Permissioned vs. Permissionless Blockchain: How and Why There Is Only One Right Choice

Siamak Solat[1, 2,*], Philippe Calvez[1], Farid Naït-Abdesselam[2, 3]
[1] ENGIE, Laboratory for Computer Science and Artificial Intelligence (CSAI), Paris, France
[2] University of Paris, France.
[3] University of Missouri Kansas City, USA.

* Corresponding author. Email: siamak.solat@engie.com
Manuscript submitted September 10, 2020; accepted December 25, 2020.
doi: 10.17706/jsw.16.3.95-106

Abstract: The blockchain structure can be utilized in an efficient way to verify and ensure the integrity of the data and prevent historical transactions from tampering, such that if one block is altered, then the rest of next blocks are no longer valid. However, it is possible to re-calculate all the next blocks in a closed and permissioned network, such that all altered blocks will become valid again. In other words, only chaining blocks to each other, based on the hash of the previous one, cannot ensure the integrity and the security of the data and protect transactions against tampering. In this paper, we motivate why permissionless blockchain in an open network should be considered as the only acceptable type of blockchain and show that permissioned blockchain and closed network makes chaining transactions (i.e. blockchain) as an unhelpful structure. Of course, this never means that we ignore the existing problems in permissionless blockchains (such as, scalability of the network and transaction throughput), but also, we argue that permissioned blockchain cannot be an acceptable solution for those problems, where chaining transactions will no longer be meaningful. In other words, in a closed network, we no longer need to chain transactions. Although, this is a central issue and has been addressed in recent years, we are trying to bring new elements and an enriched point of view.

Key words: Blockchain, tamper-evidence, tamper-resistance, data-integrity, proof-of-work.

1. Introduction

Blockchain is a type of distributed ledger which consists in a read and append-only distributed database that maintains a list of records, called blocks, and can be secured from tampering and revision in an open and permissionless network. A decentralized consensus can be achieved using specific algorithms such as proof-of-work, proof-of-stake or Byzantine Fault Tolerant (BFT) consensus algorithms. Blockchains can be used in a wide variety of use cases, such as monetary transaction like Bitcoin [1], medical records [2] or smart vehicular security [3] etc. The lack of native support for advanced programmability in early blockchain deployment, such as Bitcoin, encouraged the development of a new generation of blockchain, extending the semantic of transaction through a program, called smart contract, written in a Turing-complete language, such as Solidity [4]. The smart contracts can process data on-chain to implement complex business rules. They can be useful in automating business processes in a trusted way, by allowing all stakeholders to process and validate contractual rules as a group [5].

The blockchain is today a hyped term that may have different meanings with variant contexts. The noise and hype can lead to the increase of misunderstandings about the blockchain and result in implementing many applications based on some incorrect assumptions and hypotheses. This leads usually to incorrect applications disconnected from the real blockchain capabilities. In fact, without understanding the philosophy behind the chaining transactions and the main features of blockchain structure, it cannot be utilized in a correct way.

In this paper, we argue that a permissioned blockchain cannot achieve the goal of a blockchain system, where it does not allow open participation in either submitting transactions or participating in transaction validation process, such that sending a transaction needs some permission beyond mere possession of some way to pay transaction fees or participants cannot fairly expect the network to resist censorship. We also argue and explain that openness and being public and permissionless of a blockchain system is not optional, but it is a compulsory and necessary feature for a blockchain network. Some permissionless blockchain networks, such as Bitcoin and Ethereum, use proof-of-work (PoW) to defeat the Sybil attack and prevent validators from creating many spam blocks. PoW is usually recognized as a "consensus" mechanism, but in fact, it only forces validators to consume some resources (mainly electricity) to defeat the Sybil attack. As an alternative, several "Proof-of-X" mechanisms are proposed (such as proof-of-stake (PoS), proof-of-space-time (PoST) and many others). We accentuate that we never argue that the only solution to prevent Sybil attack is PoW, but also we criticise the existential philosophy of permissioned blockchains.

Due to the problems of scalability of the network and transaction throughput in permissionless blockchains, various types of solutions have been proposed. These solutions are categorised as follows: (I) alternative consensus mechanisms to PoW (such as PoS etc.) (II) off-chain and second layer protocols (such as Bitcoin Lightning Network) (III) parallel verification of transactions (such as Sharding blockchain [6] [7] [8]) and (IV) limiting the number of participants in a closed network controlled by a centralized entity: either a single person or company, or a consortium of known entities bound by contracts in a permissioned blockchain [9]. In this paper, contrary to the common belief that both permissioned and permissionless blockchains have their pros and cons, we will argue that although permissionless blockchains have serious issues (such as scalability, throughput etc,) however, permissioned blockchain cannot be recognized as a solution to those issues, since chaining transactions in a closed network is no longer able to protect the data from tampering. Thus, if the network is closed, we no longer need for chaining transactions (i.e. blockchain structure).

The rest of this paper is organized as follows: In section 2, we define permissionless or public, and permissioned or private blockchains, and then explain the differences between them. In section 3 we bring up various contradictory opinions regarding whether permissioned blockchain can be interpreted as a right alternative solution to defeat the existing issues in permissionless networks. We then in section 4 define the term "immutability" and explain misconceptions about this word and then describe the required steps to achieve the tamper-resistant and tamper-evident data based on blockchain structure. We thereafter in section 5 criticise the reasonability and the existential philosophy of the permissioned blockchain and also argue that why only chaining blocks and transactions without forcing validators to consume resources cannot be helpful to protect the data against tampering. For this purpose, we implemented a code in Java, based on the functionality of PoW and to show that the chained blocks can be entirely replaced by an altered chain, either in the absence of a Sybil attack prevention mechanism or in a closed and permissioned network. Although, it does never mean that we only support PoW (as a Sybil attack prevention mechanism), but also we mean that the solution of scalability and throughput limitation in permissionless blockchains is not closing the network to achieve less participants and validators through giving permissions to join the network by an authoritative entity. That is, in case of either PoW or any other alternative mechanism to PoW, the blockchain network must remain open and permissionless; otherwise, chaining blocks is no longer useful to protect the data from tampering. We finally conclude the paper in section 6.

2. Permissionless vs. Permissioned Blockchain

In the following, we explain the differences between permissionless or public and permissioned or private blockchains.

2.1 Permissionless or Public Blockchain

A blockchain is permissionless or public if participation in the submission and the confirmation of transactions is permitted to everyone. There is no special permission for submitting transactions beyond the possession of some way to pay transaction fees. Everyone also is permitted to participate in the validating transactions process in order to be selected as a validator. This must be in actual practice accessible to everyone who makes a reasonable attempt to earn it. Everyone who sends validly signed transactions to the network should be capable of fairly expecting the network to execute without having to concern that some particular group or entity can decide to prohibit their transactions in particular.

2.2. Permissioned or Private Blockchain

A blockchain is permissioned or private if it does not permit open participation in either submitting or participating in the transactions validation process, such that sending a transaction needs some permission beyond mere possession of some way to pay transaction fees or participants cannot fairly expect the network to resist censorship, meaning that not all participants have practical guarantee that their transactions would not be discriminated against in a way that considerably has an effect on their potency to leverage the network and get its profits.

Another point worth to note is that any permissioned blockchain is private and any permissionless blockchain is public, as when we say "public", it means "public to use" and not "public to view history of transactions", because a "private blockchain" can be "public to view" too, but it is not "public to use". This misconception is seen in some articles such as [10].

3. Related Works and Motivation

There are contradictory opinions regarding whether permissioned blockchain can be interpreted as a right alternative to overcome the existing issues in permissionless networks. Some researchers do not recognize permissioned blockchain as a right solution to the existing problems in permissionless blockchain [9]. Authors in [11] suggest that permissioned blockchains will not even be able to economize on costs relative to permissionless networks. Nevertheless, the common belief is still that the permissioned blockchain can be an alternative solution and is able to solve some part of the problems in permissionless blockchains. That is, such kind of blockchain is still widely utilized as a solution to the problems in permissionless networks, such as network scalability, transaction throughput, costs and fees, etc. the reason that motivated us to write this paper to demonstrate that chaining transactions in a closed network be helpful for the data security and tamper-resistance.

In [12], the authors believe that as permissioned blockchains operate under the supervision of a trusted consortium of validators, thus the cost of transactions verification is significantly reduced, whereas, in [11] has been demonstrated that permissioned systems are not even effective for cost savings. Belotti, Marianna, et al. [13] draw the key requirements when passing from permissionless to permissioned blockchains. Authors in [10],[14]–[18] believe that a blockchain, regardless of being permissionless or permissioned, and in any cases, is decentralized and works as a public distributed ledger. [19] recognizes permissioned blockchain as a right alternative to permissionless blockchains, particularly, for business applications in which the participants need some means of identifying each other. Authors in [20] believe that permissioned blockchains solve low performance and limited data confidentiality capabilities, but in the cost of sacrificing complete decentralization and involving more trust assumptions. Authors in [21] admit that a permissioned blockchain is not completely trust-less and also transactions can be rolled back by a centralized agency with override authority. Authors in [22] recognize permissioned blockchains as a system in which only an authorized set of entities are permitted to write and read the data, where they also compare a permissioned blockchain with a centralized database.

4. All Blockchains Are Not Necessarily Tamper-Resistant

Before anything, we define the term "immutability", since it is widely used as a blockchain's key feature in many papers [23]–[28]. We define "immutability" as tamper-resistance since nothing in nature is unchanging over time. The proof-of-work that is used in blockchain systems, unlike conventional definitions of the term "consensus", never achieves a state from which it cannot be rolled back, at least theoretically. The "tamper-resistance" feature, however, is achievable thanks to the main feature of a blockchain system: decentralization altogether with Sybil attack prevention mechanisms, (such as proof-of-work or any other alternative approaches), of course, where the network is permissionless and open to everyone to join.

Figure 1 shows how PoW (as a Sybil attack prevention mechanism) along with decentralization, altogether can make a blockchain system tamper-resistant. In this figure, while miners who hold the majority of computational power are working on creating block #49, a group of miners who hold the minority of computational power intend to alter a transaction in block #29. However, they face two major problems: a) having to consume a huge amount of resources (including electricity and specific required hardware) thanks to the proof-of-work, decentralization and openness of the network, and b) having to spend significant time. In practice, all these conditions result in the minority group not being successful to create block #49 sooner than the majority of miners who follow the protocol as described. The reason is that the group of attackers have to re-calculate all the computations for blocks #29 to #48, and then work on block #49, which means 21 blocks vs. only 1 block that should be computed by honest miners. The difficulty of computation is adjusted based on the total computational power of the network.

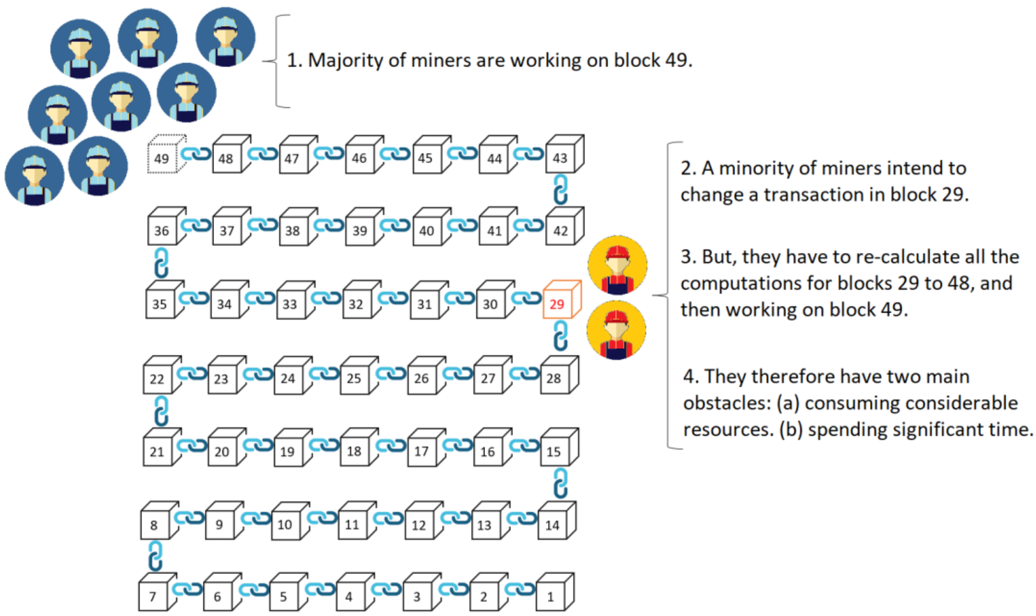

Fig. 1. How proof-of-work and decentralization altogether make blockchain tamper-resistant.

Bitcoin's blockchain was introduced as a peer-to-peer system aimed at removing Trusted Third Party (TTP) in the transactions between participants. For this purpose, Bitcoin uses proof-of-work as a blockchain consensus. The proof-of-work as a dominant consensus mechanism is used in more than 90% of the total market capitalization of existing digital currencies [29]. It was proposed in Hashchash [30], essentially amounts to brute-forcing a hash inequality based on SHA-256, as a CPU cost-function that computes a token which can be used as a denial-of-service counter-measure. That's why it is called "proof-of-work", which means that the miners who intend to do a job, (such as creating a new block in a blockchain system or sending email for non-blockchain use cases) have to prove that they have done some "work". The "work", in case of blockchain system, is doing computations to solve a cryptographic puzzle that is difficult to solve but easy to verify, through consuming resources (such as electricity and hardware equipment like CPU and processors). This prevention mechanism aims to prevent security attacks such as DoS or Sybil attacks.

In the following, we have brought up a **general** PoW-based example (and not for a specific platform, such as Bitcoin or Ethereum etc.) to show that how PoW, as a Sybil attack prevention mechanism, can make historical transactions tamper-resistant.

By the following scenario, we'll show that only chaining transactions to each other based on the hash of the previous one, cannot protect the data from tampering. In order to avoid potential misunderstanding, we would like to accentuate that we never argue that PoW is the only acceptable mechanism to prevent historical transactions being tampered, but also we show that only chaining transactions without a Sybil attack prevention mechanism cannot protect the data from tampering. As in a permissioned blockchain, the network is closed and the validators are known and identified by their signatures in a framework of legally valid contracts, thus, the factor that prevents transactions from tampering is not chaining transactions (i.e. blockchain), but the combination of signatures and contracts, where blockchain becomes an useless and unrequired structure in such closed network.

4.1. Block and Hash Function

A hash function is a mathematical function transforming any arbitrary input into a string with a set of numbers and letters, such that any slight change in the input creates a completely new output hash. In following examples, we use SHA-256 hash function, and the term "token" can be interpreted as a crypto-currency or any kind of crypto-token.

SHA-256(Blockchain Should Be Permissionless.) =
d1970de7ed25a6b488436edc1a479594203953d9e1e949f42fd08733d7f4e374

where, with a negligible change in the input, the output's hash will be changed entirely. Assuming following example, where $user_a$ sends 1 token to $user_b$:

$transaction_i$: $user_a$ sent 1 token to $user_b$ hash:
6b7fb9a011b4ee52c1c4771646c8d273909729688d7bb65499222022235e6fc1

Now, if $user_b$ changes $transaction_i$ as follows: "$user_a$ sent 2 tokens to $user_b$" (rather than 1 token) then the hash of $transaction_i$ will be changed completely as below:

$transaction_i$: $user_a$ sent 2 tokens to $user_b$ hash:
fa3007ff650fc3e4e7a4efd16c892692ee2d1680e2c8239d0a17217701d1fce0

Thus, tampering $transaction_i$ is simply "detectable" thanks to the hash function. Although, historical transactions are now "tamper-evident" using hash; however, altering transactions does not have still noticeable costs, meaning that historical transactions are not yet “tamper-resistant”.

tamper-evident ≠ tamper-resistant

4.2. Chaining Transactions

Thus, to make it more costly to alter a transaction, the record of each transaction is chained to the previous one:

…

$transaction_{i-1}$: $user_x$ sent 1 token to $user_y$

hash: SHA-256($transaction_{i-2}$ || $transaction_{i-1}$)

$transaction_i$: $user_a$ sent 1 token to $user_b$

hash: SHA-256($transaction_{i-1}$ || $transaction_i$)

$transaction_{i+1}$: $user_z$ sent 1 token to $user_k$

hash: SHA-256($transaction_i$ || $transaction_{i+1}$

…

Now, if $user_b$ still intends to change $transaction_i$, then all transactions after $transaction_i$ will no longer be valid, such that $user_b$ will have to re-calculate all the computations from $transaction_{i+1}$ to the end of the chain, while at the same time the chain is growing by newer transactions. Nevertheless, $user_b$ is motivated enough to spend a long time to re-calculate all the hashes to earn more tokens. Therefore, the chained transactions are not still tamper-resistant. To achieve this goal, an additional step is required to make it, in practice, impossible for an adversary to alter the historical transactions.

Algorithm 1 proof-of-work algorithm

```
nonce ← 0
txHashInHex ← null                      ▷ [tx hash in hexadecimal.]
txHashInDec ← 0                         ▷ [tx hash in decimal.]
target ← enter target to determine difficulty of PoW!
txPlusNonce ← null                      ▷ [tx concatenated with nonce.]
txString ← content of transaction
while true do                           ▷ [start an infinite loop.]
    nonce ++                            ▷ [nonce increases after each round.]
    txPlusNonce = txString + nonce
    txHashInHex ← SHA-256(txPlusNonce)
    txHashInDec ← hexToDec(txHashInHex)
    if txHashInDec < target then
        PoW is solved                   ▷ [current nonce is answer.]
        break                           ▷ [break infinite loop.]
    end if
end while
return nonce
```

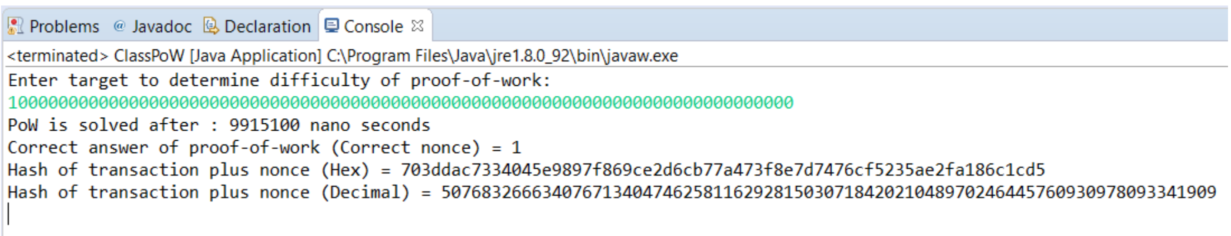

Fig. 2. The proof-of-work was solved after 0.0099151 seconds, where the Target had been initialized to 10^{77} run by a computer with a processor of i7-8650U (Processor Base Frequency: 1.90 GHz, Configurable TDP-up Frequency: 2.10 GHz) using a Java code, based on algorithm 1.

4.3. Nonce, Proof-of-Work and Mining

The next step is to add a cryptographic puzzle to be solved by the validators of transactions. In this way, a number will be added after each transaction, called “nonce”. The nonce can be assumed as a counter concatenated to the transaction data. The hash therefore will be generated from the entire this new buffer.

To show that only chaining transactions is not able to protect transactions from tampering, we implemented a code in Java, based on the functionality of PoW and SHA-256 hash function. Our code is accessible through a Github repository in [31]. This code has been implemented based on algorithm 1. The hash generated in hexadecimal will be then converted to decimal as a BigInteger in an infinite loop, in which there is a condition, comparing the decimal value of the hash with a variable, called “Target”. If the hash would be smaller than the Target, then the PoW is solved and related nonce is the correct answer of PoW.

If we consider a smaller Target, the PoW becomes more difficult, as the range of the sample space including the correct answer of PoW will be smaller, resulting in decreasing the probability that the generated hash is among the numbers of the sample space. By adding a timer to our code, we calculated the time it takes for a single hash computation using an ordinary computer with a processor i7-8650U (Processor Base Frequency: 1.90 GHz, Configurable TDP-up Frequency: 2.10 GHz) [32]. By initializing the Target variable to 10^{77}, the PoW was solved after a single attempt, and it took 0.0099151 seconds. Fig. 2 shows the screenshot of the output of our code.

4.4. Decentralization

Bitcoin’s blockchain in date of 2020/09/09 consists of 647,471 blocks [33]. A miner device such as AntMiner S9 designed especially for SHA-256 is able to compute almost 14×10^{12} hashes per second [34], meaning that it can compute a single hash in $1/14\times10^{-3}$ nanoseconds, that means by using such device it could be feasible to re-calculate all blocks’ hashes of Bitcoin in only ≈ 4.6 seconds, **if PoW is removed from Bitcoin**. This fact simply shows that Bitcoin’s blockchain without PoW with enough difficulty (as a resource consuming mechanism to defeat Sybil attack) is neither tamper-resistant nor tamper-evident. We accentuate again that we never motivate the only solution to prevent Sybil attack and making blockchain system tamper-resistant is PoW, but also, we argue that to make chaining blocks reasonable, the network should remain permissionless, either with use of PoW or any other Sybil attack prevention mechanism. As in a closed and permissioned network, the blocks are added to the blockchain using a couple of trusted and known nodes based on their signatures, well, we no longer need to chain the blocks and transactions, because after replacing the chain entirely by a tampered one, all blocks are re-calculated based on the previous block’s hash, and thus, all of them are valid again. In such network, tampering can be detectable by other mechanisms such as timestamp and digital signatures, but chaining transactions (i.e. blockchain) is no longer required to either detect or protect the data against tampering.

This fact shows that only chaining transactions using the hash of previous records cannot prevent or even detect altering historical transactions. Of course and obviously if the network would be closed and permissioned, we no longer need a PoW (or similar Sybil attack approaches), such that re-calculation of all blocks’ hashes are feasible by an authoritative entity which gives the permissions to the participants for either submitting or validating transactions. This authoritative entity can consist of multiple trusted and known participants whose trust is based on a legally valid contract, where, although, the transactions may become tamper-resistant using these legally valid contracts and signatures, but not at all due to chaining transactions (i.e. blockchain). If one claims that the blockchain can be used in a permissioned and closed network only in order to achieve tamper-evidence feature (and not necessarily being tamper-resistant), then the answer is that if the whole of blockchain’s hashes would be re-calculated, then no one can detect any alter thanks to the blockchain structure (as all blocks have been re-calculated based on the previous one and so all of them are valid again).

The difficulty of PoW, as an important factor of defeating Sybil attack, is adjusted based on total computational power of the network. Thus, to find the correct answer of PoW, the miners require to know the correct "Target" value. In Bitcoin, the miners are able to compute the current difficulty using the data about the previous blocks. That is, each 2,016 blocks should be created in two weeks [35], [36]. If this time is different, then the current difficulty will be multiplied by:

$$\frac{\text{14 days}}{\text{time spent for 2016 blocks}} \tag{1}$$

to adjust and find the correct difficulty.

By having the difficulty, the Target is found using the following equation:

$$\text{Difficulty} = \frac{\text{maximum target}}{\text{current target}} \tag{2}$$

According to the Bitcoin protocol, the maximum Target value is $\approx 2.7\times10^{29}$ [37]. By using this value for the Target, the proof-of-work will be in its easiest difficulty level. Using the above procedure and by adjusting the difficulty of PoW, it will become extremely difficult for an adversary to alter the historical transactions.

The mining process has a significant cost for the miners (i.e. considerable electricity consumption along with providing requirements such as CPU or GPU etc). Thus, to motivate the miners for performing hashing calculations to find the correct answer of PoW, they receive some rewards for every new block generation. The miners' reward is not limited to only "block reward", but they also get the fees for every transaction that users pay. Logically, to achieve more reward, the miners usually will arrange transactions with higher fees to be inserted in the blockchain, resulting in transactions with fewer fees might have to wait a long time to be validated.

To make the history of transactions tamper-resistant, all above steps are required. All blocks are linked to each other using their hash value and if someone intends to change the content of one block, they have to re-calculate the hash of all the next blocks that considering an enough difficult PoW, in practice, it is impossible or at least much too difficult.

5. Chaining Transactions in Permissioned and Closed Network Cannot Prevent Tampering Data

As a result of the above explanations as well as definitions in section 2, since in a permissioned blockchain the historical transactions are maintained and updated in a centralized manner, in addition to losing many advantages of blockchain, in general, chaining transactions can no longer be helpful for neither tamper-evidence nor tamper-resistance properties. In fact, two reasons that cause the data in a blockchain system having more authenticity are "decentralization" and "transparency". A permissioned blockchain may keep transparency; however, can harm significantly decentralization. If the authoritative and privileged nodes go rogue, or fail to reach consensus, the network will become collapsed. The value of a permissioned blockchain would have to come from some benefits of centralization, for example, accelerating transactions throughput, but at the cost of a very high level of trust in the trusted, permitted, and privileged nodes. However, the most important point is that in a permissioned blockchain, chaining transactions can no longer be useful and thus, becomes an unrequired operation, as the whole of the blockchain can be recalculated and replaced by a new one, where all transactions will be valid again, as we explained in section 4.

Nevertheless, we never deny the existing problems in permissionless blockchains (such as scalability issues, low throughput, latency etc.), but also, we try to motivate that the solution of those issues is not permissioned blockchain and closing the network, as this kind of blockchain makes chaining blocks unmeaningful. A permissioned blockchain, obviously, requires permission to join, and thus a proof-of-work or any other Sybil attack prevention mechanisms is not required, as in such a network, if a node misbehaves, they can be eliminated from the network by whatever process prevents the whole world from joining.

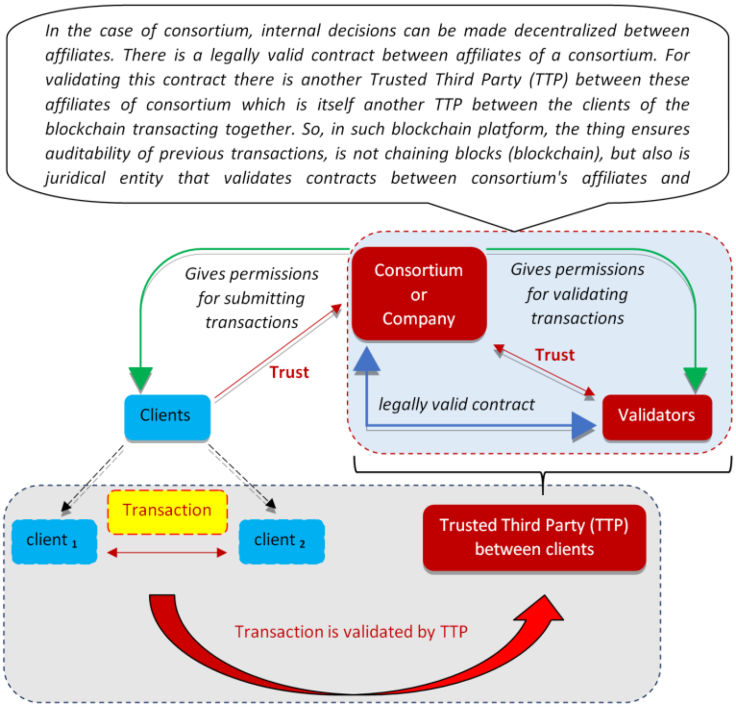

Fig. 3. A high level view of the permissioned blockchain architecture which shows that the transactions between parties are validated by a centralized Trusted Third Party (TTP).

Fig. 3 shows a high level view of the permissioned blockchain architecture and how transactions between clients are validated by a centralized Trusted Third Party (TTP). A permissioned blockchain could be extended with the rule, for example: a particular transaction must be ignored. or the tokens in a particular address must be considered to be in another address. Then, effectively altering and tampering the historical transactions is done, where the central authoritative entity (as a company or as a consortium) is able to force its validators to accept such alterations, as a permissioned blockchain, by definition, have a mechanism to hire or fire the validators, persons or entities that are permitted to append new blocks [9]. This mechanism is controlled by a centralized entity: either a single person or company, or a consortium of known and trusted entities bound by legally valid contracts. These entities are trusted by all participants using contracts that are legally valid by another Trusted Third Party (TTP) [9].

An organization that uses a permissioned blockchain, it could instead use traditional replicated database without chaining them, where the factor that ensures transactions would be tamper-resistant is no longer chaining transactions, but also in such centralized and trust-based network, transactions are protected against tampering thanks to legally valid contracts, Trusted Third Parties (TTP) as well as signatures. In such a system, also, tamper-evidence of the data cannot be ensured due to chaining transactions (i.e. blockchain), as once the whole of the chain is re-calculated and replaced by a new one, via an authoritative entity in the network, then it is not blockchain structure which proves that an alteration has occurred, but also tampering transactions is detectable thanks to the validators' signatures, as the validators are known and identified by their signatures and therefore what they alter in the chain, it will be logged along with their signatures; as a result, tampering transactions is detectable without need for chaining blocks and transactions (i.e. blockchain structure) in a closed network.

6. Conclusion

In this paper, we argued that although permissionless blockchains have serious issues (such as scalability, throughput etc,) however, permissioned blockchain, contrary to the common belief, cannot be recognized as a right solution to those issues, as chaining transactions in a closed network is no longer able to protect the data from tampering. In this way, we implemented a code in Java, based on the functionality of PoW to show that the chained blocks can be entirely replaced by an altered chain, either in the absence of a Sybil attack prevention mechanism or in a closed and permissioned network. Thus, if the network is closed and permissioned, we no longer need for chaining transactions (i.e. blockchain structure). Consequently, chained blocks in a closed and permissioned network can bring us nothing and chaining blocks only in an open and permissionless network can seem reasonable. We emphasize that we never argued that the only solution to prevent Sybil attack is PoW, but also we criticised the existential philosophy of permissioned blockchains. In other words, we never ignored the existing problems in permissionless blockchains (such as, scalability of the network, transaction throughput etc.) but also, we argued that permissioned blockchain cannot be an acceptable solution for those problems.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Siamak Solat: Conducted the primary research and drafted the article. Philippe Calvez and Farid Naït-Abdesselam: Provided active feedback on the drafts for revision and supervised the workflow of the research. All authors had approved the final version.

References

[1] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[2] Azaria, A., Ariel, E., Thiago, V., & Andrew, L. (2016). Medrec: Using blockchain for medical data access and permission management. *Proceedings of the 2016 2nd International Conference on Open and Big Data (OBD)* (pp. 25-30).

[3] Dorri, A., Steger, M., Kanhere, S. S., & Jurdak, R. (2017). Blockchain: A distributed solution to automotive security and privacy. *IEEE Communications Magazine*, *55(12)*, 119-125.

[4] Ethereum Solidity Documentation. Retrieved from: https://solidity.readthedocs.io/en/v0.7.1/

[5] Hull, R., Batra, V. S., Chen, Y. M., Deutsch, A., Heath III, F. F. T., & Vianu, V. (2016, October). Towards a shared ledger business collaboration language based on data-aware processes. *Proceedings of the International Conference on Service-Oriented Computing* (pp. 18-36).

[6] Luu, L., *et al*. (2016). A secure sharding protocol for open blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*.

[7] Kokoris-Kogias, E., *et al*. (2018). Omniledger: A secure, scale-out, decentralized ledger via sharding. *2018 IEEE Symposium on Security and Privacy (SP)*.

[8] Team, Z. (2019). The zilliqa technical whitepaper.

[9] Jorge, S., (2018). Nistir 8202 (DRAFT): Blockchain technology overview. Institute of Computing State University of Campinas.

[10] Taskinsoy, J. (2019). Blockchain: A misunderstood digital revolution. *Things You Need to Know about Blockchain*. Things You Need to Know about Blockchain.

[11] Gans, J. S., & Neil, G. (2019). *More (Or LESS) Economic Limits of the Blockchain*. National Bureau of Economic Research.
[12] Maple, C., & Jack, J. (2018). Selecting effective blockchain solutions. *Proceedings of the European Conference on Parallel Processing*.
[13] Belotti, M., *et al*. (2019). A vademecum on blockchain technologies: When, which, and how. *IEEE Communications Surveys and Tutorials 21.4*. 3796-3838.
[14] Berryhill, J., Bourgery, T., & Hanson, A. (2018). *Blockchains Unchained: Blockchain Technology and Its Use in the Public Sector, OECD Working Papers on Public Governance*. No. 28, OECD Publishing, Paris.
[15] Cascarilla, C. G. B. (2015). Blockchain, and the future of financial transactions. *CFA Institute Conference Proceedings Quarterly*.
[16] Duffie, D. (2019). *Digital Currencies and Fast Payment Systems*. Mimeo, Stanford University.
[17] Dwyer, G. P. (2015). The economics of Bitcoin and similar private digital currencies. *Journal of Financial Stability*, *17*, 81-91.
[18] Pike, C. (2018). *Blockchain Technology and Competition Policy-Issues Paper by the Secretariat*.
[19] Vukoli´c, M. (2017). Rethinking permissioned blockchains. *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*.
[20] Falazi, G., Hahn, M., Breitenbücher, U., Leymann, F., & Yussupov, V. (2019). Process-based composition of permissioned and permissionless blockchain smart contracts. *Proceedings of the 2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC)*.
[21] Liu, M., Kean, W., & Xu, J. J. (2018). How will blockchain technology impact auditing and accounting: Permissionless versus permissioned blockchain. *Current Issues in Auditing 13.2*, A19- A29.
[22] Wüst, K., & Arthur, G. (2018). Do you need a blockchain? 2018 Crypto Valley Conference on Blockchain Technology (CVCBT).
[23] Puthal, D., *et al*. (2018). The blockchain as a decentralized security framework [future directions]. *IEEE Consumer Electronics Magazine 7.2*, 18-21.
[24] Sankar, L. S., Sindhu, M., & Sethumadhavan, M. (2017). Survey of consensus protocols on blockchain applications. *Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems*.
[25] Zheng, Z., *et al.* (2017). An overview of blockchain technology: Architecture, consensus, and future trends. *Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress)*.
[26] Underwood, S. (2016). *Blockchain Beyond Bitcoin*. 15-17.
[27] Xu, X. W., *et al*. (2016). The blockchain as a software connector. *Proceedings of the 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*.
[28] Hackius, N., & Moritz, P. (2017). Blockchain in logistics and supply chain: trick or treat?. *Proceedings of the Hamburg International Conference of Logistics (HICL)*.
[29] Gervais, A., *et al.* (2016). On the security and performance of proof of work blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*.
[30] Back, A. (2002). *Hashcash-A Denial of Service Counter-Measure*.
[31] Implementing PoW in Java Github url. Retrieved from: https://github.com/ngafep/ Proof-of-Work-SHA-256-in-Java/blob/master/PoW_Java.java
[32] Intel® Core™ i7-8650U. Retrieved from: Processor.https://ark.intel.com/content/www/us/en/ark/products/124968/intel-core-i7-8650u-processor-8m-cache-up-to-4-20-ghz.html
[33] Number of blocks of Bitcoin at 2020/06/12. Retrieved from: https://www.blockchain.com/btc/block/647471

[34] Mining hardware comparison. Retrieved from: https://en.bitcoin.it/wiki/Mining_hardware_comparison
[35] What is the minimum difficulty? Retrieved from: https://en.bitcoin.it/wiki/Difficulty#What_network_hash_rate_results_in_a_given_difficulty.3F
[36] Garay, J., Aggelos, K., & Nikos, L. (2015). The bitcoin backbone protocol: Analysis and applications. *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*.
[37] What has the target been in the past? Retrieved from: https://en.bitcoin.it/wiki/Target#What_is_the_maximum_target.3F

Siamak Solat received a master's degree in computer science and networks from Sorbonne University, Paris, France. He is working on his PhD research at University of Paris, jointly with the Laboratory for Computer Science and Artificial Intelligence (CSAI), ENGIE, on how to enhance the throughput and the performance of distributed networks based on parallelism and sharding techniques.

Farid Naït-Abdesselam received his state engineering degree in computer science from Algiers University of Science and Technology (Algeria) in 1993 and his MSc degree in computer science from Paris University (France) in 1994. He joined the University of Versailles (France) in 1996 where he obtained in 2000 his Ph.D degree in computer science.

In 1998, he was a visiting research associate at the University of Western Ontario (Canada) where he worked on distributed interactive virtual environment and multimedia communications over high-speed networks. From 1999 to 2000 he was an assistant professor at the University of Lille (France). From 2000 to 2003 he was an associate professor at INSA of Lyon and a researcher at INRIA Rhône Alpes. In 2003, he moved as an associate professor to the University of Lille and was till 2007 a researcher at INRIA Lille Nord Europe. In 2009, he obtained his Habilitation to Supervise Research (HDR) in Mathematics and Informatics from the University of Lille and then promoted in 2010 as full professor position at Paris University. In 2018 he was a research professor at Iowa State University (USA) and since 2019 he is a professor at the University of Missouri Kansas City (USA).

Philippe Calvez is the head of the csai Lab of engie (R&D Corporate Centre of Engie) in which he leads activities and research projects on topics such as machine learning (computer vision, NLP), digital interoperability (knowledge representation and reasoning, semantic technologies), AI for Distributed Autonomous Energy System. Dr. Calvez is the Coordinator of the H2020 Project PLATOON (Digital PLAtformand analytical TOOls for eNergy). He is a member of the Steering Committee of the Eclipse DLT Tangle EE Working Group initiative.