



Cryptography in Blockchain

Submitted by :

Nouredine BOUHADDAOUI

B00148762

Professor :

Mark CUMMINS

Table of content

Introduction	3
1. Understanding Blockchain.....	4
a. Concept	4
b. Key principles of Blockchain	8
c. Security requirements and challenges for Blockchain.....	12
2. Security and cryptography of the blockchain.....	14
a. For granting Authentication	14
b. For granting confidentiality	15
c. For granting Integrity	15
d. For granting Availability.....	16
3. Analysis of common algorithms.....	16
a. Digital signature	16
b. Hashing	17
4. Implementation	18
a. Code source.....	19
b. Evaluation	19
c. Analysis	20
Conclusion	21
References	22

Introduction

Blockchain becomes one of the technologies which is revolutionizing our digital world. The massive adoption by the public since the launch of bitcoin cryptocurrency project by Satoshi Nakamoto in 2008, just after the financial crisis that had hit the world economy, and the increasing adoption by many institutions around the world, since 2015, pushed researchers and start-ups to innovate, to adapt the concept for many industries and sectors and to find new solutions for the different hurdles that they face in this evolution. The objective of this report is to carry out the cryptography algorithms which grant the security of this technology.

Many Blockchains have emerged since 2009, for different use cases and industries.

In this report I will start by giving an overview about the concept of blockchain, how it works, the ecosystem and the role or actions performed by each actor in this ecosystem. Then, I will list the different requirements for building a blockchain project based on what we have learnt, so far, from the existing blockchain, with more focus on security requirements. After, I will give an overview on the known cryptography algorithm used in the known blockchains. At the end we will dive in the common algorithm and principles in order to carry out the benefits, downside and alternatives of each one of them. At the end, I will illustrate the Proof of Work process by using hash256 and a piece of code. This implementation will showcase the complexity of this process and why it worth a reward.

1. Understanding Blockchain.

a. Concept

To understand the workflow of the blockchain, we will use the underlying system of Bitcoin. Most of this explanation was extracted from Satoshi Nakamoto's explanation which exists on www.bitcoin.org

As mentioned earlier, the Bitcoin is a cryptocurrency or a digital currency on a peer-to-peer network. This currency is for the real currencies what the email is for the mail. It works on a decentralized network where each customer is also a node and a server collaborating in the whole ecosystem. To manage the assets of each user in this network, it was paramount to build a decentralized ledger where we will keep updating all the transactions between users. This ledger doesn't belong to anyone and each user of the network has an updated status of it after each validation of transactions. We will see more in detail the process of validation later. This ecosystem is built on cryptographic proofs, which ensure security and trust and replace financial institutions. So, the system of blockchain should guaranty:

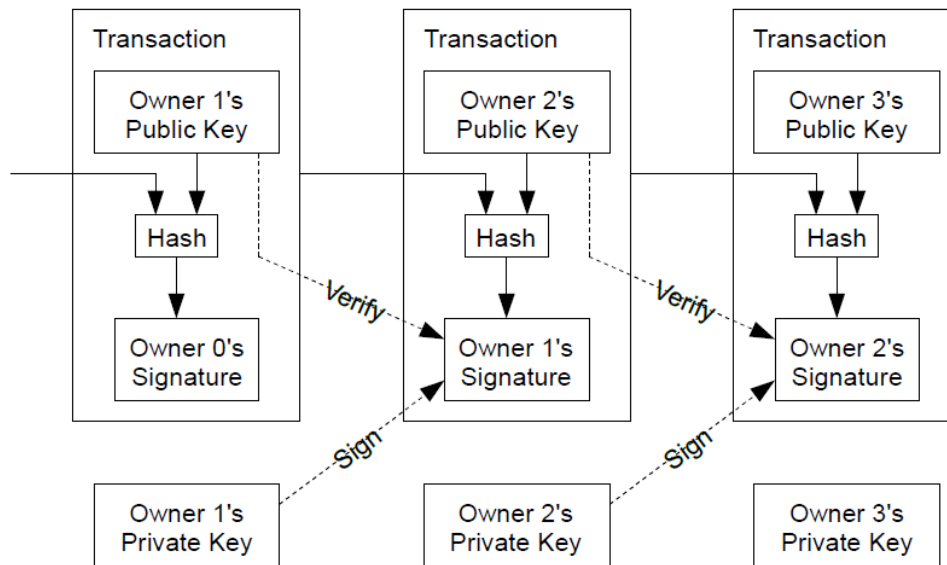
- A transaction between users without any intermediate.
- Protection of sellers against frauds by forbidding the change or delete of any transaction
- Protection of buyers from sellers who might try to sell the same asset to many buyers in the same time.
- Forbidding the double spending by using the timestamping.

This system could exist only if the computation power of the network's nodes is higher than the power of any set of nodes that is trying to attack simultaneously the blockchain's network

The transactions:

To manage the transactions, we need a system able to validate if the coin has not been spent earlier by its owner (double spending). As mentioned, the purpose of Blockchain is to delete any authority of control, and therefore use cryptographic algorithm for granting the trust between participants.

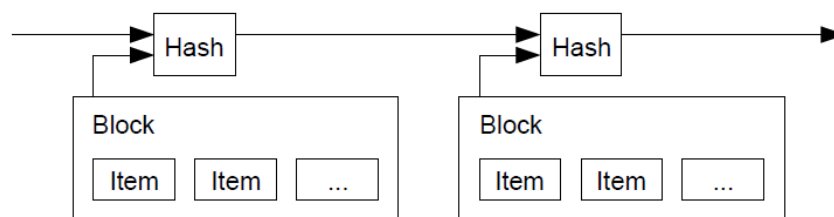
When a user A is about to launch a transaction, he generates, by using a dedicated software, a public key and private key. The private key is kept secret and known only by the user A. The public key is available for anyone in the network. As the A's public key is known by everyone in the network, this user also knows the public key of user B to whom he wants to send an amount of coins. So, the user A will give the amount that he want to send for example to another user B and he will sign digitally the previous transaction in the chain with the public key of the receiver B for creating a block. Now, the block needs to be validated by the network and added to the blockchain which is the shared ledger between all the participant in the network.



The timestamping:

The system relies on the timestamping function. The server follows these steps:

- The server collects a set of objects/items (transactions) and perform a hash of this set.
- It broadcast this hash as a message to all the nodes of the networks.
- Each hash includes the previous hashes with their timestamps, which creates a chain.



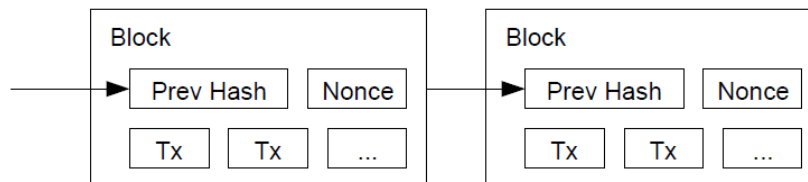
To explain more: As the hash is unique, if we have two blocks with the same transactions, but their previous transactions are different, we will get two different hashes of this blocks. Moreover, if two blocks have the same transactions but with a different ordering in each block the hashes of the two blocks will be different as well. Therefore, the blockchain is recognized by its unique hash.

Proof of work:

As mentioned before, the system needs a proof that the information of the block to add are authentic and there is no double-spending. This is the purpose of the "proof of work". We can't have a timestamp server on a peer-to-peer (distributed) network, so the "proof of work" is the algorithm which enable the consensus of the nodes within a network regarding a block which is trustworthy.

To be authenticated, a transaction should be added to a block which includes other transactions. This block will be validated by a process called "proof of work" which consist on calculate the double hash with SHA-256 (compute the hash 2 times) of the current block based on the previous one (current block= previous block + set of new transactions).

In order to add a new block to the blockchain, the participating node in the creation of the chain (called miners) should perform a cryptographic process: compute the hash (SHA256, 2 times) of the bloc, by adding and increasing a number called “nonce” in the input until we get a hash which starts with a specified number of “0”. This way we add complexity in the process. Therefore, the block cannot be changed without redoing the work on many blocks, as there will be some later blocks chained to this one after few minutes. The work to change the block would include redoing all the blocks after it.

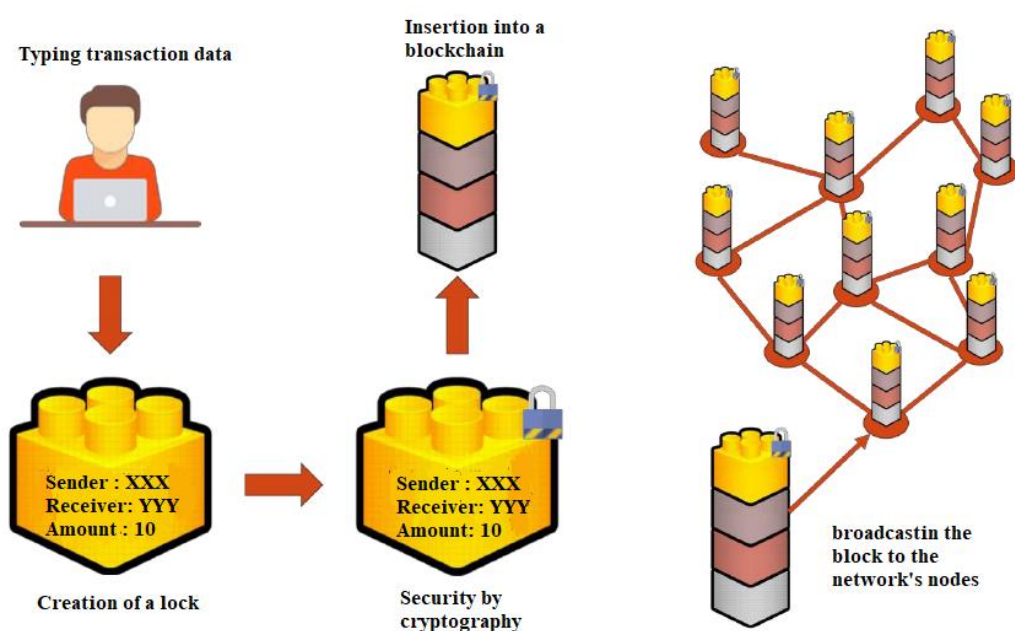


To illustrate, in this example we consider that the complexity (number of zeros in the beginning of the hash to be affected to a block) is 10 the scheme bellow showcase the activity of miners for adding a block to the blockchain:

- Compute the : $h = \text{function hash}(\text{Block}, \text{miner information}, \text{previous block hash}, \text{Nonce}) < C$

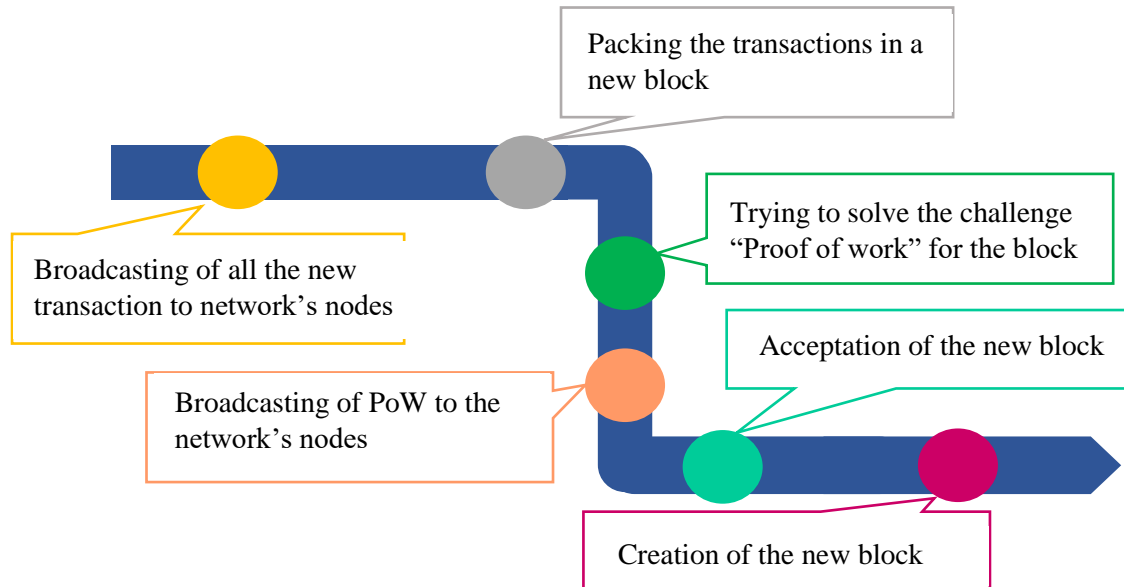
With C represents the complexity. If $C = 10$, we should find a hash smaller than 0000000000 followed by many 1s.

- As the hash function is deterministic, we should increase the nonce until getting a hash which comply with this pattern.
- Once a solution is found the block can be sent to the network's nodes.



Network:

The different rules which organize the network nodes are as followed bellow:



- The nodes accept the block proposed only if all the transactions included in it are valid.
- Nodes accept the block only if all transaction in it are valid and not already spent.
- The acceptance of a new block relies on the work done by a node on this new block, using the hash of the previous block that the node has just accepted previously.
- Only the longest chain is accepted in case of two node perform the work in the same time which can generate a fork (to be explained later).

Incentive:

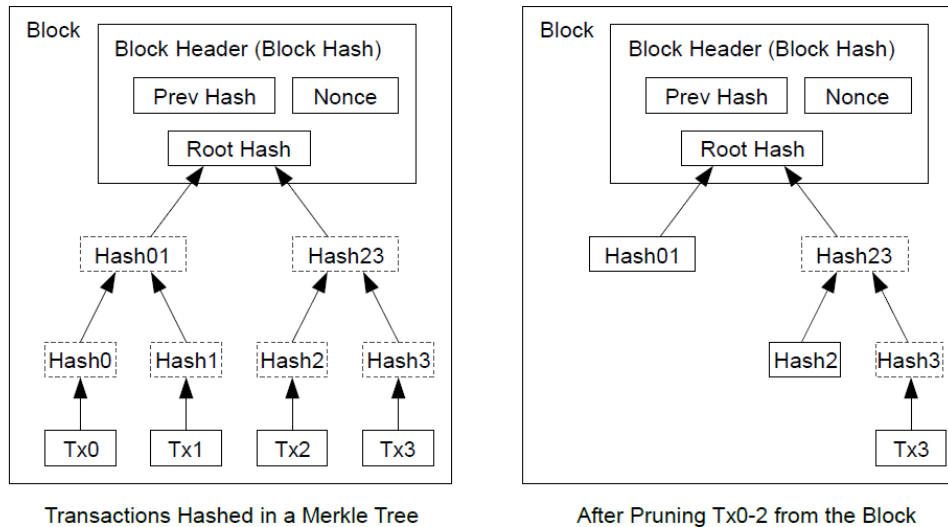
By convention, the first transaction comes from the creator of the block who starts a new coin. This adds an incentive for nodes to support the network and provides a way to initially distribute coins into circulation.

To reward the heavy work (algorithm, CPU and electricity consumption) of miners who succeed the proof of work. The system generates a transaction fees in bitcoin and gives it to the miner who solved the challenge. Once a beforehand-specified threshold of injected crypto-currency into the network is reached, the transaction fee will be the only economic model for rewarding the miners.

Disk Space Optimization:

We can imagine that a blockchain could monopolize a lot of resources, especially, in term of memory space, Satoshi Nakamoto has foreseen to optimize this space by using process like Merkle trees. When transactions are valid and confirmed since many generations of successive blocks, they could be deleted from nodes that need to optimize their space disk.

Therefore, it could be easy to check the authenticity of the Blockchain without having all the data distributed to each nodes of the network.



Thus, transactions are hashed in a Merkle Tree, with only the root included in the block's hash. Old block can be then compacted by stubbing off branches of tree.

Simplified payment verification:

It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it. So, the verification is reliable as long as honest nodes control the network, but is more if the network is overpowered by an attacker.

Privacy:

Even if transaction could be announced publicly, it's still possible to maintain the privacy by keeping public keys anonymous

In this section, I tried to explain the concept of Blockchain through the Bitcoin cryptocurrency and its principle. Let's see in general the key principles of Blockchain, with contrast to the bitcoin blockchain, in the next section.

b. Key principles of Blockchain

To recall, in the last section, we can state that the key principles that Bitcoin Blockchain rely on are:

- Defining the transactions
- Stage of validation by the network (with the timestamping and proof of work)
- Rewarding of miners
- Confidentiality.

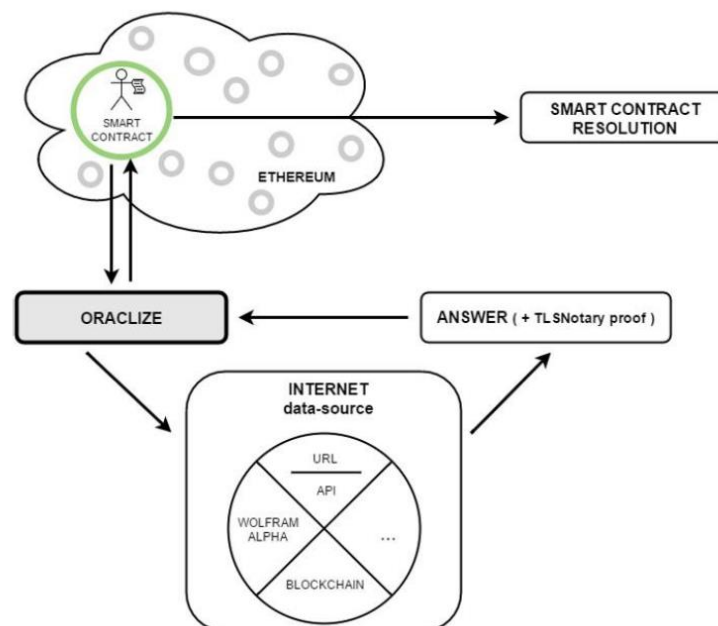
We will now see how this principle works in general for any Blockchain.

Objects of Blockchain:

The object of a blockchain can be different than a transaction (case of Bitcoin), the most known one is the Smart-Contract. A smart contract is a program inserted in a block which can be executed when some conditions are satisfied (eg: event). It implies that the Blockchain system should be able to collect information from outside the Blockchain.

For example, someone can choose to offer 1 btc to his brother, when this brother becomes parent (gets a baby). A smart contract can be created with this terms, and the blockchain system should be linked to an entity which can push to a predefined address the information that confirm when a brother got a baby. This entity is called an Oracle.

The Oracle is an automated organized which can check if the condition is realized, and when it's the case it will trigger the transaction into the Blockchain. Oraclize is a company which is operating on the Blockchain Ethereum (the crypto currency is "ether"), and provide the Oracle service to the system as illustrated below in the scheme:



Other models exist like DAO (Decentralized Autonomous Organization), it can for example provide an innovative insurance service. In this service the affiliates can pay a monthly fee through smart contracts to the organization in order to be covered when an accident occurs. When the accident occurs, and an affiliate ask to be refunded. The smart-contract will ask the oracle if an accident had really occurred, so it can compensate the affiliate or the compensation can be triggered after validation by the vote of all the affiliates.

The validation by the network:

The proof of work algorithm explained earlier permits to avoid attack on the system by creating a consensus between all the nodes of the network for maintaining a chain of valid transactions. This algorithm relies on proving that we used enough energy and CPU. This method is not the only one. Another method called Proof of stake become famous and used in many Blockchain. It relies on having crypto currency. To explain how it works, let's compare these two methods.

In proof of work, we can imagine that all the nodes have the same capacity/resources (CPU, energy, memory, ...) and each node throw a dice with same frequency in hope to get "6" face. When a node gets a 6 it announces it to all the network. If some users control more than one node, the one who control more node has more chance to succeed before. Similarly, if a node has more resource than the others, it has more chance to succeed. We can say that the chance to be the miner who compute the block and announce it to the network is correlated to the puissance of his node/controlled nodes.

In **proof of stake** method, the success of mining a block depends on the money belonged to the account of miner. For example, if an account has 10% of the money in the system, he would have 10% more chance to find the block and announce it to the network. With method, we spend less resources, but it's less secure than Proof of Work.

Miners' rewarding:

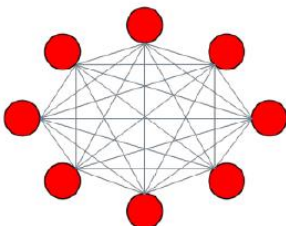
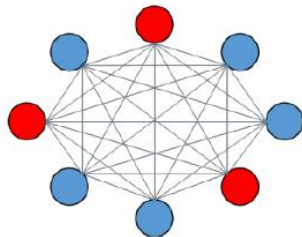
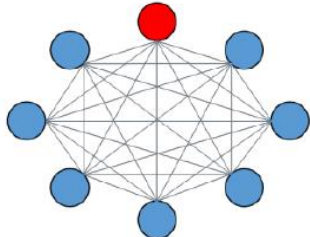
In Bitcoin, the reward is divided by 2 each 4 years. It's not the case in all the Blockchain. Ethereum rewards are the same for all the miners who add the blocks in the system every year since the beginning.

Privacy:

In the beginning the concept of blockchain relies on transparency of information related to the transactions and the validation was done in a public network.

Many organizations were interested by the blockchain but not the transparency of the system. That's why new models have emerged like the private blockchain (consortium) in addition to the public one. It relies on validation by a strictive set of known nodes in the network. The reading of whole blockchain could be also reserved to a set of nodes. It could also be hybrid organization where some information could be accessible publicly and some other information reserved to a private network.

In the private blockchain, the validation process is controlled by a unique actor. For the consultation, it could be private, public or hybrid.

Public blockchain	Blockchain of consortium	Private Blockchain
		



Nodes able to validate the block

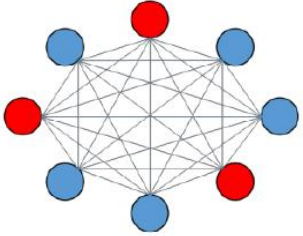
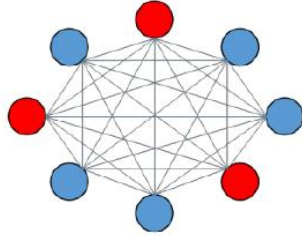
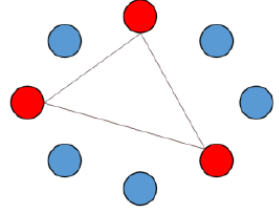


Nodes which don't have the required privileges for validating a block



Flow linked to a blockchain

In this table we see the difference in governance between different type of blockchain.

Public consultation	Hybrid consultation	Private consultation
		
All the nodes can have access to all the data	The access between nodes is open on objects with different privileges based on the members' roles	Only privileged members have access to the data



Nodes able to validate the block



Nodes which don't have the required privileges for validating a block



Flow linked to a blockchain

Based on the privacy and accessibility to information of the blockchain we can also classify the blockchain into 4 classes:

Blockchain Type	Permissionless Public	Permissioned Public	Permissionless Private	Permissioned Private
Access to the network	Anyone can join and leave.	Only participant with some privileges can validate	Anyone can join and leave.	membership is provided by the administrator or a membership authority.
Privileges of users on the blockchain	Everyone has read and write access	Everyone has read access Restricted write access to some participants	Restricted access to read the contracts and related data	Read and Write access of data are provided by admin
Benefits	Less trust Most transparent	Not fully decentralized	Collaboration in an organization without sharing information publicly	More trust Less transparent

More features of these classes are listed in the table below:

Blockchain type	Application Domain	Anonymity	Scalability	Challenges	example
Permissionless Public	Decentralized P2P Networks	High	Low	Privacy, Scalability	Bitcoin, Zerocash, Monero
Permissioned Public	Decentralized Organizations	High	Moderate	Privacy, Centralization	Ripple, EOS
Permissionless Private	Intra-Organization Networks	Moderate	Moderate	Consensus, Scalability	LTO
Permissioned Private	Organizational restricted ledgers	Low	High	Consensus, Centralization	Hyperledger fabric, Monax, Multichain

c. Security requirements and challenges for Blockchain.

Based on the analysis of existing blockchains in the previous sections, its clear that constructing a blockchain requires to address some security requirements: Confidentiality of information, Integrity of information, availability of information data privacy, anonymity and authentication.

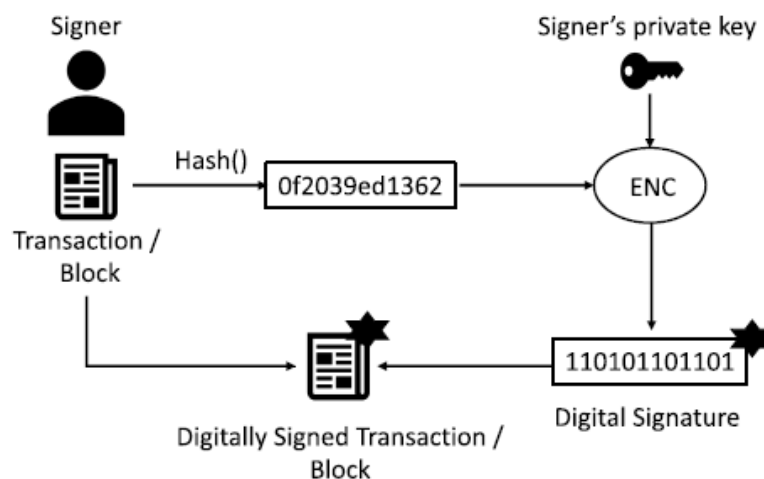
In case of blockchain, the term information used in the above context can have multiple meanings such as data in the database, smart contract data or transactions. Privacy can be defined as data privacy and user privacy (anonymity). The table below includes some cryptographic mechanisms for achieving security and privacy of information subjected to different blockchain layers.

	Confidentiality	Integrity	Availability	Data privacy	Anonymity and authentication
Smart contract	Encryption	MAC	-	Data privacy Preserving computation	Identity privacy Preserving Computation
And Transaction	-	Signature scheme	Access Structure of transactions	Zero-knowledge proofs, Mixing techniques	Zero-knowledge proofs
Consensus	-		Consensus	Access Control	Blind or Ring Signature
Network	Encryption	MAC	Protocol e.g. Gossip	-	IP anonymity e.g. TOR
Database			Access Control	Access Control	-

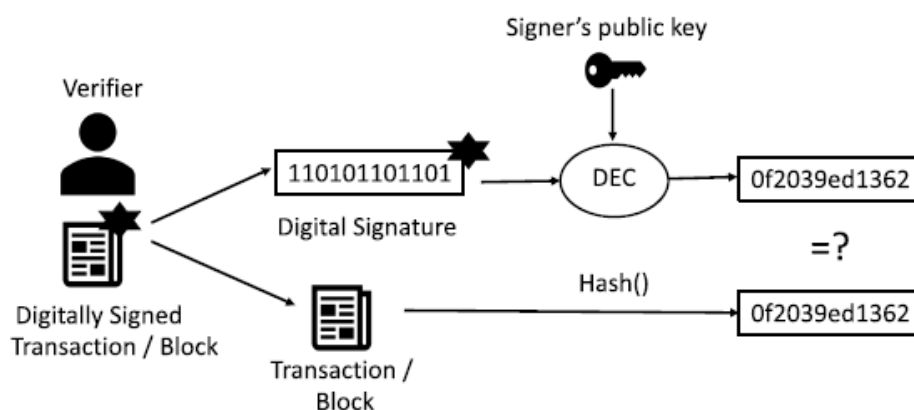
2. Security and cryptography of the blockchain.

a. For granting Authentication

For authenticating users, digital signature is used in many Blockchain and we have different implementation of this. A standard digital signature is a mathematical scheme based on public-key cryptography that aims to produce short codes called signatures of digital messages by the use of a private key, and where those signatures are verifiable by the use of the corresponding public key. Signature Schemes are used in almost all the blockchain. The figure below shows how a blockchain user creates a digitally signed transaction or block by using his private key.



Afterward, the nodes of the network will use the reverse mechanism to check if the transaction is valid by using the public key of the user who created the block, the figure below shows the process that performs the nodes of the network.



Many other signature schemes are used in different blockchains. Here is the list of some of them:

- Multi-signature
- Blind Signature
- Ring Signature
- Threshold Signature

b. For granting confidentiality

Encryption scheme:

In private blockchain, we might be interested in encoding a piece of information, so only authorized parties can access it. Symetric-key encryption is used in Hyperledger fabric for confidentiality of smart-contract and BlockChain for Smart Home. The challenge here, is how to perform computation on encrypted data. Like in searching encrypted data on the cloud, in permissioned BlockChain we can use fully homomorphic encryption and functional encryption. We can also use, for assuring confidentiality and authenticity, an authenticated encryption method based on digital signature with the public key and private key, as explained before. The information of the block could be encrypted by using the private key of the creator, and the validators who know the public key of the creator can also decrypt the information before performing the computation.

Access control:

It's about selective restriction on information or resource based on some policy or criteria.

This mechanism can be used for avoiding some incident and also in some BlockChain used in specific industries like medical applications or blockchain for Insurance industry where data is sensitive and should be accessible only for trusted and authorized parties. Here we list the different type of access control:

Role-Based Access Control - where the access is granted two participants based on their role and function in the system.

Attributed Based Access Control – where the access is based on the attribute structure (user, environment, object) for example in insurance company two different departments (Claim Handling , Audit) can have different view on the blockchain.

Organization-Based Access Control – where the model is based on three entities (subject, action, object) which define that some subject has permission to realize some action on some object.

Other algorithms for assuring the confidentiality exist, we can list bellow some of them:

- Secure Multi-Party Computation
- Secret Sharing
- Commitment scheme
- Lightweight Cryptography

c. For granting Integrity

As explained before the most used algorithm for assuring integrity (consistency in Blockchain case) is SHA256. It's used twice in the Proof of Work process for computing each version of the block with a challenge imposed by the system related of the number of zeros in the beginning of the hash. The lightweight cryptography can be used for some devices that cannot perform heavy computation because of the lack of resources in it.

d. For granting Availability

The Availability in Blockchain is tied to Integrity, and its challenge is to ensure that data has not been modified by an attacker to tamper transactions.

3. Analysis of common algorithms

a. Digital signature

Benefits

Digital signature is the most trustworthy mechanism for achieving the highest level of data security. Beside the hash value, the receiver confirms message authentication by approving the digital signature with the generated public key of the sender, which verifies the identity of the person they are communicating with. It guarantees the integrity of data transferred into the blockchain.

Downsides

It's difficult to see any downside in the digital signature today, apart the resources needed by the user to encrypt and decrypt a message, specially when the amount of data to be encrypted, decrypted and signed is important and the participants' resources are poor like in IoT.

Second downside of standard digital signature is that we need that all the parties participating in a system be trustworthy. As blockchain revolutionize the digital word by being public and often permissionless we need some mechanism that guarantees authentication and confidentiality without authenticating all that parties in the system.

Third downside is the linkability, in this model, the network can easily identify the creator of each transaction. In some industries or use cases, it might be crucial that the transaction remain anonymous.

Alternatives or area of improvements

In order to fix some of the downsides raised in the last section, many blockchains have used some different type of digital signature:

Blind signature: in this scheme the signer is different than the transaction authors. It's used to provide anonymity and unlikability.

Ring signature: in this scheme the signature can be done by a member of a group on behalf of the group, so the individual signer is not revealed. Thus, we can provide anonymity of the signing party. It could be used for ensuring untraceable payment like in CryptoNote.

Oblivious RAM: a cryptographic protocol through which a client can safely store his/her data in an untrusted server. The client performs read and write operations remotely. ORAM hides the memory access pattern from the server as well as from outside entities accessing to that part of data. ORAM provides freshness, confidentiality of data and integrity so it can be used in various blockchain use-cases and applications like Solidus framework which operates on a modest number of banks where each bank maintains a large number of user accounts. Solidus

a protocol for confidential transactions on public blockchain, uses oblivious RAM. Solidus framework operates on a modest number of banks where each bank. In Solidus, a new primitive called Publicly Verifiable Oblivious RAM Machine (PVORM) has been introduced. While ORAM is used to store user account balances, Solidus uses PVORM to verify the valid transaction set of a bank.

Zero-knowledge proofs: With these protocols we can prove the statement as 'transfer of an asset is valid' without revealing anything about the asset. These protocols are extremely useful for achieving secrecy in application. They can be used to provide the confidentiality of an asset (transaction data) in the blockchain while keeping the asset in the blockchain. Zerocoin is a decentralized mix and extension to Bitcoin for providing anonymity and unlinkability of transactions by applying zero-knowledge proofs

b. Hashing

Benefits

As I explained before, hash function provides integrity, and an effective way to validate the block. For blockchain, we compute hash256 of the candidate block two times and we should repeat the operation many times to find a hash with specified number of zero in the beginning of the hash. In this way it difficult to tamper the network with an invalid block, but there is still some downsides which push sometimes to improve the system.

Downsides

The most known one is the energy and resources spending that the system require for computing and validating the block. This makes the process unprofitable specially when the reward is decreasing. But to do this, miners need to operate in a lottery-like system where there can be only one winner. And there's no guarantee to find a hash. Besides, it will take an eternity to execute the process if the GPU and CPU power is weak, while only computers with expensive equipment can handle the processing power. Even so, these computers could consume a lot of electricity, which makes the whole process unprofitable for a miner who could find a valid hash for an invalid block of transactions. In the end, it does not make sense to confirm the "wrong" block, claiming that all transactions are valid, and proceed to hash. The rest of the computers on the network will reject the invalid block, meaning that the miner will not receive the reward.

The second downside is the management of a verry long blockchain which contain all the previous version of the blockchain. Hopefully, we can compact the chain by using the Merkle Tree principle.

The third one is the fact that the whole system rely on the complexity of computation which requires a big resource, but what will happen when the capacity of computers will increase following the Moore's law or where the quantum computing becomes accessible to everyone.

Alternatives or area of improvements

Post-quantum cryptography: The actual system foresees to increase the complexity of the challenge (e.g number of zero in the beginning of the hash) if the proof of Work process becomes relatively quick. This solution could be efficient, but one day the whole cryptography will be in a big challenge because of the quantum computing, which will enable complicated calculation in few times. At this moment all the cryptography processes will move to the post-quantum cryptography.

Lightweight cryptography: Lightweight cryptography targets sensor networks, embedded systems, and other variety of resource-constrained devices such as IoT end nodes and RFID tags. Lightweight cryptography is simpler and faster than conventional cryptography but less secure (suffers from many attacks). In IoT, embedded devices having sensors are interconnected through a public or private network. As these are resource-constrained devices, lightweight cryptography solves the issues of communication, memory, and power consumption, but still lacks security. To provide better security, blockchain can be used in conjunction with the sensor network.

A lightweight scalable blockchain (LSB) is also introduced to improve IoT security and privacy. LSB uses a lightweight hash function and lightweight consensus algorithm to achieve scalability, security, and privacy.

Blockchain is also used to cater security in electric vehicles, cloud and edge computing which use lightweight cryptographic primitives like lightweight symmetric key encryption.

Aggregate signature: This scheme, as introduced by Boneh et al. is a method for combining N signatures from N different signers on N different messages into a single signature. This single signature (and the N original messages) will convince the verifier that the N signers did indeed sign the N original messages (i.e., signer i signed message m for $i=1, \dots, N$). Typical applications for aggregate signatures are, for example, secure routing or certificate chain compression. The main benefit of aggregate signature is that it saves bandwidth, which makes it an optimal solution for networks of small, battery-powered devices that communicate over energy-consuming wireless channels.

4. Implementation

For this report I choose to implement and test a Proof of Concept of the “Proof of Work” layer based on executing a hash function on a message until finding a hash with a certain number of leading zeros.

As explained before, in BlockChain, the miners should find or generate a SHA-256 hash with a certain number of leading zeros. In this case we will add a Nonce to create hash with leading Zero. By this implementation I will test the complexity of generating this hash for different text and give a benchmark of the number on of nonce that have been tried and the time. For optimizing my time and knowing that my laptop CPU is not well shaped for mining, I decided to drop the program after 60 minutes if I don’t get any result for some inputs.

a. Code source

For this test, I picked a piece of code that I modified to change the values and compare the results. For the example bellow, I give as input “hello12” and the number of zeros leading the hash is 3. The screen shot of the execution on collab notebook give:

```
import hashlib
import sys

def PoW( val_in, zeros):

    hex_dig=''
    working=''
    nonce=0
    while True:
        val=val_in+str(nonce)
        inp=val.encode()
        hash_object = hashlib.sha256(inp)
        hex_dig = hash_object.hexdigest()
        working=working+val+hex_dig+'\n'
        if (hex_dig.startswith(zeros)): break
        nonce=nonce+1
    print ('Nonce is ', nonce)
    print ('Result is ',val)
    print ('Hash is ',hex_dig)

val_in='hello12'
zeros="000"
PoW (val_in, zeros)
```

Nonce is 747
Result is hello12747
Hash is 000dbb41da51cf16f790b4a3fc71d481e93621a29efc222c01401561153fdc7c

The original code source had been picked from this link (Bill Buchanan’s asecuritysite.com site web):

<https://asecuritysite.com/encryption/block?n=a>

The code source is available on github :

https://github.com/Nouredine-sec/Cryptohack-challenges/blob/main/blockchain/proof_of_work.py

b. Evaluation

For this evaluation, I will test different values and different challenges with increasing number of zeros that should lead the hash. The table bellow shows all this values and related results (Nonce, Hash and time of execution). As input, I tried with two message:

Message1 = ‘hello’

Message2 = “current transaction: from Nouredine to marc send 1 btc: last block
hash=0006bc9ad4253c42e32b546dc17e5ea3fe daecdabef371b09906cea9387e8695”

Input	Number of zeros	Result = Hash256	Nonce = Number of iterations	Time of execution
Message1	0	02a13c40ba00dc0fb199d3cbe5b01be59d937775890243fd411bdf001935ffc8	28	< 1''
Message1	00	001b92541ed0a22b0cb89018b561d895503206c0082c0ecf2d0b7e5182191eed	227	< 1''
Message1	000	0006bc9ad4253c42e32b546dc17e5ea3fe daecdabef371b09906cea9387e8695	10248	1''
Message1	0000	0000e49eab06aa7a6b3aef7708991b91a7e01451fd67f520b832b89b18f4e7de	60067	34''
Message1	00000	0000037660ee0e22df67a053537e000325bbfad2cce9b8b7c795f6aa961d5cb7		4'43''
Message1	000000	No result after one hour		
Message2	0	03867aeab775aa9779560d120bea0ce481bd3edb0bbfe5ac33547a8fe9682f59	3	< 1''
Message2	00	00ac7a85be1be13daa93b7877ca8d8ea91915914655af8c2b2365c18029d37cd	667	< 1''
Message2	000	0000e57d1a7e0188063ab2698f2cb666ae96bfd3bca94aaf14b65a9b2d7fdbfa	7201	<1''
Message2	0000	0000e57d1a7e0188063ab2698f2cb666ae96bfd3bca94aaf14b65a9b2d7fdbfa	7201	1''
Message2	00000	No result after one hour		

c. Analysis

Based on the results I got, I see how hard the mining work is. I tried with just simple inputs and simple challenges as number of zeros leading the hash result, and we see in table above that starting from 4 zeros the work starts to become complicated for the machine's resource. Miners should work on the algorithm in order to find a better way to optimize the execution of their programs, and as the process is mostly based on brute-forcing the challenge, they need powerful machine with high GPU and CPU, which means that this machine will consume more electricity than a standard machine. Miners could also work in a team with a distributed algorithm to increase their chance to find the block according to the challenge.

Conclusion

This research project helped me to understand the blockchain, and the underlying cryptography. It's the first time I try to understand this new area, and I am happy with what I have learnt, so far. Beside the understanding of cryptocurrency and how it works, I understand better, how the blockchain can revolutionize many industries by catering security, decentralization, and scalability. What was insightful, is the fact that we can build different BlockChain based on different protocols and cryptography algorithms, the choice of these protocols and algorithms depends on the features that we need to provide in the solution like, authentication, confidentiality, scalability, linkability, anonymity, ...

Second key point I have learnt, with this research is that blockchain could be the solution for securing services based on IoT, and some assets that could not support the standard cryptography because of the lack of resources in it (sensor networks, embedded systems and other variety of resource-constrained devices such as IoT end nodes and RFID tags). In my work, I tried to find many time an efficient way to secure IoT devices, and the only way that I found was to rely on Machine learning and SOC service in order to detect any suspicious activity related to IOT device. Now, I can try new idea based on lightweight cryptography and blockchain for this purpose.

This project gives me an idea about what might be the research project in the last semester of this master, next year. So, I start considering to invest more time in understanding advanced concepts of blockchain and security with the purpose of make it the focus of my research project.

References

The documents that have been used for the literature review as well as this report are uploaded into the github repository dedicated to this project:

<https://github.com/Noureddine-sec/Cryptohack-challenges/tree/main/blockchain>

The most important are listed below:

- SoK of Used Cryptography in Blockchain
<https://ieeexplore.ieee.org/abstract/document/8865045>
- White paper Bitcoin: A Peer-to-Peer Electronic Cash System by **Satoshi NAKAMOTO**.
<https://www.bitcoin.com/bitcoin.pdf>
- asecuritysite.com by **Bill BUCHANAN**.
<https://asecuritysite.com/eseurity/unit08>