched after an unknown miner gained control of more than 51% of the global BTG hash rate. The Blockchain underwent two reorganizations in two days, allowing it to double-spend a large sum of money. The BTG community pleaded with the Blockchain to switch to a more secure algorithm. They suspected the BTG network of having hidden ASIC mining machines. (*Understanding a 51% Attack on the Blockchain / Engineering Education (EngEd) Program, 2021*)

Ethereum Classic

In the same month of 2020, the ETC blockchain was subjected to three consecutive 51% attacks. Ethereum Classic, like Bitcoin, uses the PoW consensus algorithm. When used on massive networks like Bitcoin, a 51% attack is costly because it necessitates a massive amount of computational power. Because the ETC hash rate is smaller, it is more susceptible to 51% attacks. Because of the decentralized structure of the ETC PoW, avoiding or mitigating 51% attacks is challenging. The attacks were said to have had no substantial influence on ETC prices, but they damaged users' faith.

How can we avoid a 51% attack[77]

The best of to avoid a 51% attack on blockchain networks would be by increasing the wait time. The more the wait time, the lesser the chance of double spending. With the longer wait times, the

transactions get buried under multiple blocks and becomes more difficult and expensive to reverse transactions

It would be difficult for a single miner or even a group to outbuild the longest confirmed Blockchain and attack the network. The attacker would need to possess powerful hardware and expend much energy to carry off the attack. Furthermore, because the mining process is random, an attacker may require luck.

For example, Bitcoin's hash rate is large and sophisticated enough that renting mining equipment would be a significant starting expenditure for an attacker. Ethereum Classic, on either hand, is more susceptible to threats since it is less closely related to bitcoin in general.

What are its chances to recur?

As this attack is dependent on the attacker's hashing share in the network, a 51% attack has a chance of recurrence. To launch the second attack, an attacker can influence the Blockchain to produce new blocks faster. In conclusion, a blockchain can be attacked again. It is the responsibility of the Blockchain to make their systems safer and more resilient.

Lab Component

The objective of this lab component is to create a functional cryptocurrency with intentional weaknesses throughout the design. Let us first learn about some essential cryptocurrency components and learn the high-level working of cryptocurrency.

Components required to create a cryptocurrency

Blocks

Blocks are the fundamental data structures within the blockchain database, where transactions in a cryptocurrency blockchain are permanently recorded. A block records some or all of the most recent transactions that still need to be validated by the network. Once the transactions are validated, the block is closed and considered mined. A new block is then created to enter and

validate new transactions. A block is also considered a permanent store of records that, once written, cannot be tampered with or easily removed.[78]

Chain of blocks- Blockchain

Blockchain is a kind of shared database that is different from a typical database in the way that it stores information; blockchains can store any type of data in blocks that are then linked together via cryptographic hashing. The data entered into the decentralized blockchains are immutable, which cannot be changed once added. In the case of Bitcoin, this means that transactions are permanently recorded, and linked but irreversible.[79]

Proof of work

Proof of work is a decentralized consensus mechanism that helps to introduce integrity using a trustless system. It requires network members to solve an arbitrary mathematical puzzle to prevent anybody from tricking the system. Due to this proof of work, cryptocurrency transactions can be securely processed in a peer-to-peer system without the need for a trusted third party.[80]

Keys

Cryptocurrency heavily relies on cryptography. Public key cryptography uses a pair of keys known as a public key and a private key that can be associated with an entity that needs to authenticate its identity electronically. Generally, public key is published openly, and the corresponding private key is always kept secret. Data encrypted with a public key can only be decrypted with a corresponding private key.[81]

Wallets

The wallet securely stores the public key pair associated with the user; a wallet is a type of digital holder used to send and receive Cryptocurrency funds such as bitcoins. This can be seen as the digital equivalent to a physical wallet. However, instead of storing physical currency like cash and coins, the wallet stores critical cryptographic information used to access

Cryptocurrency addresses and perform transactions. If an attacker gets hold of a wallet's private keys, they can easily transfer the cryptocurrency to their wallet.[82]

Transactions

A transaction is a structure to send or receive cryptocurrency funds. A transaction data contains the critical information to send funds such as recipient's public address, sender's public address, amount and signature. The transaction is broadcasted to the blockchain network, so every node receives it. Transactions are public and not encrypted, so everybody can view and verify them. Multiple transactions are grouped together to create a block.[83]

Transactional pool

A transaction pool in a cryptocurrency node is a mechanism for storing unconfirmed transactions. It acts as a queue for transactions that are not yet part of any block. When a transaction is submitted, it is sent from a node to its peers, and peers again propagate it to their peers. This continues until the transaction has been received by all the nodes, ready for miners to process them into a block. The transactions from that transaction pool that are not included in the current block will be processed in the upcoming blocks. [84]

Mining

Mining refers to the process of adding new blocks to the blockchain. Mining requires a solution to a cryptographical puzzle that requires very high computation power. This is why miners use computers or specialized hardware to mine and validate cryptocurrency transactions. A miner will combine valid transactions to create new blocks, and if these blocks are accepted, they become part of a public distributed ledger on the blockchain. [85]

Linking all the components together

The blockchain is a distributed and decentralized ledger that stores data within a series of blocks that are publicly shared across all its network nodes. The blockchain consists of a collection of blocks, and each block acts as a storage unit. It has a data field that is used to store information in

the block itself. Every block is then given a unique value. It looks like a random string of characters. This value is called the hash since it is generated from a hash function, more specifically, a cryptographic hash function that generates a unique output for every unique input. The input for the block's hash includes the data it needs to store and other metadata about the block such as when it was created. The last hash value, which is for every new block, is set to the block's hash that came before.

It is a key to making a chain of blocks as the last hash creates links between subsequent blocks in the chain. Each new block reference is a hash of the last block in the current chain. Eventually, with the linkage of the last hashes, we have a chain of blocks creating the blockchain next.

Now we will look into the significance of the blockchain as a ledger. A ledger is a record-keeping book that records all the economic transactions of an organization. It records payments, contracts movements of assets. As a ledger, the blockchain serves the purpose of storing transactional data. The blockchain is a distributed ledger because the ledger itself is shared with everyone using the blockchain network. In this network, multiple nodes or multiple individuals through computers are connected to the blockchain network. They get a copy of the complete blockchain ledger by connecting to the blockchain network. The recorded blockchain provides a history of all the transactions since the blockchain was created. Since everyone receives a copy of the blockchain ledger, they get regular updates whenever a change is made to the blockchain. This relates to the idea of the blockchain being decentralized. This means that there is not one central organization but the shared responsibility of updating the ledger in a centralized model.

On the other hand, in a centralized organization, the ledger is controlled by a single entity. That has complete authority to invalidate transactions. We have to completely trust the bank, the central organization we are relying upon, to record all of our transactions.

So in order to get a decentralized trust-less model, everyone needs to have access to the history of transactions and an equal responsibility to record and validate those transactions in blockchain history.

Why is a blockchain system needed?

The blockchain is fair because it is decentralized and completely public, and due to its decentralized nature, it is also highly secure. There is no one central point of failure. Hacker needs to only take down one bank in a centralized system to access a lot of the data. However, the attacker would need to take over thousands of nodes and computers to hijack the network in the decentralized system.

This relates to the concept of a cryptocurrency, the primary use case for a blockchain. The technology of the blockchain and the cryptocurrency are significantly related to each other that the concepts have almost become synonymous.

The cryptocurrencies leverage the blockchain, but the cryptocurrency is its own technology.

It has a blockchain as only one of its pieces. Primarily a cryptocurrency is a digital media of exchange, and it has three main aspects of technology. First, it has a blockchain that leverages a distributed ledger. Secondly, it also has a concept of wallets and finally a concept of mining.

For the first aspect, the cryptocurrency leverages the blockchain to keep a public database of transactions that everyone can access. If everyone has access to the public database, we need a mechanism to prevent someone from manipulating the data and illegitimately transferring currency to themselves. To prevent such malicious behaviour, the cryptocurrency uses cryptography to protect the blockchain, which uses algorithms to obscure data. In other words, it encrypts the data.

The main application of cryptography is the ability to generate unique digital signatures. They are the digital equivalent of handwritten signatures but more inherent security. Digital signatures solve the problem of impersonation and tampering in practice. Each individual who wants to record a transaction in the blockchain stamps the transactional data with their signature, which can be verified by everyone else. These signatures are based on asymmetric cryptographic keys

where one key is public and the other is private. This relates to the second aspect of cryptocurrencies which is Wallets.

Wallets are the objects that are responsible for securely keeping the public and private keys for the individual. Similar to a physical wallet, they can track how much currency is entitled to you, your balance in the cryptocurrency, your net worth in the overall system. The public key is used as an address for your wallet. Other individuals use it to send cryptocurrency to you at that specific address. The stored private key generates the signatures that make transactions official.

Finally, the third aspect of cryptocurrencies is mining. Mining is the work of adding transactions to the blockchain. When people submit transactions to the cryptocurrency network, these transactions will join the transaction pool. When submitted, the transactions will temporarily be unconfirmed. The miners will then take a group of unconfirmed transactions and use them as the data to be officially recorded within a new block in the chain. However, to gain the right to add a new block, that miner will have to solve a computational puzzle called the proof of work algorithm.

The proof of Work is difficult to solve because it involves finding a hash that contains a specific pattern. There is a very low probability of solving it randomly, so many trial and error are required to find the answer to the proof of work algorithm. It is time-consuming and computationally expensive, but once a miner solves a proof of work algorithm, they then gain the rights to create a block consisting of those transactions. They can submit the new block containing the transactions to the blockchain network by solving the proof of work. Other miners will then recognize the solution because proof of work is easily verified once another miner presents the solution for you. After verification, the miner can form an agreement called consensus in the network to include the new block into the chain. This proof of work is also called a consensus algorithm.

The proof of work requires computation power of computer; hence a lot of energy is required to mine the transactions. So, as a reward for doing the task of mining, the miner will receive some cryptocurrency. Therefore, miners are continually trying to add new blocks in the hopes of

gaining this currency reward which provides an overall equilibrium to the system, where there is momentum for transactions to be added to the block.

Private key regeneration

For private key regeneration, I will be using a code library called ‘CADO-NFS’, a Number Field Sieve (NFS) algorithm for factoring integers[86], to factor the modulus of 256-bit RSA key and generate its prime factors. Then using a python script[87] and OpenSSL library to generate a private key from the factored prime integers. Furthermore, I have referred to the ‘Cracking 256-bit RSA’ [88] guide by Ray Doyle.

As discussed in earlier chapters that public-key cryptography with smaller key sizes is not secure. The RSA algorithm primarily uses 2 large prime numbers to compute the private and public keys. I have intentionally used the RSA 256-bit key to generate a public and private key for the wallets used in the BearCoin cryptocurrency. When a user wants to do a transaction, he would need his private key, recipient's public key address and amount to be transferred. The private key assigned to the user is used to sign the transactions and submit them to the transactional pool to be verified and confirmed. The public key for the user is used to verify the digitally signed signature of the transaction, and when the transaction is confirmed, the funds are transferred to the recipient's wallet. Only the sender knows the private key, and it should be computationally infeasible to generate a private key from the public key. However, since we used RSA 256bit key size, generating a private key from its public key is fairly easy. I was able to compute this key in less than 2 minutes using an Azure cloud Instance with 16 core CPU. Another user then uses the private key to transfer funds to themselves. These unauthorized transactions will still be valid and can be verified by the network as its signed digital signature will be the same as that if signed by the authorized private key. Once the blockchain network fully confirms the transaction, then the funds will be permanently transferred to the attacker's cryptocurrency wallet.

Why does it work?

RSA is based on the fact that it is very hard to break an Integer by factoring it but very easy to calculate the Integer from its factors. It works like a trapdoor on a pit; very easy to fall into one but very hard to climb up top. Multiplication of large numbers is pretty easy by humans or computers, but reversing the multiplication or factoring is apparently very hard, even for computers. [89]

For example, if you want to multiple :

$$17477852958781876547 \times 15241555427044345769 = ?$$

It would take minutes for you or even milliseconds for a computer to calculate this and show the result.

$$17477852958781876547 \times 15241555427044345769 = 266389664617004986624097978187739779643$$

But if we see the problem in another way, can you calculate two integers that would result in this integer when multiplied

$$? \times ? = 281512008712700373730275954373439628511$$

I can say with high confidence that it would take you more days just to find an approach to solving this.

$$17477852958781876547 \times 15241555427044345769 = 266389664617004986624097978187739779643$$

However, it would take only a minute to verify the result once solved. The amount of work needed to calculate these factors is directly proportional to the size of the final integer.

RSA algorithm generates a key pair made of a private key and a public key. The private key can be regarded as two factors (p,q), while the public key can be regarded as their product (n)

$$p \times q = n$$

So, it is easier to find a public key from a private key but relatively harder the other way around, and they get even more challenging as the size of n increases.

Digital signatures rely on a key pair (public and private key) to sign the data and provide a digitally signed signature. The digital signature algorithm takes the key pair's private key along with data to create this signed signature. This digital signature algorithm can also take the signed signature, the original data and the public key of the key pair to verify this signed signature. This verification validates that indeed the owner of the private key has signed the data, and data had not been tampered with on the way.

How does it work?

I am intentionally using RSA 256 bit key to generate key pairs for users in BearCoin to demonstrate the breaking of the RSA private key of a smaller size. When the user starts the application, a key pair is generated. The public key is used as the wallet's address, and anybody with the public key can send funds to the user's account. However, only the authorized user can send funds to another user's account by authorizing it with a private key. When the user sends funds to another user, its public key is shared and publicly available in the network. The application also has a feature to view the public addresses used to submit the transaction.

When the user starts the application, they can see its public key and its current balance. The User can view blocks of the blockchain and view unverified transactions in the transaction pool. The user can conduct transactions by entering the amount and recipient's public key address and user's private key issues to them.

Here in figure 33, the URL 192.168.1.22:3000 represents the victim's wallet. The public key address of this wallet is visible in the white text area. The public key address is shared with

others to receive the funds on this account.

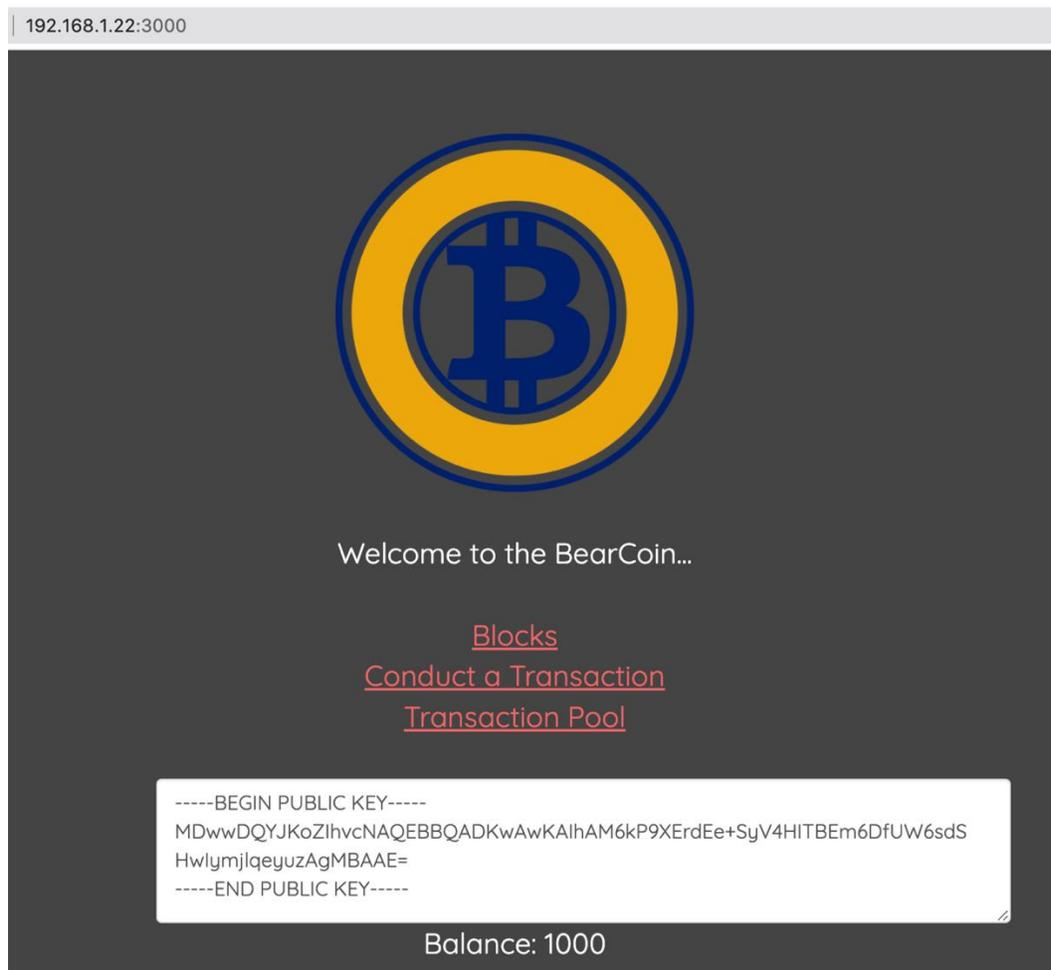


Figure 33Victim's wallet before the attack

Figure 34 displays the public key address of the attacker's wallet on URL 192.168.22:3786 that he will use to receive funds.

Currently, both Victim and attacker have 1000 BearCoins each. The Victim shared his wallets' public key address with the attacker to receive funds. But attacker will attack the victims' public key address to generate the victim's private address.

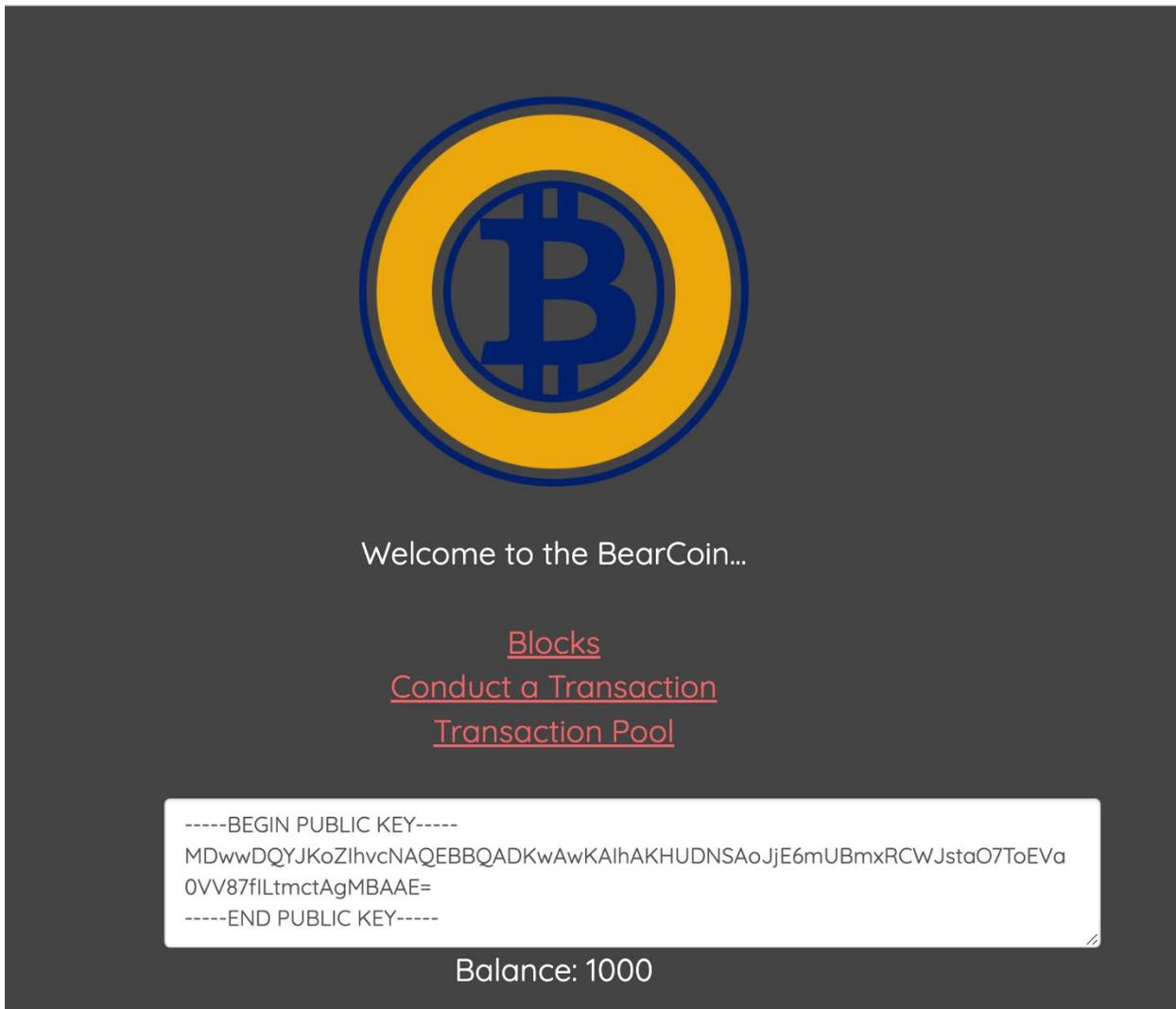


Figure 34 Attackers Wallet before the attack

Generating the private key from the public key

Step 1: Save the shared public key in a text file.

You can view the public key by command

```
# cat public.txt
```

```
gurpreet — -zsh — 80x24
Last login: Sat Feb 19 13:24:48 on ttys000
gurpreet@Gurpreets-MacBook-Pro ~ % cat public.txt
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAM6kP9XErdEe+SyV4HITBEm6DfUW6sdS
HwIymj1qeyuzAgMBAAE=
-----END PUBLIC KEY-----
gurpreet@Gurpreets-MacBook-Pro ~ %
```

Figure 35 View public key

Step 2: Use the following command to calculate the modulus of the public key. Modulus is the large integer value of n .

```
# openssl rsa -pubin -in public.txt -modulus -noout
```

We need to convert this modulus from hex value to a decimal value using command

```
# echo "ibase=16; <modulus> " | bc
```

```
gurpreet — -zsh — 85x24
gurpreet@Gurpreets-MacBook-Pro ~ % openssl rsa -pubin -in public.txt -modulus -noout ]
Modulus=CEA43FD5C4ADD11EF92C95E072130449BA0DF516EAC7521F02329A396A7B2BB3
gurpreet@Gurpreets-MacBook-Pro ~ % echo "ibase=16; CEA43FD5C4ADD11EF92C95E072130449BA
0DF516EAC7521F02329A396A7B2BB3" | bc
93466650299977732227176708060770226057067789019779220295932230024117\
746346931
gurpreet@Gurpreets-MacBook-Pro ~ %
```

Figure 36 generating modulus of public key

Step 3: To calculate its prime factors, provide this modulus to the *cado-nfs* tool.

```
# ./cado-nfs.py <modulus in decimal>
```

```

azureuser@rsa16core: ~/Downloads/cado-nfs-master
File Edit View Search Terminal Help
azureuser@rsa16core:~/Downloads/cado-nfs-master$ ./cado-nfs.py 93466650299977732
227176708060770226057067789019779220295932230024117746346931
Info:root: Using default parameter file ./parameters/factor/params.c75
Info:root: No database exists yet
Info:root: Created temporary directory /tmp/cado.c_y4xw2q
Info:Database: Opened connection to database /tmp/cado.c_y4xw2q/c75.db
Info:root: Set tasks.linalg.bwc.threads=8 based on detected physical cores
Info:root: Set tasks.threads=16 based on detected logical cpus
Info:root: tasks.threads = 16 [via tasks.threads]

```

Figure 37 provides modulus to cado-nfs program

Step 4: The 'cado-nfs' tool factorise the modulus(n) of the key to two prime numbers we can refer these values a p and q .

$$p \times q = n$$

312648365854156147254672379506487490683 \times 298951347609403301810122921209582166057
= 93466650299977732227176708060770226057067789019779220295932230024117746346931

We can easily verify the result using multiplication.

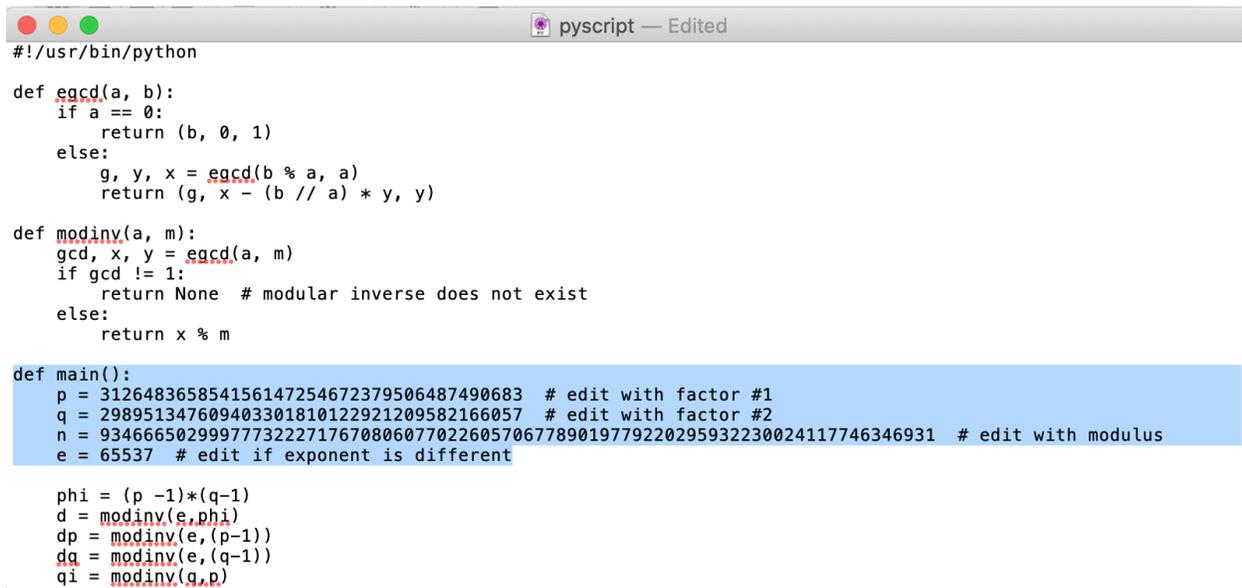
```

azureuser@rsa16core: ~/Downloads/cado-nfs-master
File Edit View Search Terminal Help
: 1.4/0.797281
Info:Filtering - Duplicate Removal, splitting pass: Aggregate statistics:
Info:Filtering - Duplicate Removal, splitting pass: CPU time for dup1: 0.6s
Info:HTTP server: Shutting down HTTP server
Info:Complete Factorization / Discrete logarithm: Total cpu/elapsed time for entire factorization: 479.74/67.4338
Info:root: Cleaning up computation data in /tmp/cado.c_y4xw2q
312648365854156147254672379506487490683 298951347609403301810122921209582166057
azureuser@rsa16core:~/Downloads/cado-nfs-master$
azureuser@rsa16core:~/Downloads/cado-nfs-master$

```

Figure 38 Cado-nfs results factored prime numbers

Now we only have to build the private key from its components. To do this, I am using a python script that generates a config file using the values of components p,q, and n.



```
#!/usr/bin/python
def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)

def modinv(a, m):
    gcd, x, y = egcd(a, m)
    if gcd != 1:
        return None # modular inverse does not exist
    else:
        return x % m

def main():
    p = 312648365854156147254672379506487490683 # edit with factor #1
    q = 298951347609403301810122921209582166057 # edit with factor #2
    n = 93466650299977732227176708060770226057067789019779220295932230024117746346931 # edit with modulus
    e = 65537 # edit if exponent is different

    phi = (p - 1)*(q - 1)
    d = modinv(e, phi)
    dp = modinv(e, (p - 1))
    dq = modinv(e, (q - 1))
    qi = modinv(q, p)
```

Figure 39 Script to generate config files

Step 5: Edit the script and enter the values of p,q and n that are calculated in the previous step.

Running this script using command

```
# pyhton3 pyscript.py
```

It creates a file name 'gen.config', which will be used in the next step.

```
azureuser@rsa16core: ~/Downloads
File Edit View Search Terminal Help
azureuser@rsa16core:~/Downloads$ python3 pyscript.py
asn1=SEQUENCE:rsa_key

[rsa_key]
version=INTEGER:0
modulus=INTEGER:93466650299977732227176708060770226057067789019779220295932230024117746346931
pubExp=INTEGER:65537
privExp=INTEGER:39540456224222692921532481361442460250167811290074335062322041407240360197073
p=INTEGER:312648365854156147254672379506487490683
q=INTEGER:298951347609403301810122921209582166057
e1=INTEGER:210625112544468743724019107185115723959
e2=INTEGER:184428672890058820139540714824062995801
coeff=INTEGER:226148115741017849104954414009173990251
azureuser@rsa16core:~/Downloads$
```

Figure 40 view config file

The OpenSSL library can be used to generate the private key from 'gen.config' file.

Step 6: use the next command to generate the private key file.

```
# openssl asn1parse -genconf gen.conf -out newkey.der
```

It creates a certificate file with the file name 'newkey.der'

Now, 'newkey.der' file is the certificate file that has the private key we needed.

Use the below command to view the private key.

```
# openssl rsa -in newkey.der -inform der -text -check
```

Step 7: Copy the private key from the terminal.

```

gurpreet — -zsh — 90x14
65:0a:25:0a:ed:1e:94:37:a5:5c:e9:dd:0d:21:ac:
15
coefficient:
6d:48:e6:e2:37:9b:0c:58:f6:b7:ce:dd:7c:44:1d:
cb
RSA key ok
writing RSA key
-----BEGIN RSA PRIVATE KEY-----
MIGpAgEAAiEAsZ0Md6RdKo/Zue7kK+vmzg8K+Itf9SmYLS1SJXQSpQcCAwEAAQIg
OYL19400W0TLJoaxQXuYd1SIY0QN+97UIyDMjgZAOXkCEQDPeOo6gOBZ679D/cOY
a3KjAhEA2yg8tPWd6liNXJ/gvpbOTQIQ02ErAKWEEJhlfIszoPsXqwIQZQo1Cu0e
lDe1XOndDSGsFQIQbUjm4jebDFj2t87dfEQdyw==
-----END RSA PRIVATE KEY-----
gurpreet@Gurpreets-MacBook-Pro ~ % █

```

Figure 41 Private key is generated

Transfer funds

The application also provides an interface to do a nodeless transaction using the private key. Now that we have the private key of the victim's wallet, we can use this key to authorize transactions. We used the newly generated private key to send 1000 BearCoins to another wallet. The application allowed this transaction and provided the success message.

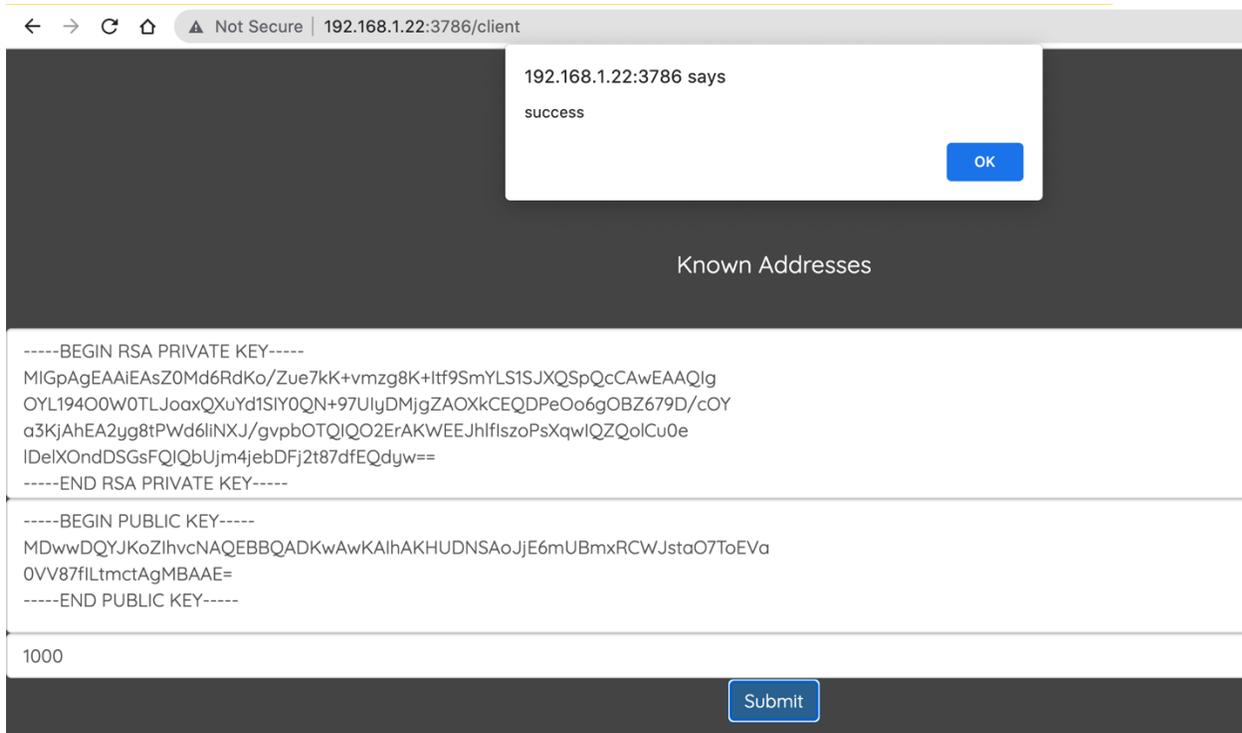


Figure 42 Application interface to conduct a transaction

The transaction then moves to the transaction pool, where multiple transactions can be added to the block. The miner then verifies the transaction in the block by analyzing the history of the blockchain and validating if the private key holder signed the transaction. Once validated, the miner does the proof of work to add this block to the blockchain.

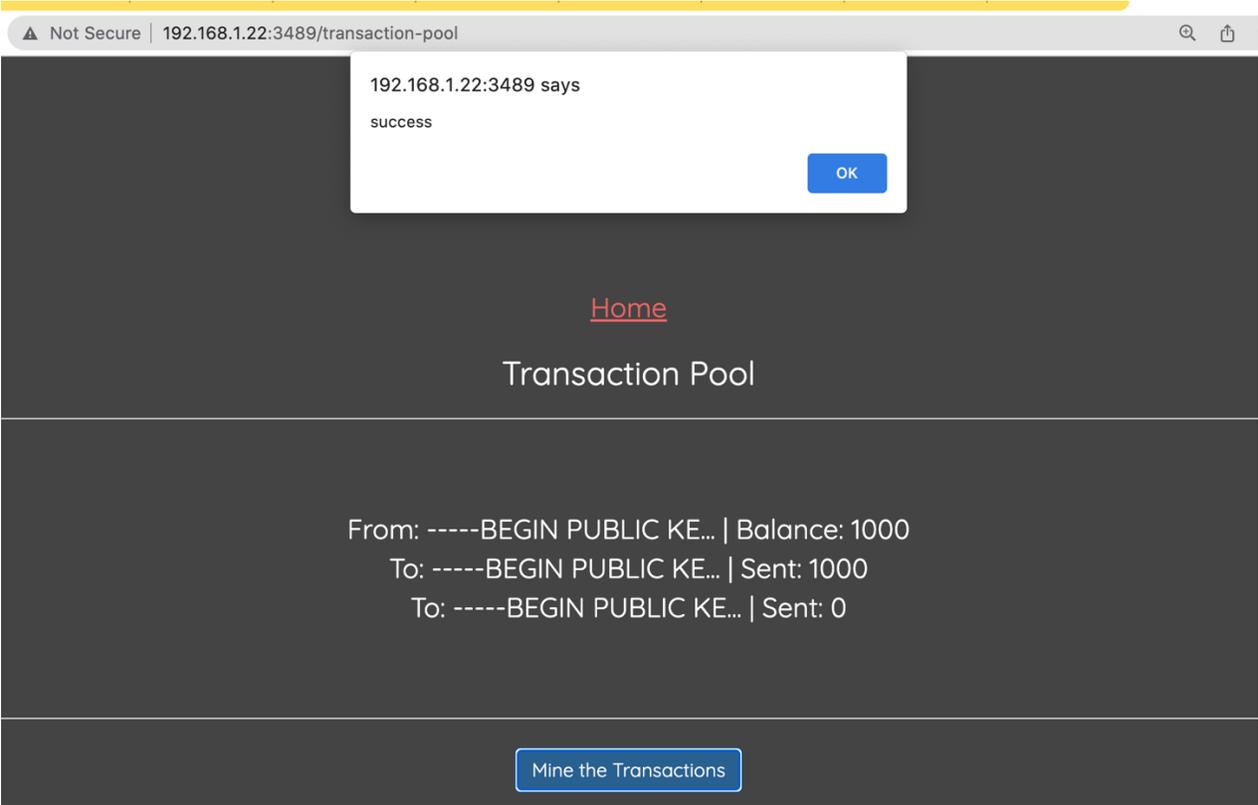


Figure 43 Application Interface to Mine transactions

Once the transaction is verified and added to the blockchain, the funds are transferred from the victim's wallet to the attacker's wallet. The transactions performed on the cryptocurrency platforms are final and irreversible. The application returns a success message after the block is mined.

Outcome

Since the attacker successfully regenerated the private key associated with the victim's public key, the attacker was able to do unauthorized transactions and steal the funds.

We have used RSA 256bit key size for this lab component, but both RSA and ECC are prone to such attacks, and it is just restricted by the reasonable amount of computing resources. However, quantum computers pose a huge risk to the contemporary cryptographic techniques. We will discuss some of these risks in the upcoming chapter.

Insecure Cryptographic hash functions:

MD5

The MD5 (Message Digest) algorithm is a cryptographically compromised but still widely used hash function that produces a 128- bits hash value. Ronald Rivest designed MD5 in 1991 to replace an earlier hash function MD4,[4] MD5 was initially designed to be used in cryptographic hash functions. However, extensive vulnerabilities have been discovered at a later point in time. Although it can still be used to verify data integrity but only for unintentional corruption, It can still be used for non-cryptographic purposes such as determining the partition of a key in a partitioned database.

As mentioned earlier, a fundamental requirement of any cryptographic hash function is that it must be very hard to find two messages with the same hash value. This requirement is proven to be invalidated in the case of MD5 because such collisions can be easily found using any ordinary home computer.

On 31 December 2008, the CMU Software Engineering Institute determined that MD5 was "cryptographically compromised and unsuitable to further use". Despite of its well-documented weakness, MD5 continues to be still widely used

The MD5 hash is susceptible to a chosen-prefix collision attack and is demonstrated in the lab implementation.

SHA1

The SHA1, acronym for Secure Hashing Algorithm, is not considered that secure anymore. It is still a widely used hash function that produces 160-bit output known as a message digest. SHA1 was designed by the United States National Security Agency and is a U.S Federal Information Processing Standard. The SHA1 is designed on similar principles to those used in MD5 but generates a larger hash value than MD5.

SHA-1 is one of many widely used in security protocols and applications such as TLS And SSL, PGP, SSH, and IPsec. SHA-1 and MD5 are both derived from MD4.

As of 2005, SHA1 is not considered secure enough and was finally deprecated to be used by NIST. All web browsers vendors stopped accepting the SHA1 in 2017 shortly after Google and CWI demonstrated a collision attack on SHA1 by publishing two different pdfs having the same SHA1 hash value. As of the year 2020, chosen-prefix attacks against SHA-1 are also practical

Table 2 Security benchmark for various hash algorithms [wikipedia]

Hash function	Security claim	Best attack	Publish date	Comment
MD5	2^{64}	2^{39}	2009-06-16	This attack takes hours on a regular PC.[90]
SHA-1	2^{80}	$2^{63.4}$	2020-01-08	Paper by Gaëtan Leurent and Thomas Peyrin[91]
SHA256	2^{128}	N/A	N/A	N/A
SHA-3	Up to 2^{512}	N/A	N/A	N/A

█ Attack demonstrated in practice — complexity is low enough to be used

Chosen-prefix collision on MD5

The MD5 hashing algorithm is cryptographically broken, and a chosen prefix can be calculated for two files that, when appended to these files, will result in the same hash with MD5.

I have two aerial view images of the University of Alberta campus with separate text on the top left corner to demonstrate this attack. Fig 47 and Fig 48 display different text in both images.

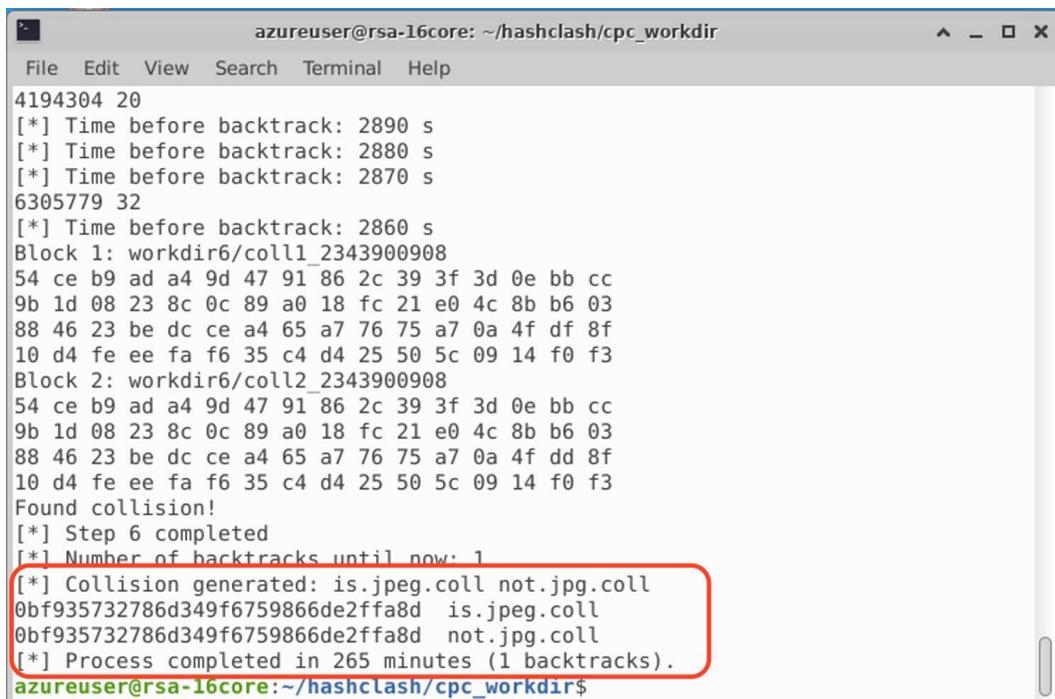
I used a tool called HashClash by Mark Stevens[92] to carry out this attack

After the installation of HashClash run `cpc.sh` script with absolute path of both files to start the attack.

```
./scripts/cpc.sh <prefix.filename1> <prefix.filename2>
```

The tool took 265 Minutes on a 16 core CPU to carry out the attack and generate identical MD5 hashes for both the files.

Fig 46 displays the output of the program after the attack.



```
azureuser@rsa-16core: ~/hashclash/cpc_workdir
File Edit View Search Terminal Help
4194304 20
[*] Time before backtrack: 2890 s
[*] Time before backtrack: 2880 s
[*] Time before backtrack: 2870 s
6305779 32
[*] Time before backtrack: 2860 s
Block 1: workdir6/coll1_2343900908
54 ce b9 ad a4 9d 47 91 86 2c 39 3f 3d 0e bb cc
9b 1d 08 23 8c 0c 89 a0 18 fc 21 e0 4c 8b b6 03
88 46 23 be dc ce a4 65 a7 76 75 a7 0a 4f df 8f
10 d4 fe ee fa f6 35 c4 d4 25 50 5c 09 14 f0 f3
Block 2: workdir6/coll2_2343900908
54 ce b9 ad a4 9d 47 91 86 2c 39 3f 3d 0e bb cc
9b 1d 08 23 8c 0c 89 a0 18 fc 21 e0 4c 8b b6 03
88 46 23 be dc ce a4 65 a7 76 75 a7 0a 4f dd 8f
10 d4 fe ee fa f6 35 c4 d4 25 50 5c 09 14 f0 f3
Found collision!
[*] Step 6 completed
[*] Number of backtracks until now: 1
[*] Collision generated: is.jpeg.coll not.jpeg.coll
0bf935732786d349f6759866de2ffa8d is.jpeg.coll
0bf935732786d349f6759866de2ffa8d not.jpeg.coll
[*] Process completed in 265 minutes (1 backtracks).
azureuser@rsa-16core:~/hashclash/cpc_workdir$
```

Figure 46 Generating files with identical MD5 hashes



Figure 47 MINT IS AWESOME



Figure 48 MINT IS NOT AWESOME

Fig 49 indicates the identical MD5 hashes of two separate images.



Figure 49 Terminal showing Identical MD5 hashes

Chosen-prefix collision on SHA1

SHA1 was no longer regarded as secure compared with other hashing algorithms such as SHA256 or SHA3 and NIST formally deprecated it in 2011. The brute force search attack on SHA1 is claimed to take up to 2^{80} operations, but in 2017, a team demonstrated a similar attack around 100000 times faster. In February 2017, Google and CWI demonstrated a first public Collision on SHA1, in which they generated two different pdf files with the same SHA1 hash code.

The attack required "the equivalent processing power of 6,500 years of single-CPU computations and 110 years of single-GPU computations"[93].

We do not have the resources available to perform such an attack, but here is an image directly from shattered.io to visualize that attack.

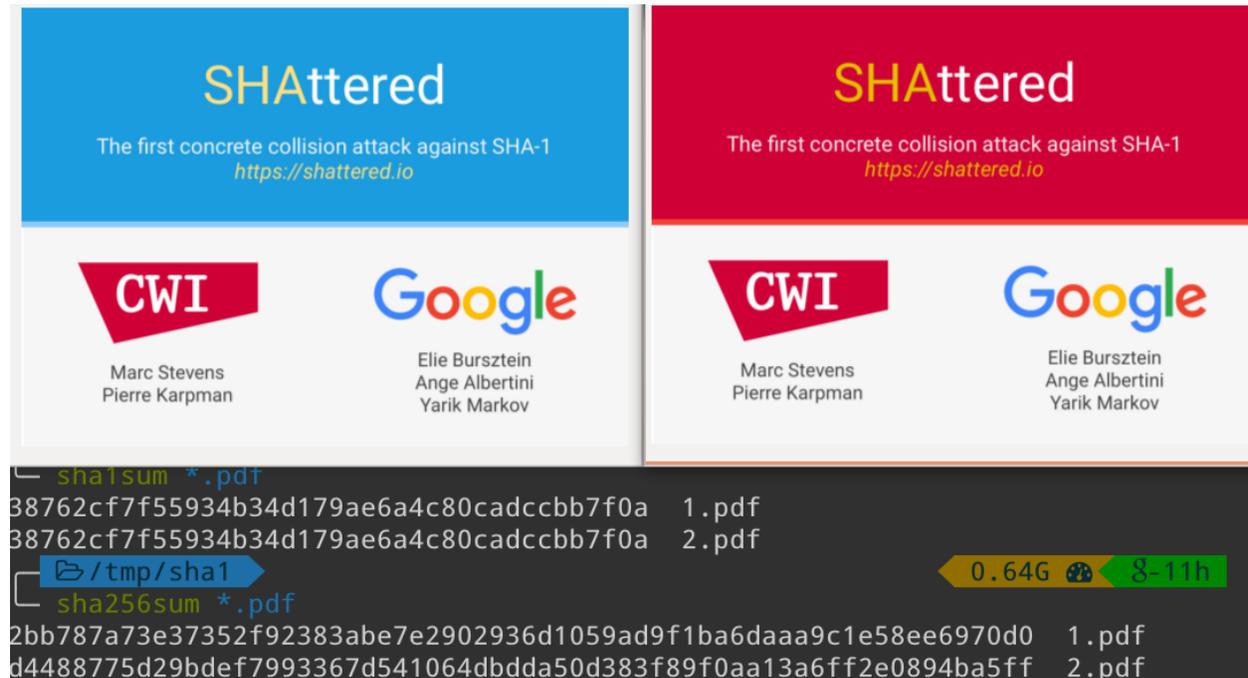


Figure 50 Image from Shattered.io depicting identical hashes for two pdfs

Implications of chosen-prefix attack:

The chosen prefix attack can be abused to authenticate corrupted files. The attacker can create two document files, say a rental agreement, one with the agreed-upon conditions and the other with favourable conditions. Then using tools like *HashClash* to create identical hashes for both the files, the attacker can get the original file digitally signed by the tenant. Then replace the original file with the corrupted file to enforce the agreement. Since the hash of both files is the same, the signature would still be valid for both the files. This type of attack can be virtually extended to any type of file, even for computer programs. Therefore, it is always recommended to use more recent hash algorithms like SHA3 to digitally sign any files.

What is a Quantum Computer?

Quantum computing is a computer that performs calculations by combining the collective properties of quantum states, such as superposition, interference, & entanglement.

Supercomputers are machines with the ability to perform quantum calculations. Even though existing quantum computers are too small to beat traditional (classical) computers for practical purposes, they are believed to be capable of tackling some computational tasks far quicker than regular computers, such as factorization (which underpins RSA encryption). The discipline of quantum computing is included in quantum information science. Quantum computers are data storage & processing devices that employ quantum physics principles. This can be highly advantageous for some workloads, as they can easily outperform most advanced computers.

Who invented the Quantum Computer?

Paul Benioff, a scientist, developed a quantum theory model of the Turing machine in 1980, which ushered in quantum computing. Yuri Manin and Richard Feynman later claimed that a quantum computer could replicate things that a classical computer could not. In 1994, Peter Shor developed a quantum method for factoring integers that may be used to decrypt RSA-encrypted communications. The first two-qubit quantum computer was created in 1998 by Isaac Chuang, Neil Gershenfeld, and Mark Kubinec. Most researchers believe that "fault-tolerant quantum computing is still a long way off" despite steady experimental advances since the late 1990s. In recent years, both the public and private sectors have raised their investments in quantum computing research. On October 23, 2019, Google AI claimed to have completed a quantum computation that was infeasible on any conventional computer in partnership with the US NASA. However, the validity of this claim is still being investigated. Data is stored in binary "bits," which can be either 0s or 1s, in traditional computers like smartphones and laptops. In a quantum computer, a quantum bit is the fundamental memory unit.

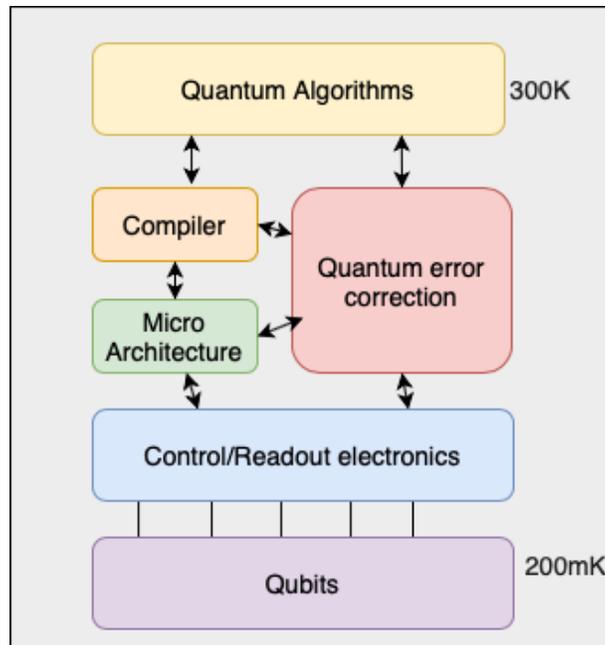


Figure 51 Quantum Architecture

Quantum Supremacy’s Potential Impact on Cryptocurrencies

Despite the technological milestones that blockchain technologies and cryptocurrencies had also achieved, fundamental technical loopholes stay open to exploitation by future technologies, such as quantum computers. Headlines about cryptocurrency make life simple to forget that, despite the dramatic price swings and regulatory uncertainties and technological benchmarks that blockchains and cryptocurrencies have achieved. While quantum computers will not entirely abolish blockchains & cryptocurrencies, it is believed that they will pose significant security risks to the whole sector.^[94]

Can It break any cryptocurrency?

To crack the same algorithm that protects bitcoin, quantum computers had to grow to a million times their current size, putting the cryptocurrency at risk of being hacked.

The cryptographic algorithm SHA-256, created by the US National Security Agency, is used by miners or devices that keep the bitcoin network secure. Ordinary computers will not be able to crack this code, but quantum computers, which can use quantum physics' features to speed up specific tasks, could theoretically be able to.^[95]

Almost all cryptocurrencies rely on public-key cryptography in one way or another, and Quantum computers possess a high risk for breaking these keys and rendering the current cryptographic technology useless. The risk exposes almost all the cryptocurrencies that are not using any post-quantum cryptography.

Shor's Factor Algorithm

Shor's algorithm is a quantum algorithm of factoring a number N in $O((\log N)^3)$ time and $O(\log N)$ space. The algorithm was significant because it implies that public-key encryption may be easily broken with a suitably powerful quantum computer. In RSA, for example, the public key N is the sum of two large prime numbers. One method of defeating RSA encryption is to factor N ; however, factoring becomes increasingly time-consuming as N grows larger; more specifically, no standard algorithm exists that really can factor in time $O((\log N)^k)$ for any k . On the other hand, Shor's algorithm can crack RSA in polynomial time. It has also been used to crack several different public key encryptions.^[96]

Shor's method is probabilistic, like all quantum computer algorithms: it produces the correct answer with a high probability, as well as the probability of failure can be reduced by repeating the algorithm. A group at IBM demonstrated Shor's technique in 2001, factoring 15 into 3 and 5 using a quantum computer with 7 qubits.^[97]

Quantum computers and cryptography

A private-public key pair is formed in asymmetric cryptography in such a way that the two keys have such a mathematical relationship. The private key is kept private, whereas the public key is made available to the public, as the name implies. Individuals can use their private key to create a digital signature that can be validated by anybody with the associated public key. This approach is often used in the financial sector to confirm transaction legitimacy and integrity.

Asymmetric cryptography's security is primarily based on a mathematical principle known as a "one-way function." The public key may be easily inferred from the private key, but not the other way around. Most algorithms for obtaining the private key from the public key take an immense amount of time/work to complete and are hence considered unfeasible. However, mathematician

Peter Shor devised a quantum algorithm that can break the security assumption of the most used asymmetric cryptography techniques. This means that anyone with a powerful enough quantum computer might use this algorithm to generate a private key from a known public key, therefore can falsifying any digital signature.^[98] [99]

Bitcoin is a value-transfer mechanism that is decentralized. In contrast to the banking system, where a bank is responsible for providing clients with a bank account, a Bitcoin user creates his own (random) address. The user's computer produces a random Bitcoin address (associated with the public key) as well as a secret (private key) that is necessary to make transactions from this address using a simple technique.

The impact of quantum computing on cryptography

Quantum computing has the potential to revolutionize everything. It is possible to employ something like Shor's algorithm, which uses quantum physics to solve the problem of integer factorization (finding prime factors for an integer N) or another hypothesis like the discrete logarithm problem. Many asymmetric cryptographic techniques, like RSA, are built on the notion that big integer factorization is computationally impossible. Ironically, symmetric algorithms (asymmetric predecessors that do not protect electronic transactions because they have a single key) like AES could nevertheless be regarded safe if they utilize a sufficiently large key (i.e., AES 256 or higher).

It is possible that quantum computing is now a reality, but it is probably too early to be concerned. Essentially, the quantum computer power required to break current asymmetric algorithms will remain prohibitively expensive, limiting its use to governments, particularly those interested in prying into other nations' secrets.^[100]

Grover's algorithm

Lov Grover created an algorithm to search unsorted databases based on quantum computers^[101]. The algorithm can search a particular entry in an unsorted database of N entries in \sqrt{N} searches, whereas a standard computer would need $N/2$ searches. Bone and Castro have described the impact of a possible application of Grover's algorithm to break the Data Encryption Standard (DES).^[102] The authors described that the algorithm only needed 185 searches to find the key in

56 key-sized DES. So, the difficulty of cracking the password can be increased exponentially by increasing the key size. Grover's algorithms have some potential applications to symmetric cryptosystems, but they are not considered effective as Shor's algorithm.^[103]

Quantum computer effects on hashing

A huge random size input is transformed into a small fixed size output by hash algorithms. A digest or hash value is the result of the hash algorithm's calculations. The hash algorithms do not require any cryptographic keys to operate and operate in a secure one-way manner. The one-way procedure means that computing input data from output data is cryptographically and technically impossible.

The hash functions also suffer from a similar issue as symmetric ciphers since their security mainly depends on a fixed output length. Grover's algorithm can be used to theoretically find a collision in a hash function in square root number of steps of its original length, just like searching an unsorted database. Furthermore, it has been proved that it is also possible to combine the Birthday paradox with Grover's algorithm, described as a quantum birthday attack^[104]. It would be possible to generate a table of size $\sqrt[3]{N}$ and utilize Grover's algorithm to find a collision. This implies that to have a b – bit security level against Grover's quantum algorithm, a hash function should provide at least a $3b$ – bit output. As a result, many of the present hash algorithms are considered non-secure for use in the quantum era. Surprisingly, both SHA-3 and SHA-1, which have longer outputs, remain quantum resistant.^[105]

POST-QUANTUM CRYPTOGRAPHY

Post-quantum cryptography, often known as quantum encryption, is the development of cryptographic systems for classical computers that can withstand vulnerabilities suspected by quantum computers.

During the 1980s, scientists hypothesized that if computers could take advantage of quantum mechanics' unique properties, they could do complicated computations far faster than regular binary computers. It was quickly obvious that a quantum computer could execute some types of challenging computations in a matter of hours by utilizing quantum properties such as quantum entanglement, whereas a computer program would take several years to complete the same calculation.

In the 1990s, all over the world began to investigate what a post-quantum cryptography system might look like when mathematician Peter Shor revealed that a theoretical quantum computer could easily defeat the approach used for public-key encryption (PKE). As of this writing, standards for implementing post-quantum cryptography are still being established.^[106]

Cryptography: pre-quantum, quantum, and post-quantum

Quantum computers use quantum physics to process data in quantum bits (qubits). Because each qubit can be a mix of 0s and 1s, a quantum computer can handle variables exponentially faster than a regular binary computer. Pre-quantum cryptography uses a type of encryption known as an algorithm to turn human involvement into secret code. Pre-quantum cryptography aims to construct encryption cyphers that are easy to understand but difficult to crack. Quantum cryptography, on the other hand, uses geometric cyphers to turn human-readable data into unbreakable secret code. One of the most challenging aspects of post-quantum cryptography is that quantum physics is a new field of study, and quantum computer prototypes are exceedingly costly to build and operate.

Challenges Associated with Post-Quantum Cryptography

The requirements of post-quantum cryptography must be accommodated in new application implementations, and new schemes must be able to adapt to them. In reality, post-quantum

cryptography requirements may shape certain future application standards. Algorithm replacement typically necessitates the modification or replacement of cryptographic libraries, implementation verification tools, hardware that implements or picks up speed algorithm performance, relying upon operating system & application code, communications devices and protocols, as well as user and administrative procedures. Security standards, procedures, & best practice documentation, as well as install, configuration, and administration documents, all need to be updated or replaced. When deciding to replace an algorithm, it is vital to consider how the algorithm will be handled in the future.

Security in a post-quantum world

Quantum computing could be destructive to Blockchain networks because its security relies on encryption techniques like ECC. Breaking ECC in Blockchains using quantum computing could break keys that keep cryptocurrency safe. Hence, attackers have higher incentives to target cryptocurrency Blockchain networks, as they have more economic value.

There are various Blockchains such as Quantum-Resistant Ledger (QRL) and IOTA (an open-source distributed ledger and cryptocurrency designed for the Internet of Things) that were developed using Post-Quantum Cryptography (PQC). However, the popular Blockchain applications still use ECC, which is currently not quantum-safe. To keep the quantum threat at bay, we'll need an industry-wide update of Blockchains, very similar to NIST's work to develop a quantum-proof encryption solution for the Internet.

The following are some of the Blockchain approaches that are being investigated as some of the strategies that can improve the security of Blockchains in the post-quantum world. [107]

- eXtended Merkle Signature Scheme (XMSS)
- Blockchain Post-Quantum Signatures (BPQS)
- Winternitz One-Time Signatures (W-OTA), as used by IOTA

Conclusion

Public key encryption algorithms like RSA and Elliptic Curve presently protect the Cryptocurrencies transactional integrity. These algorithms also revolutionized information security and have benefited the entire globe by enabling access to digital commerce, secure communications, and distant financial services. Unfortunately, a new technological risk has evolved, much like the cryptanalysts of World War II built early computers to overcome Enigma. Similarly, current public-key encryption also can be theoretically broken by quantum computers. The threat of quantum code-breaking is so serious that the National Institute of Standards and Technology (NIST) in the United States has begun developing the next generation of encryption called Post Quantum Cryptography. The goal should be to move away from theoretically insecure encryption algorithms like RSA and toward methods that have been demonstrated computationally difficult to crack.

Blockchain is not the only platform on a direct collision course with quantum computing, but nearly the entire Internet is at risk. Quantum computing is a way more serious threat, not only to Blockchain applications but to all the platforms using RSA and ECC. However, there are rays of hope in the form of post-quantum cryptography for both the Blockchain and the Internet.

I must admit that this report mostly touches on the technical aspect of Blockchain and cryptography. However, I tried to simplify the context as much as possible. I have covered concepts of Blockchain, its architecture, its applications, Hashing principles, public-key cryptography, consensus, and mining of cryptocurrencies, as well as a quantum computer and their implications on these platforms. The lab component covers the breaking of RSA and MD5 cryptography, though these are not used in real-world cryptocurrencies. However, the ECC and SHA256 cryptographical functions, which are used in some major cryptocurrency implementations, are at risk by Quantum Computing.

At last, despite the risks from quantum computers, there are few solutions that the Blockchain and Cryptocurrency platform can and must adapt to stay secure. The Blockchain industry needs to evolve to remain relevant in the post-quantum era. The industry needs to move forward towards quantum-proofing before the theoretical threats become practical.

Referred books:

Oriyano, Sean-Philip. *Cryptography: Infosec pro Guide*. McGraw-Hill Professional, 2013.

Drescher, Daniel. *Blockchain Basics: A Non-Technical Introduction in 25 Steps*. Apress, 2017.

Wong, David. *Real-World Cryptography*. Manning Publications, 2021.

Das, Abhijit, and Madhavan C E Veni. *Public-Key Cryptography: Theory and Practice*. Pearson Education India, 2009.

Raj, Koshik. *Foundations of Blockchain: The Pathway to Cryptocurrencies and Decentralized Blockchain Applications*. Packt Publishing Limited, 2019.

Bashir, Imran. *Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Expained*. Packt, 2018.

Krishnakumar, Arunkumar, and David F. Beach. *Quantum Computing and Blockchain in Business: Exploring the Applications, Challenges, and Collision of Quantum Computing and Blockchain*. Packt Publishing Ltd., 2020.

References:

[1] Nakamoto, S., & Bitcoin, A. (2008). A peer-to-peer electronic cash system. *Bitcoin*. –URL:<https://bitcoin.org/bitcoin.pdf>

[2] (Internet), <https://www.valuewalk.com/cryptocurrencies-need-to-improve-stability-and-scalability-to-see-mass-scale-adoption-says-globaldata/>

[3] (Internet) <https://www.analyticsinsight.net/blockchain-is-being-used-in-various-sectors-of-our-life/>

[4] (Internet) <https://blockgeeks.com/guides/what-is-cryptocurrency/>

[5] (Internet) <https://www.nerdwallet.com/article/investing/cryptocurrency-7-things-to-know>

[6] (Internet) <https://investors-corner.bnpparibas-am.com/markets/what-is-the-problem-with-cryptocurrency-bitcoin/>

[7] (Internet) <https://www.investopedia.com/terms/b/blockchain.asp>

[8] (Internet) <https://en.wikipedia.org/wiki/Ledger>

[9] (Internet) <https://en.wikipedia.org/wiki/Cryptocurrency#Wallets>

[10] (Internet) https://en.wikipedia.org/wiki/Financial_transaction

[11] (Internet) <https://www.computerworld.com/article/2588287/networking-peer-to-peer-network.html>

-
- [12] (Internet) <https://www.geeksforgeeks.org/consensus-algorithms-in-blockchain>
- [13] (Internet) <https://thecrypto.app/blog/the-main-types-of-consensus-algorithms/>
- [14] (Internet) <https://www.investopedia.com/tech/explaining-crypto-cryptocurrency/>
- [15] (Internet) <https://www.investopedia.com/tech/explaining-crypto-cryptocurrency>
- [16] (Internet) <https://hackernoon.com/cryptographic-hashing-c25da23609c3>
- [17] (Internet) <https://mkyong.com/java/java-digital-signatures-example>
- [18] (Internet) <https://en.wikipedia.org/wiki/Blockchain>
- [19] Marie Jeanne Tuyisenge, *Blockchain Technology Security Concerns*, 2021
- [20] Chinaka, M. (2016). *Blockchain technology--applications in improving financial inclusion in developing economies: case study for small scale agriculture in Africa* (Doctoral dissertation, Massachusetts Institute of Technology).
- [21] (Internet) <https://en.wikipedia.org/wiki/Blockchain>
- [22] Nakamoto, S., & Bitcoin, A. (2008). *A peer-to-peer electronic cash system. Bitcoin.* –URL:<https://bitcoin.org/bitcoin.pdf>
- [23] Hougan, M., & Lawant, D. (2021). *Cryptoassets: The Guide to Bitcoin, Blockchain, and Cryptocurrency for Investment Professionals. CFA Institute Research Foundation Briefs, January*
- [24] Rui Zhang And Rui Xue, *Security and Privacy on Blockchain 2019*
- [25] *A Survey on Ethereum Systems Security: Vulnerabilities, Attacks and Defenses*, Huashan Chen et al, 2019
- [26] *A Survey on Security and Privacy Issues of Bitcoin* by Mauro Conti et al., 2018
- [27] *A survey on the security of blockchain systems* by Xiaoqi Li et al, 2017
- [28] Mosakheil, J. H. (2018). *Security threats classification in blockchains.*
- [29] *The Security Reference Architecture for Blockchains: Toward a Standardized Model for Studying Vulnerabilities, Threats, and Defenses* by Ivan Homoliak et al., 2021
- [30] *A Survey on Security and Privacy Issues of Bitcoin* by Mauro Conti et al., 2018
- [31] *A survey of Blockchain Security Issues and Challenges* by Iuon-Chang Lin et all, 2017
- [32] Natoli, C., & Gramoli, V. (2017, June). *The balance attack or why forkable blockchains are ill-suited for consortium.*
- [33] Wani, S., (2021) *Distributed Denial of Service (DDoS) Mitigation Using Blockchain—A Comprehensive Insight.*
- [34] (Internet) <https://mathworld.wolfram.com/HashFunction.html>
- [35] (Internet) <https://mariuszprzydatek.com/category/others/>
- [36] (Internet) <https://www.coursehero.com/file/95055352/Module2pptx/>
- [37] (Internet) https://en.wikipedia.org/wiki/Hash_function

-
- [38] (Internet) <https://www.synopsys.com/blogs/software-security/cryptographic-hash-functions/>
- [39] Rogaway, Phillip, and Thomas Shrimpton. *Cryptographic hash-function basics*:
- [40] Tsudik, Gene. *Message authentication with one-way hash functions*.
- [41] Cormen, Thomas H. *Introduction to algorithms* (3rd ed.).
- [42] Merkle, Ralph C. *Protocols for Public Key Cryptosystems*.
- [43] Back, Adam. *Hashcash—a denial of service counter-measure*. 2002
- [44] (Internet) https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm
- [45] (Internet) https://en.wikipedia.org/wiki/Security_of_cryptographic_hash_functions
- [46] (Internet) <https://privacycanada.net/hash-functions/hash-collision-attack/>
- [47] (Internet) <https://www.docdroid.net/kBDmpYH/infytq-data-structures-using-python-part-2-pdf>
- [48] (Internet) <https://www.sparknotes.com/cs/searching/hashtables/section2/>
- [49] (Internet) <https://bitcointalk.org/index.php?topic=4521362>
- [50] (Internet) https://en.wikipedia.org/wiki/Preimage_attack
- [51] Elena Andreeva, Charles Bouillaguet, Orr Dunkelman, Pierre-Alain Fouque, Jonathan Hoch, et al. *New Second-Preimage Attacks on Hash Functions*. *Journal of Cryptology*, Springer Verlag, 2016, 29 (4), pp.657 - 696. [ff10.1007/s00145-015-9206-4](https://doi.org/10.1007/s00145-015-9206-4). [ffhal-01654410f](https://doi.org/10.1007/s00145-015-9206-4)
- [52] (Internet) <https://privacycanada.net/hash-functions/hash-collision-attack/>
- [53] Video, <https://www.youtube.com/watch?v=7cFIG04DsiE>
- [54] (Internet) <https://www.sciencedirect.com/topics/computer-science/asymmetric-cryptography>
- [55] (Internet) https://en.wikipedia.org/wiki/Public-key_cryptography
- [56] See Van Tilborg, Henk, and Sushil Jajodia, eds. *Encyclopedia of cryptography and security*. New York: Springer Science & Business Media, 2014.
- [57] (Internet) <https://www.dlib.org/dlib/september97/ibm/09lotspiech.html>
- [58] (Internet) <https://hackernoon.com/asymmetric-cryptography-in-blockchains-d1a4c1654a71>
- [59] (Internet) <https://www.coinbase.com/cloud/discover/dev-foundations/digital-signatures>
- [60] Video, <https://www.coursera.org/lecture/cryptocurrency/digital-signatures-bx6si>
- [61] (Internet) <https://medium.com/snapp-mobile/using-the-signature-class-to-verify-data-ff1add1da348>
- [62] (Internet) <https://towardsdatascience.com/signature-fraud-detection-an-advanced-analytics-approach-a795b0e588b2>
- [63] (Internet) <https://www.techtarget.com/searchsecurity/definition/asymmetric-cryptography>
- [64] Paar, Christof; Pelzl, Jan; Preneel, Bart (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer.

-
- [65] (Internet) [https://en.wikipedia.org/wiki/RSA_\(cryptosystem\)](https://en.wikipedia.org/wiki/RSA_(cryptosystem))
- [66] (Internet) https://en.wikipedia.org/wiki/Elliptic-curve_cryptography
- [67] (Internet) <https://medium.com/certik/how-bitcoin-works-mining-and-consensus-3d64bf893ba2>
- [68] (Internet) <https://www.edureka.co/blog/blockchain-mining/>
- [69] (Internet) <https://www.deltacbank.com/2021/10/05/bitcoin-transaction-validation-what-exactly-goes-on-under-the-hood/>
- [70] (Internet) <https://anyflip.com/yslwr/uron/basic>
- [71] (Internet) <https://en.wikipedia.org/wiki/Double-spending>
- [72] (Internet) <https://www.gemini.com/cryptopedia/double-spend-attacks-bitcoin>
- [73] Understanding a 51% Attack on the Blockchain. (2022). Engineering Education (EngEd) ProgramSection. <https://www.section.io/engineering-education/understanding-the-51-attack-on-blockchain/>
- [74] Nahar, P. (2021, August 31). What are 51% attacks in cryptocurrencies? *The Economic Times*. <https://economictimes.indiatimes.com/markets/cryptocurrency/what-are-51-attacks-in-cryptocurrencies/articleshow/85802504.cms>
- [75] Mcshane, G. (2021, October 12). What Is a 51% Attack? *Coin Desk*. <https://www.coindesk.com/learn/what-is-a-51-attack/>
- [76] Sayeed, S., & Marco-Gisbert, H. (2020). Proof of Adjourn (PoAj): A Novel Approach to Mitigate Blockchain Attacks. *Applied Sciences*, 10(18), 6607. <https://doi.org/10.3390/app10186607>
- [77] (Internet) <https://www.investopedia.com/terms/b/block-bitcoin-block.asp#:~:text=Blocks%20are%20data%20structures%20within,yet%20validated%20by%20the%20network.>
- [78] (Internet) <https://www.investopedia.com/terms/b/blockchain.asp>
- [79] (Internet) <https://www.investopedia.com/terms/p/proof-work.asp>
- [80] (Internet) <https://www.ibm.com/docs/en/ztpf/1.1.0.14?topic=concepts-public-key-cryptography>
- [81] (Internet) <https://www.investopedia.com/terms/b/bitcoin-wallet.asp>
- [82] (Internet) <https://en.bitcoin.it/wiki/Transaction>
- [83] (Internet) <https://academy.binance.com/en/glossary/mempool>
- [84] (Internet) <https://www.canada.ca/en/revenue-agency/news/newsroom/tax-tips/tax-tips-2022/mining-cryptocurrency.html>
- [85] (Internet) <https://cado-nfs.gitlabpages.inria.fr/>
- [86] (Internet) <https://github.com/doyler/SecurityTools/tree/master/RSAGenKey>
- [87] (Internet) <https://www.doyler.net/security-not-included/cracking-256-bit-rsa-keys>
- [88] (Internet) <http://www.loyalty.org/~schoen/rsa/>
- [89] Marc Stevens; Arjen Lenstra; Benne de Weger (2009-06-16). *"Chosen-prefix Collisions for MD5 and Applications"* (PDF).
- [90] Gaëtan Leurent; Thomas Peyrin (2020-01-08). *"SHA-1 is a Shambles: First Chosen-Prefix Collision on SHA-1 and Application to the PGP Web of Trust"* (PDF).

-
- [92] (Internet) <https://github.com/cr-marcstevens/hashclash>
- [93] *The first collision for full SHA-1*, Marc Stevens, Elie Bursztein.
- [94] (Internet) Wikipedia contributors. (2022, February 16). *Quantum computing*. Wikipedia. https://en.wikipedia.org/wiki/Quantum_computing
- [95] Sparkes, M. (2022b, January 25). *Quantum computers are a million times too small to hack bitcoin*. *New Scientist*. <https://www.newscientist.com/article/2305646-quantum-computers-are-a-million-times-too-small-to-hack-bitcoin/>
- [96] *Shor's factoring algorithm* | Quantiki. (2021). Shors. <https://www.quantiki.org/wiki/shors-factoring-algorithm>
- [97] (Internet) *Shor's algorithm*. (2022). IBM Quantum. <https://quantum.computing.ibm.com/composer/docs/irqx/guide/shors-algorithm>
- [98] (Internet) *Quantum computers and the Bitcoin blockchain*. (2022, January 10). Deloitte Netherlands. <https://www2.deloitte.com/nl/nl/pages/innovatie/artikelen/quantum-computers-and-the-bitcoin-blockchain.html>
- [99] (Internet) <https://www.namasteui.com/ways-the-quantum-computer-might-affect-the-blockchain/>
- [100] Stubbs, R. (2021). *Quantum Computing and its Impact on Cryptography*. *Quantum*. <https://www.cryptomathic.com/news-events/blog/quantum-computing-and-its-impact-on-cryptography>
- [101] L. Grover, "A Fast Quantum Mechanical Algorithm For Database Search," *Bell Labs, New Jersey, Tech. Rep.*, 1996
- [102] S. Bone and M. Castro, "A Brief History of Quantum Computing," *Surveys and Presentations in Information Systems Engineering (SURPRISE)*, vol. 4, no. 3, pp. 20–45, 1997, <http://www.doc.ic.ac.uk/~nd/surprise97/journal/vol4/spb3/>
- [103] D. Bernstein, E. Dahmen, and Buch, *Introduction to Post-Quantum Cryptography*. Springer-Verlag Berlin Heidelberg, 2010
- [104] G. Brassard, P. Høyer, and A. Tapp, *Quantum Cryptanalysis of Hash and Claw-Free Functions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 163–169
- [105] (Internet) Vasileios Mavroeidis, *The Impact of Quantum Computing on Present Cryptography*, 2018
- [106] (Internet) <https://www.techtarget.com/searchsecurity/definition/post-quantum-cryptography>
- [107] Santosh Ghosh, Rafael Misoczki, and Manoj R. Sastry, *Lightweight Post-Quantum-Secure Digital Signature Approach for IoT Motes*