Consensus Algorithm for a Private Blockchain

Conference Paper · July 2019

DOI: 10.1109/ICEIEC.2019.8784500

CITATIONS
11

READS
4,333

2 authors, including:

Rafael V. Páez
Pontificia Universidad Javeriana
31 PUBLICATIONS **100** CITATIONS

SEE PROFILE

Consensus Algorithm for a Private Blockchain

Bouvarel, Lucas; Páez, Rafael V.
Department of Systems Engineering
Pontificia Universidad Javeriana
Bogotá
{bouvarellu, paez-r}@javeriana.edu.co

***Abstract*— In this paper we give a quick overview of the Blockchain. We explain quickly how it works thanks to the hash function and the distributed architecture. Then come the description of the two most famous consensus algorithm: Proof of Work and Proof of Stake. We also present some security issues for each algorithm. This paper contains also a little explanation of others consensus algorithms that are inspired of these 2. Finally, a new consensus algorithm for private Blockchain is presented.**

Keywords— Blockchain; consensus algorithms; proof of work; proof of stake; hash function

I. INTRODUCTION

Today almost every transaction made in the world are controlled by a third-party. Imagine that Carl wants to transfer 10 dollars to Marc. For this to be done, they need a third-party that ensures this transaction and keeps it safe in case of necessity. But the problem is that this third-party could be subject of a lot of troubles. For example, it is possible that this party could be corrupted by someone and so it can change the terms of the transaction or delete it. The problem comes from the centralization of the transaction: everything depends on a single organization, causing trust to be insufficient. In Blockchain technology, in order to make the system trustful, the ledger which contains every transaction is shared and controlled by many organizations, which can be termed nodes or parties.

Blockchain is so called because of the form of his structure. When transactions are accepted and verified by some nodes, they are put together into a block that is added to a chain of blocks which contains older transactions. Blockchain has solved the problem of changing the original low-trust centralized ledger held by a single third-party, to a high-trust decentralized form held by different entities, or in other words, verifying nodes. The first Blockchain appeared in 2008 with the famous crypto money Bitcoin [9], developed by an anonymous called Satoshi Nakamoto. Thanks to the success of Bitcoin, Blockchain has become a technology with an intense attention. If the Blockchain and the bitcoin have been created together, today a lot of actors (companies, governments, etc) are considering the Blockchain technology for other utilizations than the crypto money.

There are two important types of Blockchain public and private. If anyone who wants to maintain the ledger can join, it is a public Blockchain. In the other case, it is a private one. But whatever the type of the Blockchain, each of them needs a consensus algorithm. A consensus algorithm decides how agreement is made to append a new block between all nodes in the verifying network. Again, there are a lot of different consensus algorithm but the two most used are the Proof of Work (PoW) and the Proof of Stake (PoS).

In this paper we will see first a large view of the Blockchain technology. Then in a second part we will talk more precisely about consensus algorithms which are the ones that make Blockchain work. Finally, in the last part we present a new consensus algorithm developed for a private Blockchain.

II. BLOCKCHAIN PRESENTATION

A. The hash function

Blockchain technology is a way to store and transfer information. For this technology to be usable, it needs to respect some rules. This technology needs to be lasting. It means that even after years, if the Blockchain works, the data are still present and available. Blockchain and the data that it contains have to be unforgeable. Once a data gets in the Blockchain, it should be impossible to remove it or to change anything of it. And finally, the architecture

which stores all the data is distributed. Like we said in the introduction, Blockchain is different of the centralized common way of storing data. The centralized architecture stores every data in a single place and distribute them to everyone. In the decentralized architecture, everyone is connected with everyone. So even if we cut a connection with one node, all the other nodes stay connected.

Blockchain is so famous today because since the creation of the Bitcoin, nobody nobody has managed to falsify the information or has found a way to generate bitcoin for itself. First of all, we need to understand the tool that makes everything work. This tool is called the hash function. This is a mathematical function which generates a short chain from any type of file (text, image, video…), to what we called a hash. A hash is chain of bits (generally 256), which is frequently represented by an hexadecimal chain of characters.

If this hash function is so powerful, it's because it's impossible to obtain the original file from its corresponding hash but this hash function identifies in a univoque way the file. If we calculate the hash of two very similar texts with only one character different, we will get two completely different hashes while the texts are practically the same. For example, in the figure 1 we can see the hash calculated with the SHA-256 algorithm (the one used in Bitcoin) of two simple texts, they only differ from one character. However, we can see that the two hashes are very different.

Figure 1: Example of two hashes

At the first view, we have the impression that this function does not have logic. Rather, this function is completely determinist. From the same file, it will always return the same hash. The idea is that we cannot predict at all what the function will return. But what is the point of this hash? It is like the fingerprint or the DNA of someone which allow to identify a person between all the population. It is possible to recognize between all the files that exist only one with his hash. The result of this calculation is the characteristic of this file. We should not find two files that create two same hash.

The hash can be useful in different fields. The storage of password for example uses this hash. For security reasons, we do not store password into the server database. Imagine that someone can access illegally to this database, he will have access to all passwords and it represents a very great threat for every user's privacy and also for the application. So, instead of storing passwords, the hash of this passwords are stored. So every time that someone tries to log in, when he/she writes the password, a hash function is used and then we compare the result of this hash function with the hash that is stored in the database. The hash can also be used for example in integrity verifications of files, it is nameda a MAC (Message authentication Code). You just need to compare the two hashes of two files to see if they are exactly the same.

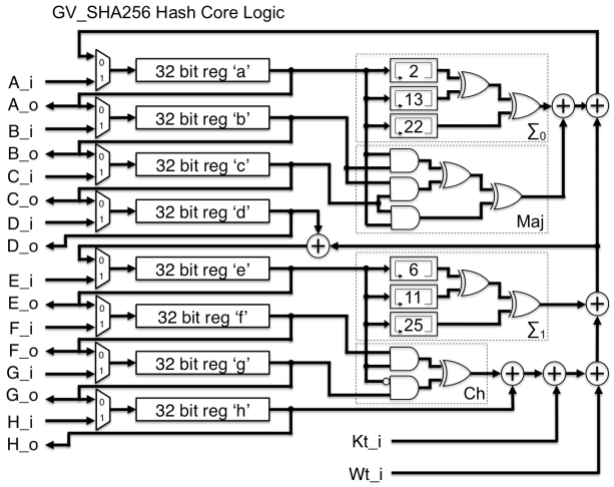

Figure 2: SHA256 architecture
https://opencores.org/projects/sha256_hash_core

The hash function works thanks to logical ports. Logical ports are tools that act on bit strings. there are many logical ports but the most famous are the NOT, the AND, the OR and the XOR. All of them make simple operation over bit strings. A hash function will transform numerical files with many bit operations thanks to a complex architecture of logical ports. For example, in the SHA-256 function from a message of 256 bits, you cut it in 8 parts of 32 bits and then all of this parts pass into a series of logical port as we see in figure 2. This process is realized 64 times in a row. So if we change one thing in the original file, the output of this function is completely different.

The reason of the creation of this function is simple: avoid collisions. A collision is when two different files will return two equal hashes. Of course, collisions exist, for example, there are only 256 bits to a SHA-256 function. So there are many of possible collisions. But the thing is, we don't know how to find a collision. We don't know how to construct two files that will have the same hash. For the hash function MD5, a research has found a way to calculate collisions [4]. That's the reason why it is today inadvisable to use it. The algorithm SHA-256 used in Bitcoin is assumed safe because no one have found a way to make a collision attack. But there is still one problem in mathematical research. We don't have mathematical proof that it is impossible for a hash function, such as SHA-256, to calculate collision. So today, we still don't know if these hash functions actually work and if we can create a Blockchain that is resistant to collisions and attacks. There are various hash functions so if we find a way to make collisions in one function, it is also necessary to change it.

As long as there is no successful attack on a hash function, it is considered secure because of the complexity of the mathematical problem. So nowadays hash functions are very useful in many fields and in the Blockchain technology.

B. The Blockchain's architecture

Like we have said in the introduction, a Blockchain is so called because of his architecture which consists in a list of blocks connected to each other, through hash functions, like a chain. In the block header, there are many fields that are useful for the good operation of the Blockchain:

- *Version*: The version number of the Blockchain
- *Previous block hash*: Contains the reference to the hash of the previous bock in the chain. This field allows the Blockchain to be ordered because new blocks need the latest block hash to be created so when a block has the reference of another previous one, the last one is inevitably younger. This field also allows security in the Blockchain. We will see in the next section that thanks to this previous block hash, it is almost impossible to change an older block.
- *Merkle Root or Tx_Root*: Contains the hash value of all validated transactions of the block. All the transactions are hashed, then they combine with each other pair-by-pair, and are inputted to another hash function. This work is repeated, until there is only a single entity, which stands for the Merkle root.
- *Timestamp*: The creation time of the block
- *Difficulty target or Bits*: contains the PoW algorithm difficulty target of this block. More precisions will be given in the next section.
- *Nonce*: A value that proves the efforts that a node has paid for getting the right to append his block to the chain. This field will be presented in the next section.

A block is the merging of a certain number of data records, where the form and meaning of them depend on the application. In the case of Bitcoin and other payment systems, these data records are monetary transactions, and each block typically contains in average more than 500 transactions [3]. The block is completed by a header that contains others information that are useful. From any block, we can move back up to the initial block, which is called the genesis block.

III. CONSENSUS ALGORITHMS

In this section, a first little part is dedicated to the verification of the sender and the validity of the transactions, then the two main consensus algorithms will be presented. These algorithms are essential because their mechanisms are used to add a new block in the chain and also to have controls on the Blockchain.

A. Identity and transactions verifications

The first thing to do before adding a block in the chain is to verify the identity of the sender, that is to say, we need to be sure that the person who makes the transaction is really the right person. We use two keys to do this work: public and private keys, they are known as digital signatures [8]. When a transaction is made, it is "signed" by the user's private key. In other words, the transaction and his private key are inputted in a sign function in order to create a personal signature. After that, the sender sends to the verifier a request transaction with the transaction he wants to make and his signature created by the algorithm. To verify if this request belongs to the right sender, the verifier uses the sender's public key, which is known by everybody. The verifier has a mathematical function that allows him to verify, with the signature and the public key, if the request belongs to the right person. It's

impossible to fake a signature because it is composed of 256 bits and today's computer cannot calculate 2^{256} cases quickly in order to make a false transaction.

Moreover of controlling the identity of the sender, the verifier also has to check the validity of the transaction of the block, that is to say, if the sender has enough money to make his transaction. This task is simply done by looking at the public ledger. Indeed, the ledger contains every past information so it is easy to validate a transaction or not.

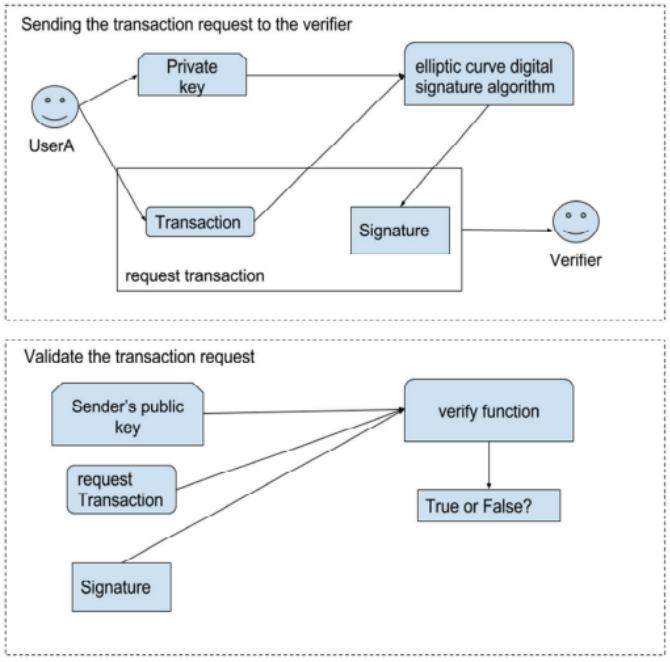

Figure 3: Sender's identity verification [2]

B. Proof of Work algorithm

The proof of work algorithm is the first algorithm created for Blockchain. Indeed, the first one was created with the Bitcoin in 2009. That is why today this algorithm is the most famous around the world and wrongly considered by many people as the only type of Blockchain. This consensus algorithm consists in resolving a puzzle more or less difficult to gain the right to append a new block in the chain. This puzzle is a very hard mathematical problem based on a cryptographic hash algorithm. Every node tries to find a solution of this puzzle to gain this right because if you are the lucky one, you will be rewarded. Indeed, everyone tries but when one node finds a solution, every other node stops their research and verify this solution. If this solution is accepted by all the other nodes, the finder will gain a reward. In the case of the Bitcoin for example this work is called mining and the ones who are resolving the puzzle are called miners. When a miner finds a solution in the Blockchain of Bitcoin, he will be rewarded of 12,5 [3] bitcoins per block. But mining is not only the fact of resolving a puzzle that is really hard and be rewarded, it is also a way to secures the Blockchain system and enables the emergence of network-wide consensus without a central authority. The creation of crypto currency is an incentive system which helps to secure the decentralized ledger.

We have seen in the previous section that blocks are linked to each other thanks to the previous block hash field. Indeed, every block has a unique hash that represents them. This hash is used in the puzzle of the PoW algorithm. When a block is ready to be append to the BloccChain, every field of the header are filled except the Nonce one. The Nonce value is the solution of the mathematical problem. It must be very long to find because it avoids the fact that everyone can create block every time they want. If anyone could add a block in a simple way, it will be difficult to control who added a block. The Blockchain forks problem will become really important which would make the Blockchain unusable because of security issues. So, when a block is ready to be appended, we need to find an appropriate Nonce value. The difficulty target field can be understood as a challenge, where the target is to find a hash with a determined number of 0s starting the resulting hash of the block. The goal is to find a Nonce value that results in a block header hash that begins with this necessary number of 0. In other words, mining is the process of hashing the block header repeatedly, changing one parameter, until the resulting hash matches a specific target. We have seen that the result of a good hash function cannot be determined in advance. There is no other way to find a specific solution of the problem than to try many times, until a hash which solve the challenge (a sufficient number of zeros at the beginning) is founded by repetition.

When a miner calculates a Nonce value, he adds him to the header block and hash it. If this hash value is smaller than the difficulty target, the Nonce value is a possible solution of the puzzle [3], and so the block and the solution is proposed to every node. To understand more easily, we can imagine that all the header block is a sentence and the Nonce value is a number that we can add at the end of this sentence. To make an easy challenge let's set a target at 3 which means that the hash function needs to begin with 3 zeros. In Figure 4 we have assumed

that the sentence "I am a simple sentence" is equivalent to the header block and the number that follows this sentence is the Nonce Value. The difficulty target was 3, and it takes us 20 attempts to find and hexadecimal which starts with 3 zeros. An actual computer with a great processor like an Intel I7 can calculate thousands of thousands of hashes each second so the difficulty needs to be more than 3 zeros. Currently for the Bitcoin's Blockchain the result needs to begin with 72 zeros to be validated. So, you need to calculate, on average, 2^{71} times the SHA-256 hash function. For a normal home computer, it would take thousands of years to find the appropriate Nonce. But it is like the lottery, if you have only one lottery ticket you will have low probabilities to win but if you have millions of tickets, you will have more probabilities to win. Because of the notoriety of the Bitcoin, it is today almost impossible to mine a block alone. People gather into what we called a pool to try together to find the right Nonce Value. It also exits places like large warehouses filled with calculating machines which calculate millions of hashes per second to hope to find the right solution and get the reward.

```
I am a simple sentence 0 => 2544b01848c35edf5 ...
I am a simple sentence 1 => ec038405afd090d1f ...
I am a simple sentence 2 => a079ab19ea32da05 ...
I am a simple sentence 3 => 7bde3a7127f984f89 ...
I am a simple sentence 4 => 30a5bb8b97d976e9 ...
I am a simple sentence 5 => 75113707155eb6e5 ...
I am a simple sentence 6 => 3f4e0f10f508f959d9 ...
I am a simple sentence 7 => 37916d288e24f432e ...
I am a simple sentence 8 => 2a7f67c5d71d4c51f1 ...
I am a simple sentence 9 => bac83bd11bb0ec561 ...
I am a simple sentence 10 => 6213850285b15c25 ...
I am a simple sentence 11 => b43600e8b9240757 ...
I am a simple sentence 12 => 39cc56bbf25f79344 ...
I am a simple sentence 13 => 7f4ed105d058d23e ...
I am a simple sentence 14 => 35fba6149b8bd0fe1 ...
I am a simple sentence 15=> be97a313cd03fa469 ...
I am a simple sentence 16 => 535348dcbe36a2ef ...
I am a simple sentence 17 => d812c3868a1065ad ...
I am a simple sentence 18 => d7d727bd1bddfa31 ...
I am a simple sentence 19 => e4ee2694f8a47f878 ...
I am a simple sentence 20 => 1ea3a78bee4d61e0 ...
```

Figure 4: Example of the mining work

The difficulty target in Bitcoin Blockchain is adjusted every 2016 blocks so that the average time stay at 10 minutes. This difficulty is calculated according to the computing power of every node.

Although, it represents big environmental problems because today the value of bitcoin is very expensive and a lot of people try to mine to get the reward. So, the difficulty of the target is very high and it requires a very high electricity consumption to resolve it.

When someone finds a solution, he broadcasts it to every nodes of the chain in order to add the block into the public ledger. Even if it is a really hard to find this proof of work, it may happen that before that everyone receives the validated block, another node finds a solution and starts broadcasting it. The other miners receive the first block sent and ignore the other one. That is to say, not all nodes have the same ledger as it should be. This is called the forking problem. There are different versions of the ledger with different chains of block. To resolve this problem, every node continue with the block they received first. After a certain time, a chain will be longer than the others because they can be more to mine for example and so they will have more chance to extend the chain. In this case, the rule is simple, the longest chain wins. So when a fork problem occurred, we continue the process and later, the longest chain is accepted by every node. Therefore, the youngest block in a chain can be subject to variation but thanks to the difficulty of adding a block, the stabilization of the Blockchain is sure.

C. Security problems in Proof of Work algorithm

Blockchain is today considered as a technology really secure because of the difficulty to falsify a block or to change the ledger. However, there are some security issues that have been found and this resulted in the development of variants proof-based algorithms.

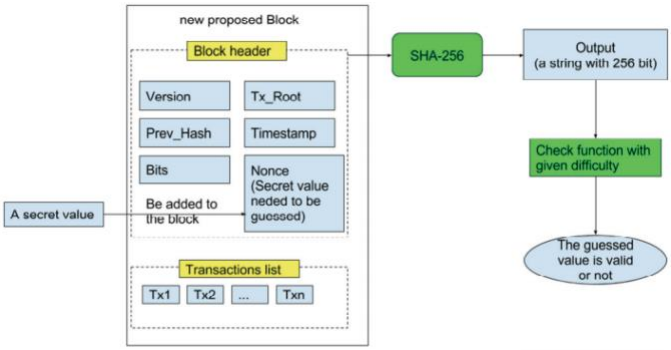

Figure 5: The mining process [2]

Imagine for example that we change the contents of an older block. The hash of this block becomes no longer good because the Nonce value found for the block was appropriate for the older transactions. But like we have seen, if we change a little thing in the input of a hash function, everything changes in the output. Consequently, every other block that follow this invalid block are also invalid. In fact, every block are linked to each other like a chain, thanks to the previous hash block field in the header. The header is used as the input of the hash

function so for every following blocks, the Nonce value is no longer good. That is to say, thanks to this hash function, changing a little part of the Blockchain, makes everything that follows false and you need to find again all the Nonce value of the following blocks. Because of the difficulty of the puzzle and the rule that the longest chain wins, it seems impossible to falsify the Blockchain. But imagine that this person has a really huge resource of modern hardware that allows him to have more computing power than all the network's node. In this case, he will be able to falsify all the Blockchain. A famous attack consists in starting mining a fraudulent branch at the end of the Blockchain and trying the best to be longer than the honest fork. This attack is called the Double spending attack or the 51% attack in reference of the minimum computing power of the network required in order to do this work.

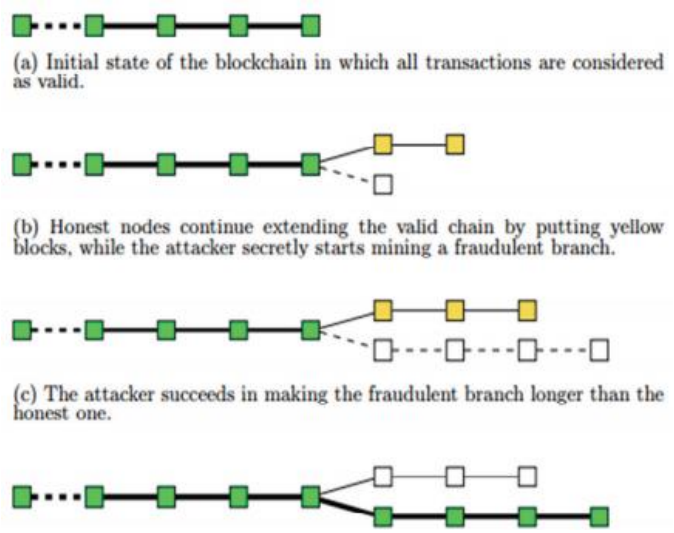

Figure 6: The double spending attack [2]

Today, it is impossible to do this alone but thanks to mining pool, this kind of attack could be possible. To avoid this problem many research has been done [5]. In mining pool, the reward would be shared by everyone in equal part. To disturb pool mining, Miller et al. [5] proposed an evolution of PoW algorithm. They called it the non out-sourceable puzzles. This consensus algorithm was developed to discourage the mining pool and to do that, they created a mechanism which give chance to a miner inside a pool to win all the rewards without making any effort. Another technique is used in the Ethereum crypto currency, it is called GHOST (Greedy Heaviest Observed SubTree). This consensus algorithm requires the mining nodes to include, in the header of the block they want to validate, the headers of the recently orphaned blocks known as uncles. Orphaned blocks are blocks that have been added on parallel branches of the main Blockchain. For Bitcoin, an uncle is therefor a block that would be considered as an orphan because it is not located on the longest chain. In GHOST strategy the longest chain is not chosen, it is the one with the most PoW contributing that will be selected as the valid chain. Ethereum encourages minors to include a list of uncles when they validate a new block. This technique has two main effects:

- It reduces incentive of centralization by always rewarding (minimally) miners who produce obsolete or orphaned blocks because they are not part of a large group and get noticed about other blocks later (due to propagation delays of the network).
- It increases the safety of the chain by increasing the amount of work on the main chain. As a result, less work is wasted on alternative branches in favor of the main branch.

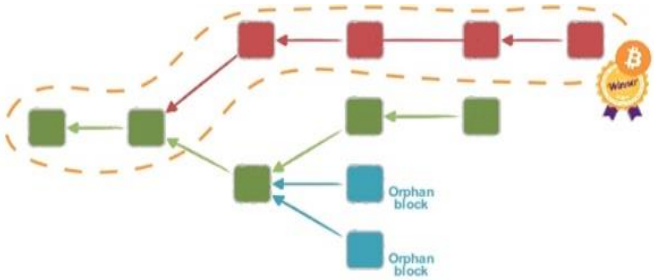

Figure 7: The surrounded chain is selected in the longest chain rule https://medium.com/@godefroy.galas/analyse-et-comparaison-des-m%C3%A9canismes-de-consensus-dans-la-blockchain-f91aee511ea3

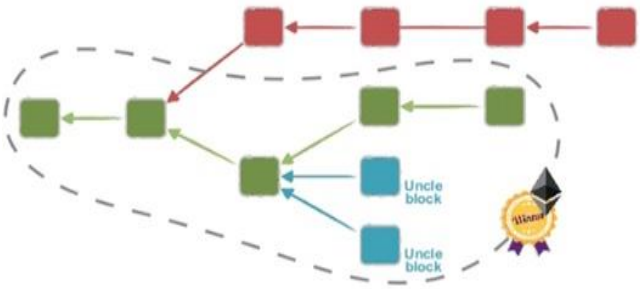

Figure 8: The surrounded chain is selected in GHOST strategy https://medium.com/@godefroy.galas/analyse-et-comparaison-des-m%C3%A9canismes-de-consensus-dans-la-blockchain-f91aee511ea3

D. Proof of Stake algorithm

We have just seen, in the previous section, the operation of the Proof of Work mechanism where each

node of the network simultaneously tries to solve a complex mathematical problem in order to validate and diffuse to the rest of the network a new block of transactions. This node, which is called a minor, is rewarded for this complex work.

The operation within a Proof of Stake consensus mechanism is quite different. In this system, each node of the network has to prove that it has a certain part of the circulating supply if it wishes to take part in the process of block validation. The network algorithm will then choose to delegate the validation of a new block to one of the nodes of the network according to an algorithm taking into account the amount of possessed coins. In a simplistic way, in the context of a PoS type consensus mechanism, the probability for a node to be selected to validate a new block corresponds to its holding percentage, its stake of the circulating supply. If a node owns *x* coins and if there is a total of *y* coins, its chance to append the next block is *x/y*. This selection is done in a pseudorandom way to prevent a node from knowing in advance when it will be his turn to validate the next block. Nevertheless, taking into account certain additional parameters, such as the possession time of the coin, allow the "richest" nodes to be almost always selected. Finally, the validation of a block does not strictly give rise to a remuneration, it is rather the holding of a certain amount of crypto currency that pays a remuneration (similar to the interests).

Compared to the PoW method, PoS has two main advantages:

- *Saving energy*: The PoS is a mechanism that consumes much less energy than PoW (which, in turn, requires a large number of cryptographic calculations to find the proof of work required for the validation of each block).
- *51% attack are more difficult*: In a PoS-type system, the 51% attack requires controlling more than half of the circulating supply, which is usually much more expensive than controlling 51% of computing power in the PoW system.

However, PoS algorithms have some disadvantages and security problems too. One famous problem is called Nothing-at-Stake (Figure 9). In pure PoS algorithm, nodes are not encouraged to vote for the chain that would be the most likely to be legitimate (i.e the longest for the Bitcoin platform). In presence of several potential chains (in case of fork), and in order to maximize their probability of obtaining the reward, the nodes will therefore allocate their "stake" uniformly and thus vote in parallel for the last blocks composing the potential chains. Unlike PoW where mining on multiple chains simultaneously costs energy and so money to the miner, mining on multiple chains costs nothing in a PoS system. Consequently, by satisfying their personal interest and by doing so, the miners facilitate the realization of double spending type attack. An attacker may be able to send a transaction in exchange for some digital good (usually another crypto currency), receive the good, then start a fork of the Blockchain from one block behind the transaction and send the money to themselves instead, and even with 1% of the total stake the attacker's fork would win because everyone else is mining on both.

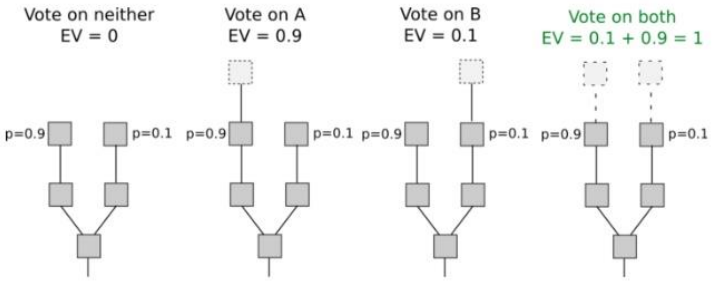

Figure 9: Nothing-at-Stake problem [2]

In pure PoS, the miner is selected on the pure stake he owns: more stake a miner has, the more chance he will get to become the block appender. Unfortunately, pure PoS would lead to an (undesirable) consequence of centralization: the richest member would always have an advantage. To avoid this, several methods have been developed.

E. Other form of consensus Algorithm

In the following Satoshi consensus [6] algorithm, the miner is chosen based on the state of the block. A Satoshi is the smallest currency unit of a crypto money. To decide who is going to append the next block, a random index number is chosen between 0 and the total number of Satoshi. Then, every transaction which have used this Satoshi are found out and the current owner of this Satoshi will become the one appending the next block. In delegated proof of stake consensus algorithm, every node which own stake will have the right to vote for a delegation which includes witnesses, who are miners verifying the transactions and maintaining the chain. The more stake a node has, the more powerful voting he has to assign the witness. The witnesses inside the delegation

have the right to verify transactions and to append new blocks in the chain. The list of witnesses is frequently changed by making new votes. This allows new stake holders votes to be taken into account. In PPcoin [7] consensus algorithm a new definition has been proposed, it is called the coin age. The coin age is calculated by the stake of a node multiplied by the time it has owned it. For example, if a person has 10 coins and keeps them during 3 days, he will have 30 coin-day. In order to get the right to append a new block, a miner has to spend a certain amount of coin age. The amount of money spent on this transaction will provide the miner more chance to mine a new block. Afterwards, he will have to do a puzzle, like PoW. The more money he spends on the transaction, the easier the puzzle he has to solve. Actually, the coin age consensus algorithm is an hybrid form of PoW and PoS. If any miner solves the puzzle first, he will get 1% of the amount of coins he has spent in the transaction, but the coin age accumulated by these coins will be reset to 0.

In the Figure 10 [2] a comparative analysis between the PoW, PoS and their hybrid form is shown. As we can see, only the PoS allows energy efficiency and do not requires modern hardware. It seems to be the perfect consensus algorithm but there also disadvantages. The pool mining is difficult to be prevented in PoS so if a blockchain do needs to control the pool mining, it seems that PoS is not the consensus algorithm that should be chosen. The hybrid form of PoW and PoS can be a good alternative to a blockchain that needs both advantages.

Criteria	PoW	PoS	Hybryd form of PoW and PoS
Energy efficiency	No	Yes	No
Modern hardware	Very important	No need	Important
Forking	When two nodes find the suitable nonce at the same time	Very Difficult	Probably
Double spending attack	Yes	Difficult	Yes, but less serious than in PoW
Block creating speed	Low, depends on variant	Fast	Low, depends on variant
Pool mining	Yes, but it can be prevented	Yes, and it is difficult to prevent	Yes, but less serious than in PoW
Example	Bitcoin	Nextcoin	Ppcoin

Figure 10 : Comparison between PoW, PoS and their hybrid form [2]

IV. Consensus Algorithm for private Blockchain

In this section we present a new consensus algorithm based on the Proof of Luck [10]. It has been modified in order to simplify the implementation and adapt it to be suitable for a private blockchain. Indeed, in Proof of Luck, trusted execution environments are used to be sure that a node is executing the well code and that it is not intending to attack the Blockchain. As we are in an environment where the nodes can be considered as trustful, we do not need theses trusted environments anymore.

A. Torneo consensus algorithm

When a node wants to join the blockchain he asks the right for by sending a request to every node. If the node is confirmed as authorized, he will receive the blockchain, the right to create transaction and the right to try to add the next blocks. Every 15 seconds (this time can be changed), a request is sent to every connected node to pick randomly a number between 0 and 1. Every node will broadcast this number and wait to receive every random number. When a node has received every random number, he selects the biggest one and sends to the node who had selected this number a "winner vote". When a node has received as many "winner vote" as the number of connected nodes, it means that he is the winner and he has the right to add the next block. Then the node has to mine the block in the same way as a proof of work consensus algorithm but with a very low difficulty allowing a normal computer to find the nonce value in less than a second. In this way the loss of time and the energy waste are avoided.

B. Analysis

A few tests have been made in network between two computers to see the efficiency of this consensus algorithm. In these tests, one transaction is a little character chain. In Figure 11 we can see that the time needed to broadcast x transactions increases linearly as the number of transaction increases too. It takes approximately 1,2 seconds to create and broadcast 1000 transactions.

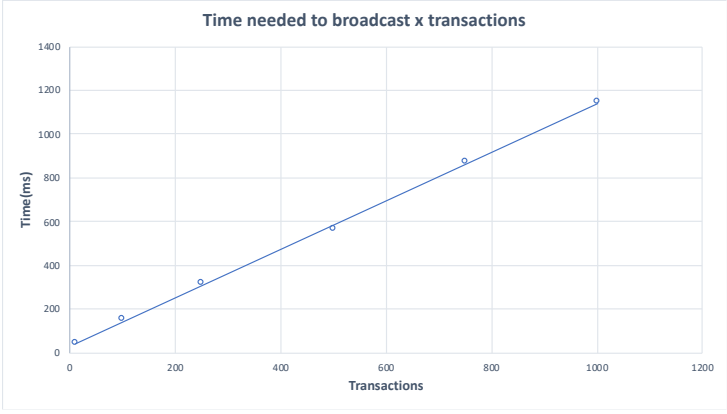

Figure 11 : Time needed to broadcast x transactions

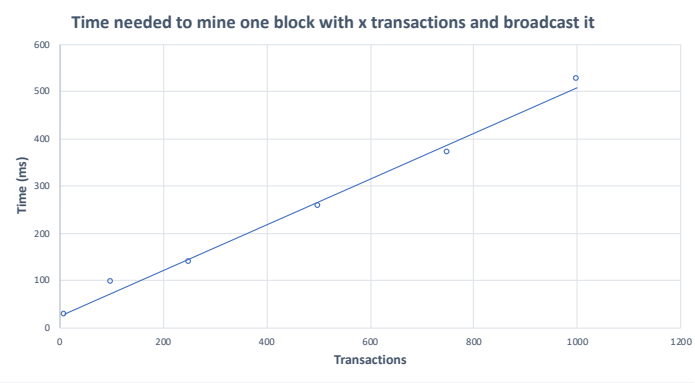

Figure 12 : Time needed to mine one block with x transactions and broadcast it

In Figure 12 we also see that it takes only a few milliseconds to broadcast a block with a lot of transactions.

C. Future work

In order to have more significant tests, we should try this consensus algorithm with more than 2 computers to see how the time increased and we should also try to do it through the internet and not only in LAN.

V. CONCLUSION

The Blockchain is considered today as a technology that can revolutionize the way we make and store transactions or information. It proposes an alternative to the low trust third-party by proposing a distributed ledger. It is considered today as a technology very secure thanks to the hash function that make everything work. However, we do not know if these hash functions are really secure and if there is no way to find collision. It exists a lot of possible troubles in PoW like the double spending attack for example but it is still considered as a very secure consensus algorithm. Crypto currencies using PoW have found ways to prevent these ones by developing other form of PoW. Even if it is really famous and trustable in the case of public Blockchain like the bitcoin, PoW have some disadvantages that could be very problematic in some cases. For example, the power consumption that is necessary to find the puzzle answer or the large time between each block. That's the reason why a lot of other consensus algorithm have been developed the last few years. The PoS is one of them and is based on the stake of each node. It exists a lot of different consensus algorithm that are an evolution of these 2 famous one. There are also other consensus algorithms which have nothing to do with them and that we haven't talk in this paper such as the Proof of Luck or Proof of Time. We cannot say that a consensus algorithm is better than another, it depends on each Blockchain characteristic. A new consensus algorithm has been invented to be suitable for a private blockchain inspired of the Proof of Luck. It is possible to change a lot of consensus algorithm in order to adapt them for a particular Blockchain.

VI. REFERENCES

[1] A. Miller, A. Kosba, J. Katz, and E. Shi, "Nonoutsourceable scratch-off puzzles to discourage bitcoin mining coalitions," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, 2015, pp. 680-691.

[2] Giang-Truong Nguyen and Kyungbaek Kim," A Survey about Consensus Algorithms Used in Blockchain", 2018

[3] Andreas M.Antonopoulos, Mastering Bitcoin : programming the open BlockChain, 2nd edition, 2017

[4] Xiaoyun Wang and Hongbo Yu, "How to break MD5 and other hash functions",

[5] I. Eyal and E. G. Sirer, "How to disincentivize large bitcoin mining pools," 2014

[6] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," in *Financial Cryptography and Data Security*. Heidelberg: Springer, 2016, pp. 142-157.

[7] S. King and S. Nadal, "PPcoin: peer-to-peer crypto-currency with proof-of-stake," 2012 [Online]. Available: https://decred.org/research/king2012.pdf.

[8] Elliptic curve digital signature algorithm, 2017 [Online].Available:https://en.bitcoin.it/wiki/Elliptic_

[9] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008 [Online]. Available: https://bitcoin.org/ bitcoin.pdf.

[10] M. Milutinovic, H. Wu, W. He and M. Kanwa, "Proof of Luck: an Efficient Blockchain Consensus Algorithm", 2017, [Online]. Available: https://eprint.iacr.org/2017/249.pdf