mathematics

MDPI

Review

Blockchain: Consensus Algorithm Key Performance Indicators, Trade-Offs, Current Trends, Common Drawbacks, and Novel Solution Proposals

Yaçine Merrad [1], Mohamed Hadi Habaebi [1,*], Elfatih A. A. Elsheikh [2], Fakher Eldin. M. Suliman [2], Md Rafiqul Islam [1], Teddy Surya Gunawan [1] and Mokhtaria Mesri [3]

[1] Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lupur 53100, Malaysia; yacinechoupot@yahoo.fr (Y.M.); rafiq@iium.edu.my (M.R.I.); tsgunawan@iium.edu.my (T.S.G.)

[2] Department of Electrical Engineering, College of Engineering, King Khalid University, Abha 61421, Saudi Arabia; eelsheikh@kku.edu.sa (E.A.A.E.); fmsuliman@kku.edu.sa (F.E.M.S.)

[3] Department of Electronics, University Amar Télidji of Laghouat, Laghouat 03000, Algeria; m.mesri@lagh-univ.dz

* Correspondence: habaebi@iium.edu.my; Tel.: +60-3-93290652

Abstract: Consensus protocols stand behind the success of blockchain technology. This is because parties that distrust each other can make secure transactions without the oversight of a central authority. The first consensus protocol emerged with Bitcoin. Since then, many others have appeared. Some of them have been implemented by official blockchain platforms, whereas others, for the time being, remain as proposals. A blockchain consensus is a trade-off. The new solutions promise to overcome the known drawbacks of blockchain, but they may also bring new vulnerabilities. Moreover, blockchain performance metrics are not clearly defined, as some metrics, such as delay and throughput, which are key factors for the efficiency of standard networks, are purposely constrained by most mainstream blockchain platforms. The main body of this paper consolidates knowledge of blockchains, focusing on the seminal consensus protocols in large-scale market capitalization platforms, and how consensus is achieved for large-scale, decentralized, blockchain architectures. The benefits, limitations, and tradeoffs, as well as the subsequent trend in current consensus development, and its limitations as a general paradigm, are highlighted. The paper also sheds light on overlooked potential performance metrics, and it proposes some novel solutions to some of the identified problems.

Keywords: blockchain; consensus; decentralization; performance metrics; tradeoffs; proposals

MSC: 68M14

check for updates

Citation: Merrad, Y.; Habaebi, M.H.; Elsheikh, E.A.A.; Suliman, F.E.M.; Islam, M.R.; Gunawan, T.S.; Mesri, M. Blockchain: Consensus Algorithm Key Performance Indicators, Trade-Offs, Current Trends, Common Drawbacks, and Novel Solution Proposals. *Mathematics* **2022**, *10*, 2754. https://doi.org/10.3390/math10152754

Academic Editors: Hossein Hassani and Nadejda Komendantova

Received: 29 June 2022
Accepted: 25 July 2022
Published: 3 August 2022

1. Introduction

Blockchain is the underlying technology for most of the potential applications which have been gaining traction in recent times [1–3]. The blockchain network is a decentralized, distributed ledger technology wherein all nodes must agree on the validity of the ledger they are sharing. This is to ensure that any data included are perfectly valid; hence, after being committed to the blockchain, the data are impossible to delete, deny, or tamper with [4]. Blockchain is a transformation of the Byzantine Generals (BG) Problem, where at least two-thirds of the network should be willing to maintain the integrity of the network so that the impact of malicious parties can be nullified [5]. In order to agree on the new data (blocks to be added to the chain), and to ensure trust in the system, the nodes must rely upon a consensus protocol. These algorithms comprise the backbone of a blockchain; therefore, to a large extent, these algorithms determine the performance of such a system, and moreover, what the system can achieve. To date, increasing attention has been paid to a range of consensus protocols, and different cryptocurrencies have implemented different protocols. Consensus algorithms in blockchains have been subjected to numerous research studies and they are still open to new ones. Despite the fact that the proposed algorithms endeavor to strengthen some of the existing algorithmic weaknesses, they will nevertheless be subject to new constraints. Some consensus protocols are intensely robust, whereas others may cause centralization and trust issues, despite the fact that blockchains were created to avoid such problems [6].

In this paper, by studying blockchain consensus, we have attempted to produce a comprehensive and systematic categorization and analysis. Indeed, we aimed to define the main trend in blockchain consensus development, and we sought to identify its weaknesses. To this end, we intended to gather these algorithms and their related features, which, in some cases, are presented separately in the literature. The blockchain consensus protocols that are presented in this paper were not given equal attention, as we chose to prioritize the most established ones. First, the paper begins by giving an overview of blockchain, tracing its origins in the Bitcoin digital cash system, and explaining its different features; these include blockchain technology and architecture, services and applications, and how consensus regarding truth is reached in an environment that is often considered untrustworthy. This is achieved by ensuring the theoretical foundations and prerequisites are understandable. Thus, the structure of the article can be outlined as follows. The second section gives an overview of the technology, noting its fundamental concepts and primary applications. The third section presents a wide range of consensus algorithms that are used by typical blockchain platforms, including those which are used in both permissionless and permissioned blockchains, as the paradigms and trends followed are different in each one [7]. It also aims to provide an understanding of their strengths and weaknesses, as well as an understanding of the tradeoffs that emerge between different consensus approaches. Section 4 discusses blockchain consensus performance. Moreover, it puts forward the idea that fairness is an important performance feature in blockchain that tends to be overlooked. In the fifth section, the current common trend in consensus development is identified, along with some critics of the trend. Section 6 presents novel proposed solutions and new strategies for proof of work constraints, namely, the 51% attack, the miner transaction boycott, high energy consumption, and the Sybil attacks. Section 7 provides insight into quantum computing, which presents itself as an imminent threat that could cause problems for blockchain technology in its current form. Finally, Section 8 summarizes the concepts and analyses presented in this paper.

2. Overview of Blockchain Technology and Its Applications

As its name indicates, a blockchain is simply a chain of blocks containing information; these blocks are linked together in a chain-like structure, using cryptography. The term DLT (distributed ledger technology) is also used to refer to a blockchain. There have been rapid developments within blockchain technology, and recently, it has been subject to a great deal of scrutiny. This section will begin by attempting to provide a concise overview of this technology, and the widely known and noteworthy applications of blockchain will also be described.

2.1. Blockchain Concept as the Technology behind Decentralized Cryptocurrency

Blockchain first appeared as a technology that enabled the first decentralized cryptocurrency [8], where no bank or entrusted central authority was needed to confirm or deny transactions. Since then, it has expanded to other decentralized applications, leading to a paradigm shift in web application architecture and several revolutions in the industry. Blockchain is a peer-to-peer network that has no point of failure [9,10]. It combines technologies that have been established for decades in an innovative, ingenious way. It combines hashing technology (such as SHA256 in Bitcoin or Keccak256 in Ethereum), consensus algorithms, and digital signatures [4]. After the financial crisis of 2008, cryptocurrency emerged, and such currencies are based upon blockchain technology. This enables each issued transaction to be propagated to all nodes in the network and confirmed by the consensus algorithm [8,11]. Transactions comprise a data structure with several fields, namely, the addresses of the issuer and the recipient of the transaction, the amount to be transferred in coins, and the issuer′s digital signature; this ensures that the transaction was issued by its claimer [12]. When a peer connects to the blockchain (node), joins the network, and creates an account, it must generate a unique key pair using a well-known mathematical model, namely, elliptic curve cryptography [13]. This involves a public key that is publicly known and essential for signature verification, and a private key that is used to sign the transaction. It is the responsibility of each node to keep its private key secret so that transactions are not signed on its behalf [14]. The security of this system is based on the fact that each transaction can only be signed by the owner of the private key, and the signature can only be verified with the corresponding public key [8]. Moreover, the blockchain is tamper-proof, meaning that data or transactions can never be deleted, denied, or altered once they are stored in the blockchain ledger. It is imperative that all peers on the network have an identical copy of the most recent blockchain ledger. In this sense, the blockchain contains the history of all issued transactions, making it possible for all peers in the network to keep track of all account balances. The blockchain as a tamper-proof structure relies on two concepts [4]:

- The first concept: blocks are linked together through their hashes. To find a valid hash, miners need to solve a difficult mathematical puzzle, which requires hashing power. Commonly, the hash algorithm SHA256 [15] is used, though other secure hash algorithms can also be used. Other hash algorithms that can be used include Scrypt, which is used by Litecoin [16], and SHA256d, which is used by Peercoin [17]. In other words, any hash algorithm can be used as long as it complies with the following main principles: (i) different inputs may never lead to the same hash; (ii) when given a computed hash value, there is no mathematical way to derive the input that leads to a particular hash value, therefore, each new block hash is computed using the hash of the previous block, and thus, chain tampering can only be detected if all hashes are changed accordingly.
- The second concept: the tamper-proof property of the blockchain is based on the "longest chain wins" rule. This means that if someone is busy manipulating the ledger and changing the hashes accordingly, a new block must have been added during an interim period.

In addition, miners are rewarded for adding new valid blocks. The first node to add a new valid block gains a financial reward in the form of coins; this has caused adding new blocks to become a competitive activity amongst miners. Moreover, although a dishonest node might be tempted to tamper with the current ledger, other nodes would have already added new valid blocks, thus enabling a new longest chain to be created.

Blocks in the blockchain contain two sections: a header section containing the block hash, the timestamp, and the hash of the previous block, and another section containing the transactions collected by the block′s miner. In the case of Bitcoin, the header section contains the nonce used to compute the hash, an additional field called the Merkle Root, and the transactions are arranged in a Merkle tree structure. The Merkle tree is a type of tree wherein the lower base elements are called leaves and the top element is called the Merkle root. Each leaf node is a hash of a particular transaction within the block, and each non-leaf node is a hash of its two children, as shown in Figure 1. Merkle trees are used for simplified payment verification. If the transactions are not organized in a Merkle tree structure, to verify that a particular transaction is in a particular block, a user would have to download all the transactions in the block, output the corresponding hash, and compare it with the block hash. Thus, for a block with N transactions, the user would have to download N transactions. On the other hand, if the transactions are organized in a Merkle tree structure, to check a single transaction, the user would only need the hashes that route his transaction to the Merkle tree. Moreover, for a block with N transactions, the user would only need to download $Log(N)$ elements.

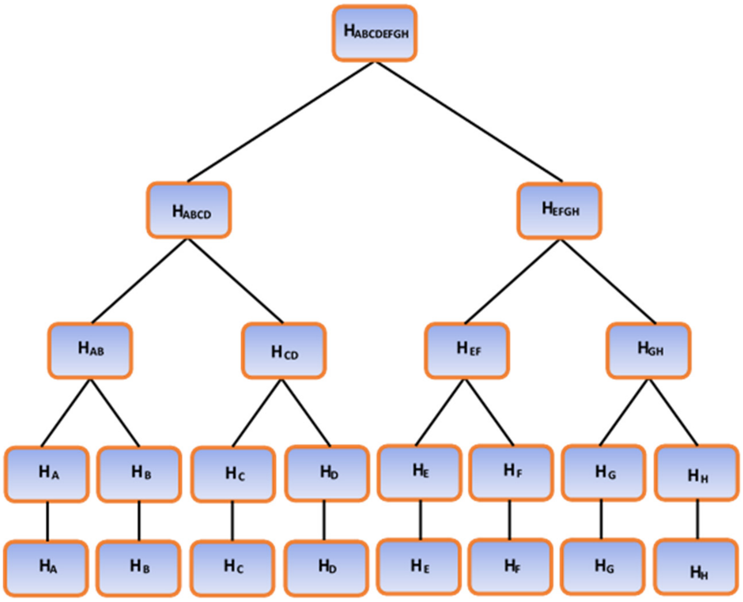


Figure 1. Bitcoin transactions organized in a Merkle tree structure.

The honesty of a miner is judged by the validity of the transactions in the proposed block. The validity of new block transactions is checked against the available transaction history in the public ledger; however, the public ledger must be reliable and tamper-proof. In the blockchain, tampering with the ledger would require a corresponding change to the block hashes. If a node is busy tampering with the ledger and a new block has been added in the meantime, the ledger it tampered with is no longer valid because it is no longer the longest chain. In addition, miners have to prove that proposing the new block was costly for them; therefore, they cannot afford to propose a block with invalid transactions that can be easily detected as they would have wasted resources for nothing.

2.2. Blockchain beyond Cryptocurrency

Blockchain technology has evolved tremendously since Satoshi Nakamoto first introduced the idea of Bitcoin as a decentralized cryptocurrency [8]. Bitcoin has reached a key milestone today, and many businesses now accept Bitcoin payments. Given that Bitcoin has proven its worth for a decade, and due to its enormous popularity, a number of other cryptocurrencies have since emerged, including Litecoin, Namecoin, Peercoin, and Dogecoin [18]. Today, the blockchain concept has surpassed its original purpose. Blockchain has evolved from being the technology behind cryptocurrencies to being the technology behind decentralization [9].

2.2.1. Blockchain for Decentralized Applications and Software

Blockchain has been implemented to enable the functioning of many decentralized applications and software [9]. On regular server–client web apps, all pages and software reside on a server or on the cloud infrastructure. They are then delivered to the client who is requesting them. If a client wants to share some data with another node, data are first uploaded to the server, and then it is delivered to the appropriate addressee on behalf of the sender. Combating this form of centralization is achievable through decentralization with blockchain, wherein information such as software, data, or web pages are shared on the blockchain so that each node possesses a copy of the ledger [19].

Decentralized blockchain apps may be considered as better alternatives to traditional web apps in the following ways:

- It enables a censorship-free network, as depicted in Figure 2; this is because any data that are shared to the blockchain are unchangeable and unerasable. Indeed, for any content to be dismissed, new blocks need to be added to the block that precedes the one containing the undesirable content until that block is ingratiated into a smaller branch that can be bypassed [20].

- It is 'hack-proof' in that a copy of the blockchain ledger is held by all nodes; therefore, a hacker must break into half of the nodes in the network to crack the system [21].

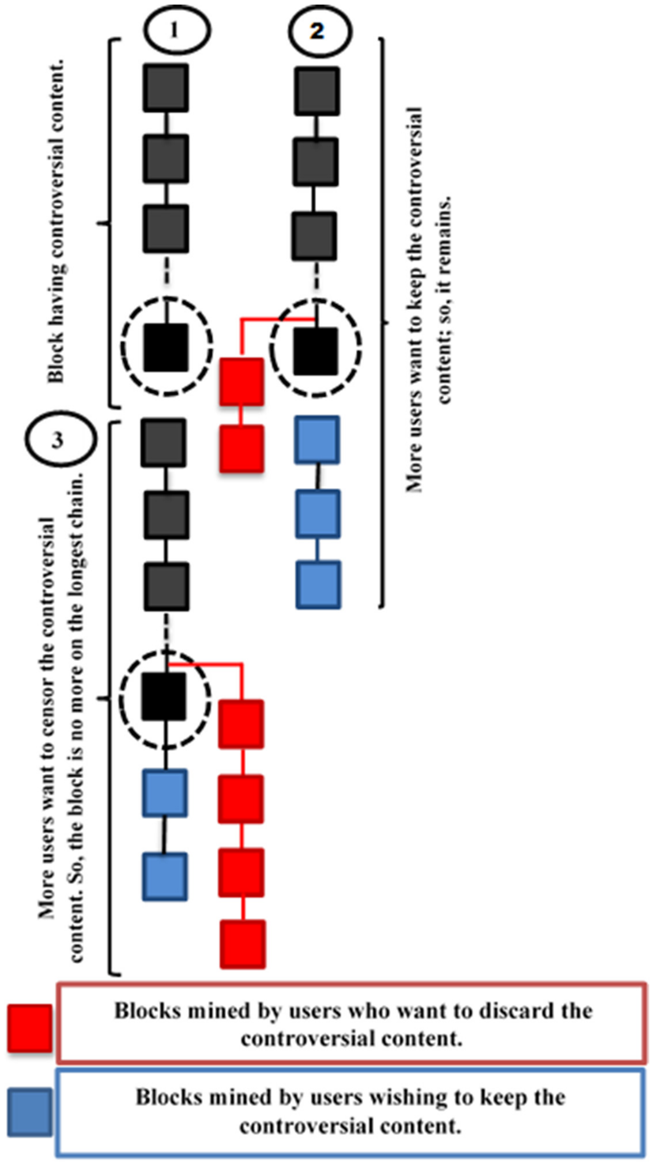

1
2
3
Block having controversial content.
More users want to keep the controversial content; so, it remains.
More users want to censor the controversial content. So, the block is no more on the longest chain.
Blocks mined by users who want to discard the controversial content.
Blocks mined by users wishing to keep the controversial content.

Figure 2. The censor-proof property of blockchain-based apps and software.

2.2.2. Blockchain in the Industrial World

The current popularity of blockchain technology is peaking. It has inspired the broader use of blockchains in different applications in the industrial world [22], not as a mere startup technology, but rather, as an established technology that can be implemented and trusted by pioneer and giant industrial firms alike. One of the potential applications of this nascent technology is industrial supply chain management [23].

Blockchain is now evolving towards compatibility with permissioned platforms for industrial companies and organizations [24]. In contrast to an open blockchain, where nodes are pseudo-anonymized and the network is open, a permissioned blockchain is a closed environment wherein parties belonging to the same industry or organization can interact in a decentralized, distributed manner—rather than a traditional, centralized manner—using a central database server. Using a permissioned blockchain has the following incentive; it enables parties to avoid relying upon a centralized architecture wherein data integrity depends on certain administrators having total access to the database, and thus, having full control of the system and the ability to tamper with it. In contrast, using a decentralized architecture ensures that each node or party within the organization holds a copy of a tamper-proof ledger and the ledger's validity; moreover, correctness is agreed upon and a consensus is reached between all involved parties. In addition, a decentralized architecture is immune to hackers as they would have to gain access to more than half of the nodes in the network to be able to corrupt the data [25]. Conversely, in a traditional, centralized network architecture, there is a single target which comprises the central database. This enables origin information for goods and furniture to be made available, and it improves logistical transparency and supply chain quality. Indeed, with regard to the supply chain management scenario, blockchain is supposed to trace the origin of each type of component or raw material that is used in the manufacturing process of the product [23]; therefore, any inoperative component or material can be traced back to its origin, and thus, it is impossible for the responsible party to deny culpability, ensuring that any products with defective components can be removed from the market. Moreover, why use blockchain in supply chain management over traditional logs in a central database? In actuality, a supply chain may involve different industrial parties from different countries (i.e., each entity would have its own database and its own exclusive access to the chain); therefore, in the unfortunate event where a faulty component is found to exist, it is unlikely that the responsible party would want to share data that might reveal their culpability in the matter. The entities involved would have the option to agree to rent a common database hosted in the cloud, but all parties must have access to it, and they must be able to tamper with the data at will. In addition, all parties must be able to contribute to the cost of the cloud services. In fact, such an incident has already happened; in 2012, Walmart hypermarkets received contaminated pork and mango, resulting in the poisoning of several customers. Walmart tried to recall the contaminated products, but they were unable to trace the respective provenances of the products; therefore, Walmart had to remove all pork and mango stocks from their stores to prevent further sales. This caused significant losses, and they bore full responsibility for the ill-health of the affected customers [26]. Walmart subsequently decided to use IBM′s blockchain solution, which is based on Hyperledger Fabric, in order to keep track of all their purchased products [27]; this is to ensure that defective components and raw materials are traceable.

Beyond supply-chain management, blockchain technology has been implemented in any process that involves different parties working separately; however, the outcomes of their tasks have been interdependent and they required synchronization [27]. Moreover, whenever any team or individual completes any task, it is logged in the blockchain; thus, as a result of certain key characteristics of blockchain technology, the history and related information of all the achieved tasks can be saved forever, and shared between the involved parties in a transparent, decentralized way, therefore making the synchronization of tasks easier to achieve. In sum, when a problem occurs, blockchain technology allows the root of the problem to be traced with ease, including the person who may have been responsible for the difficulties caused; this is significant as some professional faults can have severe consequences [28].

2.3. Smart Contracts

It would be unfair to mention blockchain technology without discussing smart contracts. Smart contracts are computer codes that specify the terms of agreement between two or more parties, and they must be confirmed by network members [29]. These computer programs (protocols), which are published on the blockchain, are used to enable credible transactions that can be conducted without a third party. These transactions are trackable and irreversible [30]. In 1994, smart contracts were first proposed by Nick Szabo, who first coined the term, and later, this became a generic term associated with Bitcoin. Indeed, Bitcoin allows a script to be added to any transaction to set or fulfil a condition [31]. Bitcoin transactions must have a specific format called Unspent Transaction Output (UTXO), and they must contain an input and an output [8]. Figure 3 illustrates Bitcoin transactions in an UTXO format. For example, if Alice issues a transaction in which she sends three Bitcoins to Ali, the input for this transaction is three Bitcoins, and as an output, the transaction notes that these three coins are for Ali. If Ali wants to use these Bitcoins and send them to someone else, the input of the transaction must reference Alice′s transaction output so as to trace the coins to their point of origin. In the same manner, Alice′s transaction to Ali must have its input point reference the transaction output as the origin point from which she received those coins. This ripples back until a coin base transaction is detected; this kind of transaction offers the mining reward to the deserving miner node, and it has no input. This input–output format makes it possible to verify any set condition or specification as a script, thus affecting the circumstances concerning the transfer of the coins. For example, if Alice has set a condition for her transaction, stating that the coins she is transferring can only be spent two months after the transaction issuing date, as Ali would have to reference Alice′s transaction output in the input of his transaction, anyone can thus check the predefined conditions, and accordingly approve or deny that transaction. Given that the coins cannot be issued until two months after the transaction date, anyone can check whether Ali is allowed to spend the coins, and therefore, whether they should accept the transaction or not.

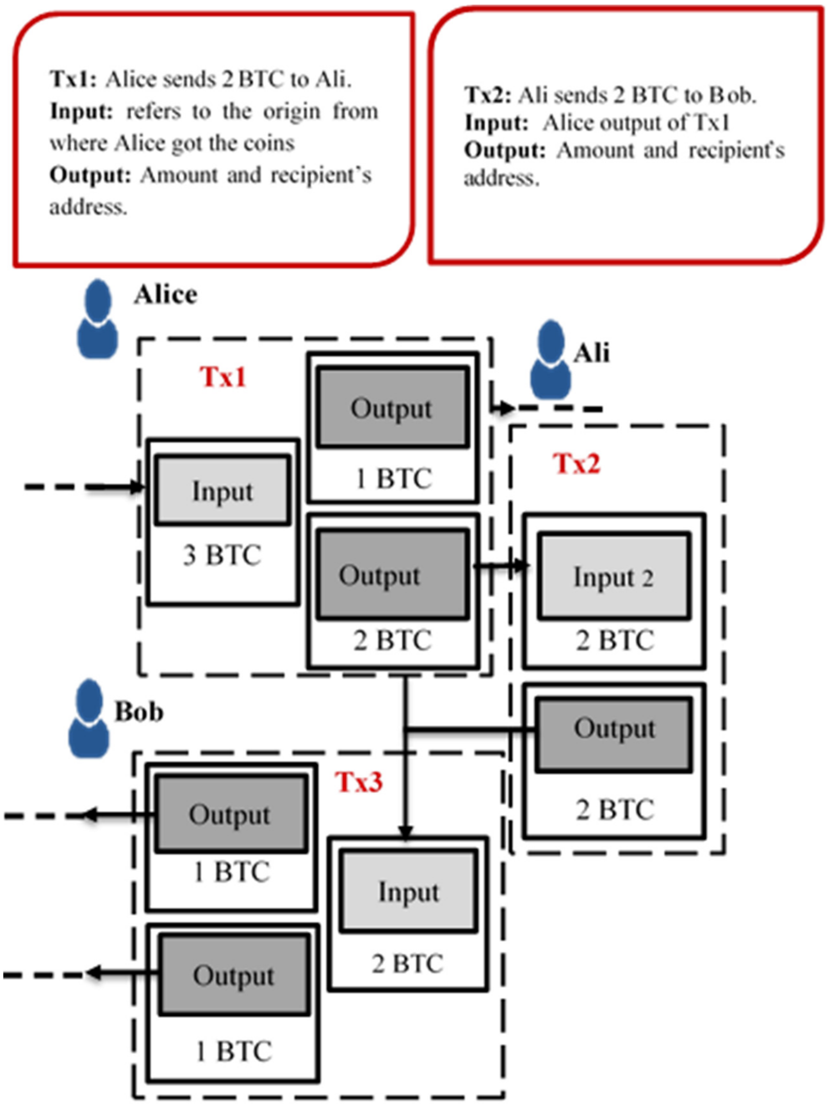

Tx1: Alice sends 2 BTC to Ali.
Input: refers to the origin from where Alice got the coins
Output: Amount and recipient's address.
Tx2: Ali sends 2 BTC to Bob.
Input: Alice output of Tx1
Output: Amount and recipient's address.
Alice
Tx1
Output
1 BTC
Input
3 BTC
Output
2 BTC
Ali
Tx2
Input 2
2 BTC
Output
2 BTC
Bob
Tx3
Output
1 BTC
Input
2 BTC
Output
1 BTC

Figure 3. Transaction input and output formats in Bitcoin.

This idea was subsequently extended from simple script forms to more advanced algorithmic codes and computer protocols that were specifically designed to set more complex conditions in the form of contract codes. These codes govern the automated flow of data distributed through the network nodes, using a permanently locked set of rules, without requiring third-party authentication. Blockchain smart contracts were first introduced by Ethereum and then integrated into other mainstream blockchain platforms [32].

3. Blockchain Consensus Protocols

First, consensus is about using protocols that mainly dictate how the nodes in the network agree with each other, with regard to the ledger that they hold, and the order in which the transactions are listed on the newly mined block. This can be quite a difficult task without the use of consensus algorithms, as they are meant to serve two prominent requirements: safety (a new, validated block that is committed to the blockchain is not changeable anymore) and liveness [33]. This means that an honest user will ultimately be approved if he tries to submit a block that has been proven (in a verifiable manner) to be valid; thus, a good decentralized blockchain relies on a good consensus algorithm to moderate it by enabling the nodes to achieve an agreement, and to maintain the updated blockchain [34]. Furthermore, this section aims to provide a comprehensive overview on how consensus is reached in blockchains by rigorously detailing the consensuses used in blockchain platforms that have the most market capital.

3.1. Consensus in Permissionless Blockchains

Permissionless blockchains such as Bitcoin [8], Ethereum [35], and Monero-based systems do not require pre-established entities to achieve a decentralized consensus [36]. Although Ethereum can be used to enable decentralized applications on a private blockchain network [37], public nodes can join the blockchain without needing permission. Moreover, consensus aims to ensure the validity of transactions and data that are committed to the blockchain in an environment where users do not know or trust each other [38]. The nature of consensus in such an environment is mostly proof-based [39], with nodes competing to earn the privilege of appending a new block to the current chain. In a proof-based consensus, it is compulsory for any node that proposes a new block to include proof that is able to be verified by the other nodes; this is to ensure that they would be penalized if the block contains invalid data or transactions. Moreover, since the costs for proposing the new block are significant, the node that is proposing the block could experience a severe loss if the proposed block is not accepted by most of the other nodes in the network [40].

3.1.1. Proof-Based Consensus Protocols

In this subsection, we briefly sketch out the basics of proof-based consensus algorithms. In addition to the original Proof of Work (PoW), proposed by Nakamoto [8], many other variants have evolved, which we discuss here to determine the advantages and disadvantages of incorporating them into the system.

Proof of Work (PoW)

- PoW is a mathematical puzzle which the network nodes have to solve in order to be able to add a new block to the ledger. The process is called mining and the nodes involved are miners. The puzzle consists of finding a hash for the block to mine, and the hash has a specific number of zeros at its rightmost point. The number of zeros defines a property known as the mining difficulty. Moreover, the block's hash is computed by concatenating all the block fields, thus forming a long string that is hashed using the SHA256 algorithm. Hashing the same string with the SHA256 algorithm always produces the same result. For this reason, PoW uses a field called nonce to prove the effort exerted by a node; this is to ensure its right to append its block to the chain. The miner's role is to try different values of the nonce until it comes up with a value that meets the difficulty conditions. This means that mining the block is costly in terms of resources and time. Indeed, mining a block for Bitcoin costs between 531 USD to a stunning 26,170 USD [41]. Whenever a miner solves the puzzle by guessing the secret value (the nonce field), it adds the block to the chain and sends it to the network. The time when the block was found is called the Timestamp. Its chain is only accepted as the real new ledger if it is the longest one. In other words, whenever a node in the blockchain network receives a new chain, it first checks its validity by verifying that all the hashes are valid, and that each block's hash is computed by using the hash of the previous block. The validity of the transactions in the last block must also be verified, and finally, its length must be greater than the one it currently holds. When these conditions are met, the node will accept the chain as the new valid ledger. The first node which solves the puzzle and adds the new block to the chain will broadcast the longest new valid chain, and they would receive a reward for mining, thus making the process of mining a difficult, competitive process and a lucrative investment at the same time [8]. After appending the new block to the blockchain, the cycle restarts; therefore, the size of the blockchain can be increased indefinitely. PoW makes it more financially advantageous for the miners to show honest behavior by keeping the system in good order rather than trying to tamper with the ledger or include invalid transactions [42]. The most famous cryptocurrencies that use the PoW consensus algorithm are Bitcoin and Ethereum [43]. The PoW algorithm has always required proof as it is the oldest existing algorithm in blockchain technology. It also allowed Bitcoin to prevail as the most secure and reliable cryptocurrency. Indeed, since 2009, Bitcoin has been ranked as number one in terms of market capitalization [44]. The PoW algorithm makes blockchain technology tamper-proof, strong, and secure, and it avoids double spending; however, it has limited throughput capabilities. Indeed, as a result of the algorithm, it is estimated that the blockchain can only facilitate 27 transactions/s. This is due to the fact that limitations are placed upon the authorized size of the blocks and the time taken to mine the blocks [45]. In addition, there are significant costs associated with blockchain mining. The mining process consumes a great deal of energy due to the large quantities of energy that are required to supply the extremely powerful hardware that mines the blockchain [46]. Figure 4 reveals that annually, Bitcoin uses more electricity than the whole of Argentina. If Bitcoin was a country, it would be in the top 30 energy users worldwide [47].

- As mining a block is linked to the miner's hardware hashing power, this may lead to the monopolization of mining by nodes with enormous hashing power. In fact, the more processing power a miner has, the more likely they are to hold a monopoly over the network. If a user possesses 51 percent of the hashing power, this would ruin the blockchain because nobody would be able to compete with them; this means that the ledger's integrity would be completely at their mercy. They would be able to double spend at will, forking on the block which precedes it, and they would be able to accept or boycott the transaction in accordance with their desires. In other words, blockchain integrity would be destroyed if it were to undergo such an attack [48]. Figure 5 demonstrates how fraudulent parties may win the mining race if they have sufficient hashing power. Forking is an important concept in blockchain technology that must be clarified. A fork takes place if blocks are mined by two miners at the same time. If such an instance occurs, the miners will have to choose to a fork to mine their new blocks on; then, the longest chain prevails.

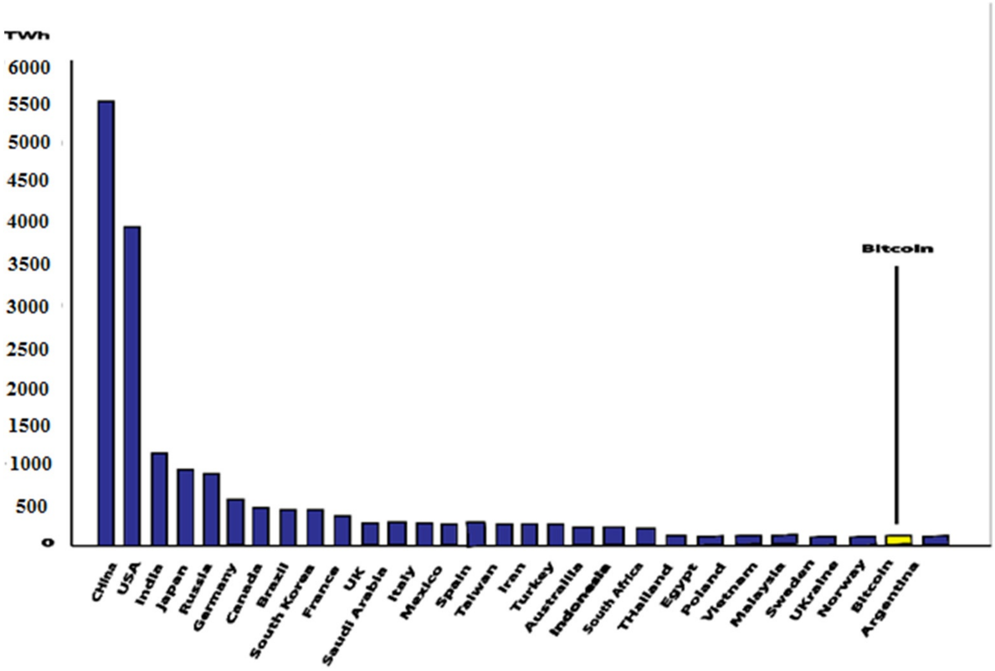


Figure 4. National energy use in TW/h. University of Cambridge Bitcoin Electricity Consumption Index [47].

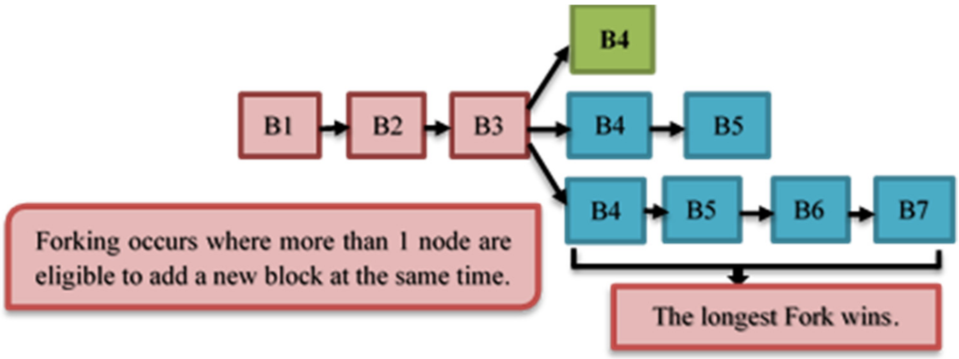

B1
B2
B3
B4
B4
B5
B4
B5
B6
B7
Forking occurs where more than 1 node are eligible to add a new block at the same time.
The longest Fork wins.

Figure 5. The most extended version of the chain–win principle in blockchain technology.

However, forks may be intentional, and users may agree not to mine a specific block, instead, they may choose to mine the previous block. If they succeed mining enough blocks and they win the mining race, all of the blocks after the fork would be undone [49]. Forks are illustrated in Figure 5.

Moreover, there are several other problems that may be caused by Proof of Work if the hashing power is not evenly split between miners. They are as follows:

- Sybil attack: this attack consists of faulty or malicious entities that may present as multiple identities in order to gain control of the system. Since the nodes in the PoW-based network are identified using only their public key, it is possible for a user to generate as many public keys as they wish, thus enabling them to potentially create an infinite number of accounts [43].
- Selfish mining: if a party (miner or miner-pool) is sure that they can mine newly created blocks at least two times faster than any other node in the network, they may be partaking in what is called selfish mining. Selfish mining occurs when strong nodes can withhold blocks, which they create, and selectively postpone their broadcast. As a result, a fork is deliberately generated in a chain that would have otherwise been the longest chain, and consequently, this might force the honest network to discard some

of its blocks [44,45]. Thus, selfish mining is one such downside of the "longest chain wins" principle, as illustrated in Figure 6.

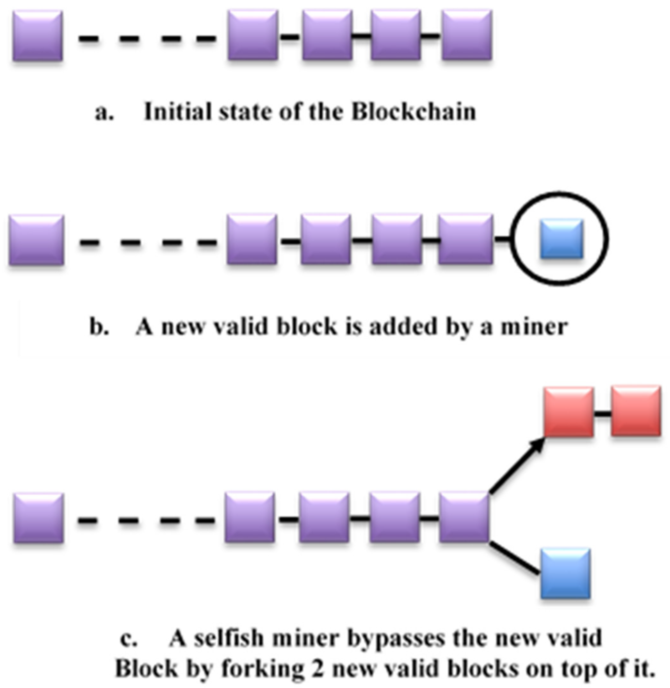

a. Initial state of the Blockchain
b. A new valid block is added by a miner
c. A selfish miner bypasses the new valid Block by forking 2 new valid blocks on top of it.

Figure 6. Selfish mining.

Proof of Stake (PoS) and Its Variants

The Proof of Stake consensus requires nodes to apply to be validators by locking a certain number of coins to a specific smart contract account. Then, a single validator is selected using pseudorandom selection, or it is based on the age of the set coins, otherwise, the validator is deselected. This is illustrated in Figure 7. Most PoS-based blockchain platforms use Verifiable Random Functions (VRF) to select the validator from a group of eligible candidates.

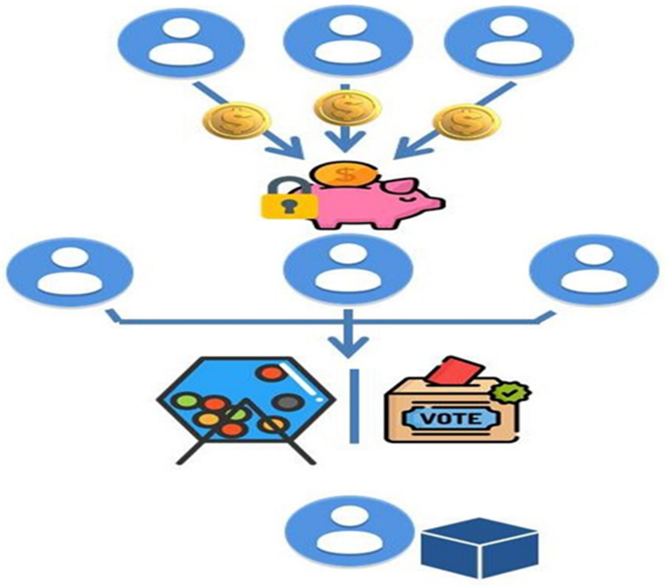


Figure 7. Process of validating a new block when using the PoS consensus.

Ethereum looked to Proof of Stake as a solution to solve the aforementioned problems with PoW. In contrast to PoW, the Proof of Stake algorithm [50] favors miners who have the most hashing power, and thus, it avoids the issue that causes a network to be monopolized by those who have the most powerful hardware. Consequently, it means that the mining process is far less energy-consuming during the transaction validation and consensus processes; therefore, the mechanisms associated with PoS are much greener than PoW. With PoS, new, specific vocabulary is used. For example, adding a block to the chain is called validating a block, and the nodes that are involved in the block are called validators.

a. Proof of Stake

A form of PoS algorithm can be seen on the Nxtcoin platform [51]. Sunny King and Scott Nadal first published their research on PoS in 2012. [52]. In comparison to other algorithms, block validation is completed by a set of validators who have staked a sufficient amount of cryptocurrency in the network. The greater the amount of cryptocurrency that a validator has staked, the greater their chances of validating a new block; however, the validator who has staked the most cryptocurrency is not automatically chosen to validate a new block. If this were the case, mining a block would be monopolized by those validators, and as a result, the blockchain would neither be decentralized nor reliable since the consensus would be solely decided by a fixed number of users who have the greatest amount of cryptocurrency staked in the network; however, in actuality, an element of randomness is added to the process of choosing the next block validator [53].

There are many variants of the PoS algorithm [50]. In general, there is a certain threshold of coins that each user must stake in the network in order to become a candidate validator. Staking one's coins in the network is undertaken by a defined type of transaction that transfers the coins to a defined account wherein the stakes are locked. Once the owners have validated the block they mined in the network, they can take back the amount that they staked; however, they can no longer participate in the validation process.

b. Chain-based PoS

In essence, a chain-based-PoS algorithm assigns the right to create new blocks to stakeholders in a pseudo-random manner [54]. For instance, the chain-based-PoS algorithm might have to contend with each validator proposing a new valid block; the block would then be chosen using a verification algorithm which considers the size of the generated hash and the amount that has been staked by the validator. Since hash computation is unpredictable, it adds an element of randomness to the validator selection process. This is used in NXT cryptocurrency [55]. A variant of this approach is the coin age approach, which combines how long the stakes have been held for, as well as their amount. This is the PoS system used by Peercoin [55].

c. Casper PoS

Casper Vlad Zamfir is often referred to as the "Front of Casper", the new PoS algorithm proposed for Ethereum [56]. Previously, Ethereum has officially stated that it plans to imminently move from a PoW algorithm to a Casper PoS algorithm [57]; however, only one hard fork has been released that implements PoS, called GHOST, which, thus far, has been subject to three fraudulent attacks [58]. These three attacks have been on Proof of Stake Ethereum and full consensus conversion has not yet taken place. Currently, there is talk of merging the two algorithms, rather than fully migrating to the Casper PoS algorithm; this has been announced for the second quarter of 2022. Moreover, what makes Casper different from other PoS protocols?

The central motivation behind Casper is to address the "nothing at stake" problem in the PoS consensus algorithm. Moreover, Casper uses an accountability function protocol that penalizes any malicious behavior or attempted fraud [52,59]. With PoW, mining a new block on both chains forces the node to split his hashing power by two; this, risks losing the mining race, and thus wasting energy and hashing power for nothing. In contrast, with PoS and PoI algorithms, each time an eligible validator faces a fork, a new block can simply be added to both chains; consequently, the network ends up with two chains of the same length in which coins can be double spent [60].

What is proposed with the Casper algorithm, is that to even qualify as a validator, one needs to lock a significant proportion of one's fund to use as a stake; even then, one can only receive a reward that is proportional to the staked bet. If a validator acts maliciously, in a "nothing at stake manner", they will immediately be punished, and they will find that their invested stake is slashed and removed. This is depicted in Figure 8.

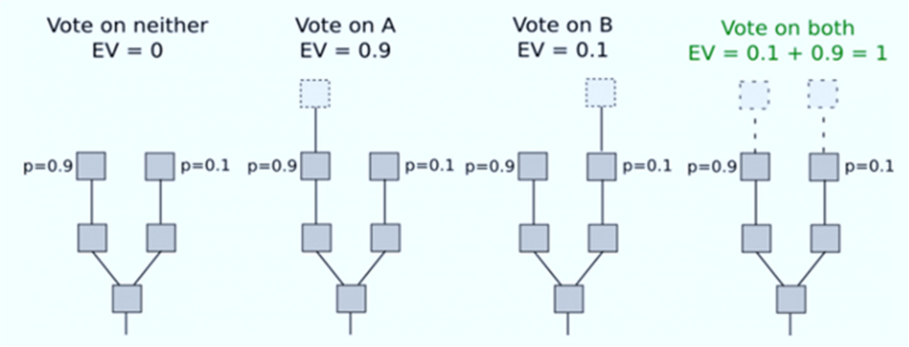


Figure 8. Process of validating a new block using Casper PoS.

d. Delegated PoS (DPoS)

Delegated Proof of Stake is another variant of PoS [61], wherein stake holders vote for a delegate to be responsible for adding the new block, instead of voting for or adding it themselves. The stakeholders possess a number of votes that is proportional to the amount they staked in the network. Compared with PoS, DPoS can significantly reduce latency, which is very attractive in IoT networks; however, it may cause the system to become centralized.

e. Proof of Importance (PoI)

With Proof of Importance algorithms, which are used by NEM [62], each account is assigned an importance score that represents its acquired importance to NEM. The nodes which have the highest importance score, have more opportunities to reap a block. In the case of NEM, the importance score is computed depending on the amount of vested money, which is equivalent to the amount of money that has been staked. NEM requires a balance of at least 10,000 vested XEM (XEM is the cryptocurrency enabled by NEM) to be able to harvest, which is supposed to limit the number of accounts that are expected to fall victim to Sybil attacks. In addition, such attacks are unlikely in PoI, thus causing transfers to decay over time, as only 10% of this amount can actually be vested by the network in 24 h. Accordingly, it would require ten days to vest 10,000 XEM, whereas several weeks are needed to ensure that inflows are fully vested into an account. Despite implementing these countermeasures against Sybil attacks, PoI still suffers from the "nothing at a stake" problem [63]. In PoW, mining a new block on both chains forces the node to split their hashing power by two, in spite of the risk of losing the mining race, and in addition to wasting energy and hashing power for nothing. Conversely, any time a validator is confronted with a fork, they can simply add their new block to both chains. Consequently, the network will end up with two chains of the same length, wherein coins may be double spent [60]. Figure 9 illustrates the importance score in PoI algorithms. Table 1 provides the major advantages and disadvantages of Proof of Stake-based algorithms, whereas Table 2 shows the top ten cryptocurrencies of 2022 in terms of market capital, in addition to the corresponding consensus protocols used for those cryptocurrencies.

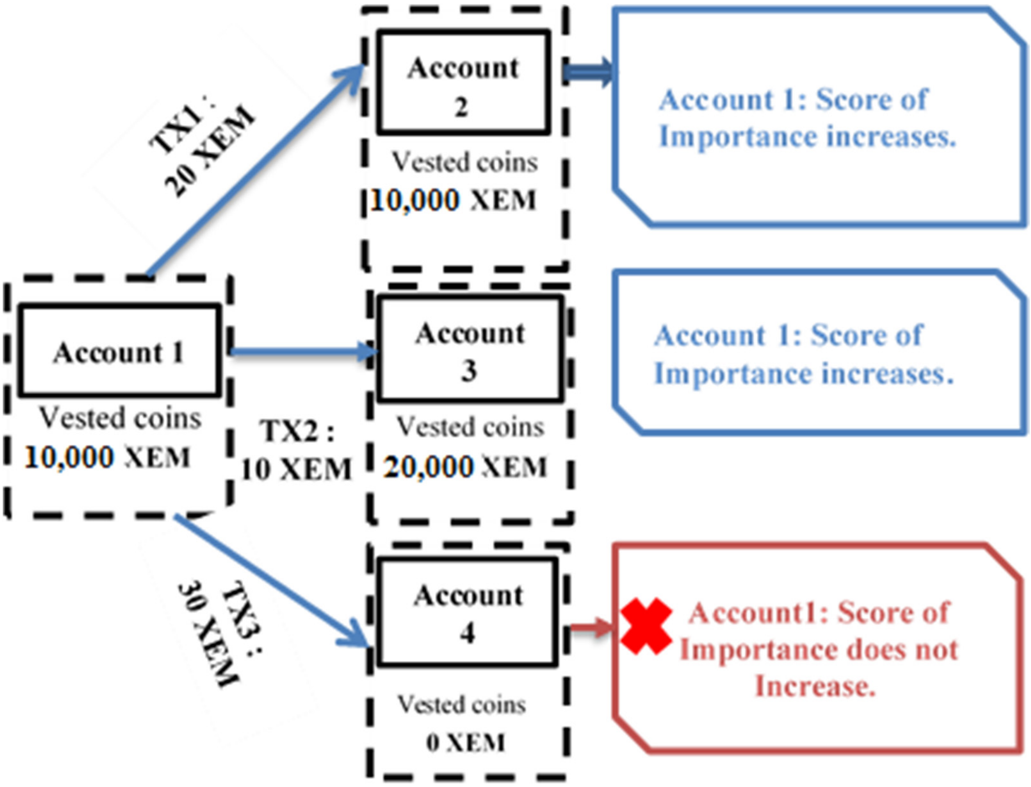


Figure 9. Increasing Importance Score in NEM PoI algorithms which contribute to the network.

Table 1. Proof of Stake-based algorithms.

Strengths	Limitations
- A much greener way to add new blocks. - Risk of centralization reduced as anyone who staked a sufficient amount of cryptocurrency in the network is able to become a potential block validator. - Faster to add a block with a PoS algorithm compared with a PoW algorithm. - More scalable in terms of the number of transactions processed per second. - More resistant to Sybil attacks.	- The "nothing at a stake" problem. - Not suitable for non-financial decentralized applications.

Table 2. Top ten cryptocurrencies according to market capital and the corresponding consensus protocols used, 2022.

Cryptocurrency	Market Capital EUR	Consensus Protocol Used
Bitcoin	564,381,542,293 EUR	PoW
Ethereum	111,207,696,199 EUR	PoS
Tether	18,623,408,862 EUR	PoW and PoS
XRP	11,046,513,838 EUR	Ripple
Litecoin	8,983,001,102 EUR	PoW
Cardano	8,316,233,583 EUR	PoS
Polkadot	7,232,689,163 EUR	PoW
Bitcoin Cash	6,769,504,611 EUR	PoW
Stellar	6,263,250,767 EUR	Stellar
Tezos	1,619,850,592 EUR	PoS

The Mining Power Consumption/Nothing at a Stake Trade-Off

As previously mentioned, the PoW algorithm comes with the huge burden of mining power consumption. PoS has been proposed as a greener consensus protocol; however, reducing mining power consumption comes with another compromise. In actuality, forks in blockchains are supposed to be as short as possible, because multiple valid chains of the same length would lead to nodes with different balances, and thus, double spending would be possible. Unintended forks have not been documented; however, the length of unintended forks can be deduced from the number of orphaned blocks in the blockchain. The longer a fork is, the more likely it is for blocks to be orphaned, as a single branch would ultimately prevail. Table 3 and the graph in Figure 10 show the number of orphaned blocks versus the mining power consumption in different blockchain-based cryptocurrencies. As illustrated in Table 3, the average power consumption for block mining is a function of:

- The wattage of the mining device used, (P).
- The number of devices used to mine a block, (ND).
- The average block mining time, (T_{Av}).

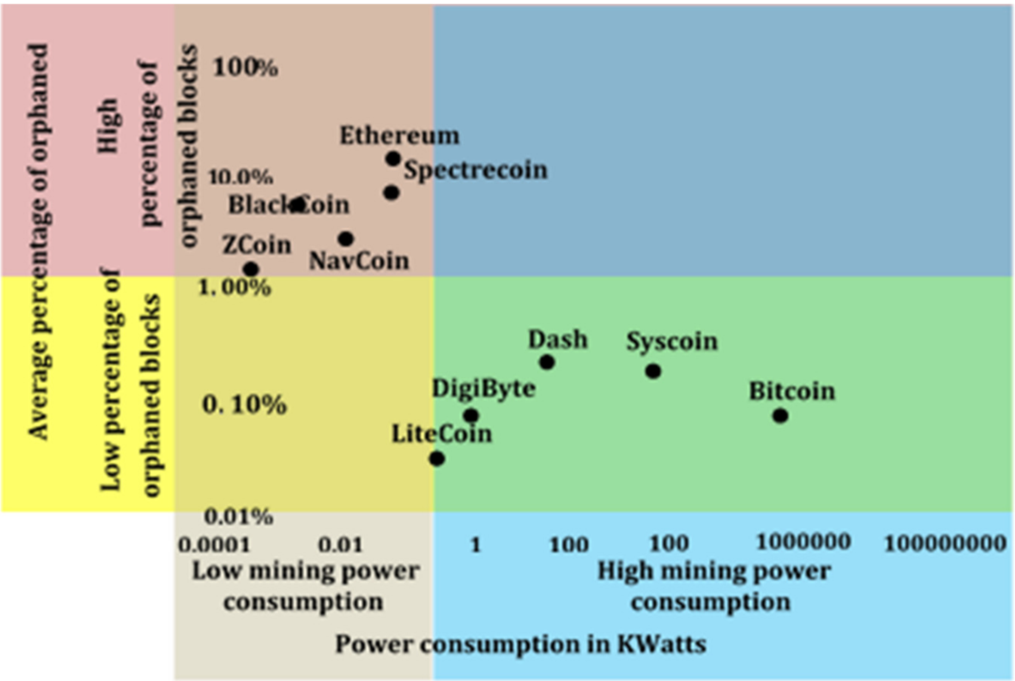


Figure 10. Percentage of orphaned blocks per day versus power consumed during block mining in blockchain-based cryptocurrencies, colors are used to demonstrate that blockchain located in high percentage of orphaned block zone intersects with the low mining power consumption zone and vice versa.

Furthermore, assuming that all networks are using an Antminer mining device, 14 Th/s, of 1320 watts [64], the number of devices used to mine a block in the desired time is given in (1) as:

$$ND = HR(Th/s))/(14\ (Th/s) \quad (1)$$

where HR is the network hash rate, and 14 Th/s is the hashing power of a single mining device, used.

Moreover, the power consumption for block mining (W_c) is computed for each blockchain network according to the expression in (2):

$$W_c(KWh) = P(KW) \times T_{Av}(h) \times ND \quad (2)$$

Documentation regarding the network hashing rate was not available for the Stratis, Navcoin, and Spectrecoin cryptocurrencies. References [65–67] were thus used to obtain the corresponding current mining difficulties (D). Moreover, we were then able to deduce the network hashing rate according to (3) [68]:

$$HR = D \times 2^{32} \; \times T_{Av}(s) \tag{3}$$

Table 3. Hashing rate, mining time, and orphan block rate for different blockchain platforms.

Blockchain-Based Cryptocurrency	Hashing Rate (Jan 2020, T Hash/s)	Daily Average of Orphaned Blocks [69]	Daily Average of Mined Blocks [69]	Daily Average Percentage of Orphaned Blocks	One Block's Average Mining Time (s) [69]	Average Mining Power Consumption per Block (Antminer 14 T hash/s) [70]
Bitcoin	146 M [71]	0.1	158	0.06	584	2,233,104.76
Ethereum	348 [70]	717	6371	10	13	0.12
DigiByte	84 K [72]	6	5770	0.08	15	3.3
Litecoin	280 [73]	149	0.2	0.03	578	1.09
Dash	5 K [74]	1	549	0.18	157	72.06
Syscoin	21 M [75]	2	1372	0.14	62	34100
Zchah	0.06 [76]	3	286	1	302	0.5×10^{-3}
Blackcoin	21 [77]	50	1296	3.85	66	36.3×10^{-3}
Stratis	4 [65]	70	1264	5.24	68	7.12×10^{-3}
Navcoin	78, 67 [66]	80	2828	2.75	30	61.81×10^{-3}
Spectre-coin	80 [67]	60	1361	4.22	63	0.132

Figure 10 reveals that cryptocurrencies with low mining power consumption have a high percentage of orphaned blocks and vice versa.

Hybrid PoS and PoW Algorithms

As previously noted, PoS and PoW algorithms are compromised. On the one hand, the PoW algorithm makes it difficult to mine a block. This improves the reliability and robustness of the chain; however, its inherent flaws concerning power wastage and high computational power requirements could potentially risk monopolization, and it means that the system is not ecofriendly. On the other hand, the PoS algorithm is much greener in terms of energy consumption, and it has the potential to prevent monopolies forming in the network; however, being able to easily add blocks may generate the "nothing at stake" problem, which may lead to coins being double spent. Consequently, it may be argued that using a mixture of PoW and PoS algorithms could produce an optimized solution for reaching a consensus in blockchains. In such a scenario, the number of coins that the node has staked in the network, or its importance score, would determine the difficulty of the PoW puzzle that needs to be solved; the higher one's importance score, the easier the puzzle. This concept was introduced for the first time in a paper by King and Nadal [52]; however, even though it seems promising, the idea did not receive much interest, and few blockchain platforms have adopted it.

Other Proof-Based Consensus Algorithms

In addition to PoW and PoS algorithms, there are other other kinds of proof-based consensus algorithms. Several of these are detailed below.

a. Proof of Burn

Instead of staking coins that may be taken back by stakeholders (in cases where the validated block is correct, and the miners do not wish to be part of the validation process anymore), validators are required to burn coins by sending them to a blocked account; this means that they are forever irretrievable. The more coins burned by a node, the higher their chances of mining a new block [78]. The Proof of Burn (PoB) is used in the cryptocurrency called Slimcoin.

b. Proof of Elapsed Time

With the Proof of Elapsed Time (PoET) algorithm [79], nodes have to identify themselves before joining the network. Blocks are not accepted if they have been mined prior to the minimum defined threshold time.

PoET has a low power consumption, but it exhibits high latency, making it particularly noteworthy for users with limited resources, but not time constrained IoT applications [80].

c. Proof of Capacity/Space

Similarly, to PoW, Proof of Capacity/Space algorithms need miners to solve a puzzle; however, instead of hashing power, it relies on the hard disk capacity of the miners. Thus, it consumes significantly less energy, but creating new blocks requires high latency. This method is not especially appropriate for devices with limited storage capacities [81].

Table 4 compares different proof-based consensus algorithms according to a set of essential blockchain property.

Table 4. Comparison between blockchain consensus protocols for the compilation of a set of essential blockchain properties.

Consensus Protocol	Blockchain	Eco Friendly	Tolerated Malicious Power	Data Model	Development Language	Execution	Examples
PoW	Public	No	Computing power, 25%	Transaction-based, Account-based	Golang, C++, Solidity, Serpent, LLL	Native, EVM	Bitcoin, Litecoin, ZCash, Ethereum
PoS	Public	No	Stake, 50%	Account-based	Michael son	Native	Peercoin, Tezos, Tendermint
PoI	Public, Private	Yes	Unknown	Transaction-based, Account-based	Java	NEM	XEM
PoA	Public	Partial	Online stake, 50%	Account-based	Solidity, Java, Python	EVM, Dockers	Parity
PoB	Public	No	Computing power, 25%	Transaction-based, Account-based	Golang, C++, Solidity, Serpent, LLL	Native, EVM	Slimcoin
PoET	Public	Yes	Unknown	Key-Value	Python	Native	Sawtooth Lake
PoC	Public	No	Unknown	Key-Value,	Unknown	Unknown	File Share
DPoS	Public	Partial	Validators, <51%	Transaction-based, Account-based	No scripting	Native	Bitshares

An objective way to assess which of the different PoS consensus variants is the best, is to observe which PoS variant has the highest market capital. This is because the number of users that a blockchain platform attracts is reflective of the robustness and security of the consensus protocol it uses. Cardano has the highest market capital [82] among the blockchain platforms that use PoS algorithms. Cardano uses VRF to select the block validator among eligible stake holders.

3.2. Consensus in Permissioned Blockchains

In permissioned blockchains, all the nodes in the network theoretically know each other, though the number of nodes that interact is limited; thus, the consensus algorithms used for such platforms are much laxer, and they entail fewer burdens compared with the algorithms used for permissionless blockchains [68].

3.2.1. Voting-Based Consensus Protocols

Unlike the proof-based consensus algorithm, wherein nodes can leave and rejoin the network at will, with voting-based consensus protocols, the nodes within the verifying network should be identified and made adjustable. Voting-based consensus protocols are also used in some blockchain open networks using the Federated Byzantine Agreement (FBA); this is used for Ripple and Stellar cryptocurrencies [83]. Voting-based consensus protocols can be categorized into two types: (i) Crash fault tolerance-based consensus protocols and (ii) Byzantine fault tolerance-based consensus protocols.

Crash or Network Fault-Tolerant Consensus

Crash and network fault-tolerant consensus protocols enable consensus to be reached in a distributed, decentralized network, but they do not support Byzantine fault-tolerance protocols. This means that nodes can reach a common consensus by using this kind of algorithm in such networks, despite the fact that some other nodes or network connections are technically faulty; however, consensus cannot be achieved when nodes behave maliciously.

a. Paxos

As a consensus protocol that is used in distributed systems [84], Paxos is an analogy for the type of parliamentary voting that took place on the ancient Greek island of Paxos, where there were three main agents: there were proposers of new laws, acceptors who voted for the laws, and finally learners, who were in charge of writing and recording the new laws. Each time a node wants to add a new block in the blockchain, so that all nodes update their ledgers, the chosen validators play the role of the acceptors, and they vote to either accept or reject the new block. If most acceptors vote to accept the block, all the other nodes (learners) update their ledgers and add the proposed block. Fault tolerance is achieved because several acceptors vote for the proposal; therefore, even if one is faulty, consensus can still be reached unless more than $N/2 - 1$ acceptors in the network are faulty (N is the total number of acceptors in the network). However, Paxos does not tolerate Byzantine faults, and it can only operate if all nodes mutually trust one another. In the case of smart contract execution, this is implemented in the form of a state machine, and it only needs one node machine to be executed. Then, if accepted, the state machine is replicated in all the nodes, and thus, they can add the new contract execution state to their ledgers.

b. Raft

This algorithm has been proposed as an alternative to Paxos [84]. The name 'Raft' refers to the structure that was used to escape Paxos. Raft offers solutions to potential issues in Paxos. These can be summarized as follows:

If more than one node suggests a proposal simultaneously, a single proposal must be agreed (voted) upon by the acceptors, and the others must be denied; however, they may be proposed again in upcoming voting rounds. When considering a Raft cluster, it is notable that verifying nodes can be one of three different types of server; therefore, at any given time, each server is either a leader, follower, or candidate. At the beginning of each term, all nodes in the network start as followers. In order to propose a new log for replication in the network, a node will seek to become a candidate so that they may become a leader, prompting them to ask for votes; each node can only vote once per term. Moreover, a candidate wins an election if they receive the most votes from the servers in the cluster; this must take place in one term. Then, the leader′s proposal is replicated by all nodes in the network. Additionally, whenever a node is elected as leader, a framework of time for their term duration is set out. In cases where a term elapses with no leader, the node is considered to be faulty; therefore, a new term begins when a new leader is elected. Raft′s performance is almost similar to that of other consensus algorithms, such as Paxos. The most important performance indicator concerns the minimal number of messages Raft uses when an established leader is replicating new log entries. Moreover, it is open to possible further improvements [85].

Byzantine Fault-Tolerant Consensus

There are more elaborate consensus protocols used when reaching this second type of voting-based consensus; namely, the Byzantine fault-tolerant protocols, which allows consensus to be accomplished on the distributed network, even though some nodes behave maliciously or are technically faulty [86].

a. Practical Byzantine Fault Tolerance (PBFT)

PBFT was proposed by Castro and Liskov [87]. It is the most commonly used set of consensus protocols in permissioned blockchains [88]. It has been implemented by the Tendermint blockchain, IBM′s open chain, Iris DB, and Hyperledger.

Reaching consensus with PBFT involves three types of agents (client nodes, commanders, and lieutenants), and there are three phases. Firstly, in the pre-prepare phase, the client sends a request, proposing a new block to the commander (which is a defined node that has been designated as commander in the network), so it can be verified by lieutenant nodes (and assigned nodes). If this request is valid, a prepare notification is then sent to the commander, who is supposed to receive f prepare notifications out of the $2 * f + 1$ possibilities in the network. Finally, during the commit phase, the request is validated as a new log that can be added to the ledger. In the example illustrated in Figure 11, Sami is the commander and there are three lieutenant nodes. Accordingly, the network can tolerate only one faulty or malicious node; in the example given, this is Nadir. The commander receives the client request and sends a pre-prepare request to the lieutenants to check its validity. He then receives two out of three prepare notifications, and he considers the last node to be faulty or malicious. Since two thirds of the lieutenants are honest nodes, and as they agreed on the validity of the client request, the log is then validated and propagated in the network, and can therefore be added to the ledger.

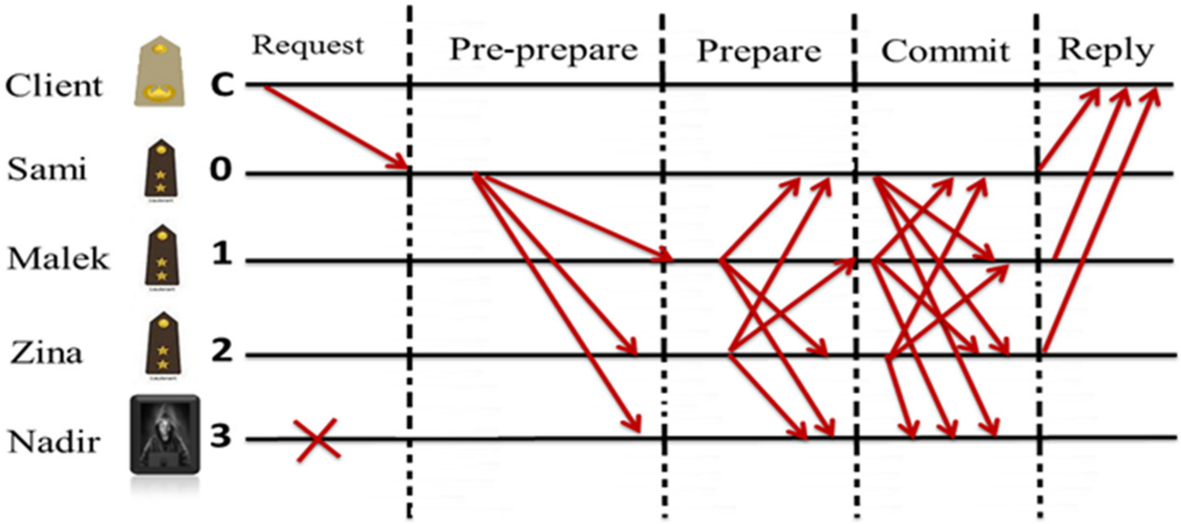


Figure 11. Reaching a PBFT consensus with three lieutenant nodes.

b. Stellar consensus protocol (SCP)

A Stellar consensus algorithm is the permissionless blockchain version of the Practical Byzantine Fault Tolerant consensus (PBFT). It is a more decentralized alternative to PBFT as it is open to the public. It uses what is known as the FBA.

PBFT was intended to be a consensus for permissioned blockchains, where a set of validators is predefined in the network, usually by the company or organization that uses the blockchain platform. With Stellar, validators are not predefined, but rather, they are selected in a more decentralized fashion.

This algorithm owes its strength to quorum slices, which consist of a list of trusted validators that are proposed by each node in the network.

Once all nodes in the network are within a quorum slice, validators become the nodes inside the most overlapping slices; in other words, validators are voted out by an entire group of network nodes to avoid being predefined by a central authority [83], as shown in Figure 12.

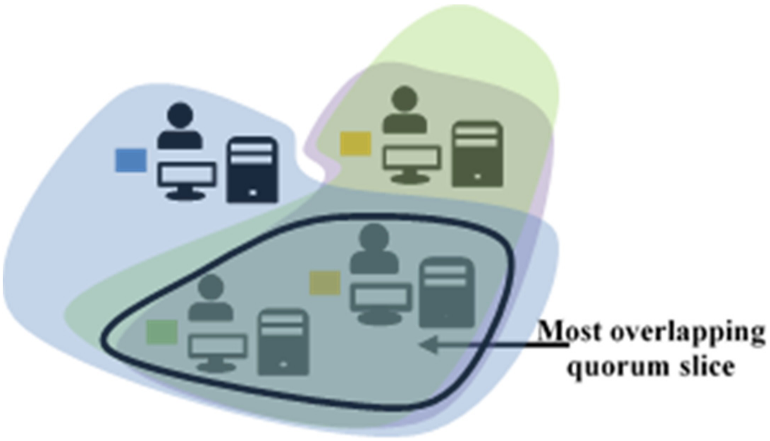


Figure 12. Validators voting in the Stellar consensus protocol.

4. Blockchain Consensus Performance Metrics

Blockchain technology limits human error and provides more cost- and time-effective management by removing intermediaries. It is important to note that the performance metrics in blockchains are different from those in traditional networks. This is because the main concern in blockchain networks revolves around the reliability of the shared ledger. Throughput is a key performance indicator in conventional networks, although, in blockchain networks, many consensus algorithms purposely limit the throughput of the blocks committed to the blockchain. The main performance metrics, which all blockchain consensuses aim to achieve, are the maintenance of the integrity of the network and the annihilation of malicious attacks, such as double spending and tampering with the ledger. Moreover, the cost at which this integrity is attained is also a targeted performance endpoint. Nevertheless, researchers still aim to achieve it, striving to accrue the fewest possible burdens without compromising the decentralized aspect of the blockchain. Another important feature to be considered, regarding performance, concerns mining fairness in the network. Blockchain is a technology for decentralization; however, as more new blocks are mined and monopolized by a certain number of nodes, platforms become less decentralized. In this instance, the worst-case scenario might be the 51% attack that could ruin the integrity of the blockchain.

To capture the fairness of the network [89], a mining fairness index (FI) has been created to demonstrate the extent to which all nodes take part in the mining process. Intuitively, we expect that in a completely fair network of n nodes, each node must mine $(100/n)$% of the total blocks. Given that user number i has allocated a portion of resources, referred to as x_i, fairness can be computed using the Jains Fairness Index formula in (4), which is applicable to any resource sharing or allocation problem [90].

$$FI(x_1,\ x_2,\ \ldots,\ x_n) = \left[\sum_{i=1}^{n} x_i\right]^2 / \sum_{i=1}^{n} x_i^2 \tag{4}$$

The Fairness Index lies between 0 and 1. Consequently, the more (FI) tends toward zero, the fairer the mining process in the network, whereas FI =1 corresponds to a totally fair system where all contenders have been allocated resources equally. In the case of mining, the contenders are miners competing to add new blocks. Moreover, each miner is allocated resources based on the percentage of the block that is mined by the contender in question.

Motivated by this important metric, we can now look at concrete examples of Bitcoin and Ethereum blockchains, and assess mining fairness accordingly.

As Bitcoin and Ethereum nodes are pseudo-anonymized, it is thus difficult to assess the percentage of blocks mined by a single node, as a single node may be using several different public addresses; however, due to the enormous hashing power required to mine a block, it is doubtful that a single private individual possesses such hashing power. For that reason, there exist what are known as mining pools. These comprise of a set of miners that combine and join their hashing power to mine new blocks. The mining reward is then divided proportionally, in accordance with each node's contribution to the overall hashing power. With regard to Bitcoin and Ethereum, most mining pools are identifiable by their mining contribution; thus, it is possible to compute mining fairness with respect to mining pools and their mining contribution, as shown in Table 5 [91].

In accordance with the fairness expression stated in Equation (5), the mining Fairness Index of Bitcoin, BTC_FI is

$$BTC\_FI = 0.36 \tag{5}$$

This indicates that Bitcoin is unfair to 64% of its miners. Table 6 also illustrates Ethereum mining pool contributions (from 1 February to 4 February 2022) [92].

Table 5. Bitcoin Mining Pool Contributions (from 1 February to 4 February 2022).

Mining Pool	Bitcoin Mining Pool Contributions (1 February 2022–4 February 2022)
Unknown	37.39%
F2Pool	13.77%
Poolin	12.16%
Antpool	9.66%
Huobi.Pool	8.58%
ViaBTC	6.08%
SluchPool	5.37%
BTC.TOP	2.33%
EMCD pool	1.07%
Novablock	1.07%
OKExPool	0.89%
WAYI.CN	0.89%
BTC.com	0.54%
SBI Crypto	0.18%

Table 6. Ethereum Mining Pool contributions (from 1 February to 4 February 2022).

Mining Pool	Ethereum Mining Pool Contributions (1 February 2022–4 February 2022)
Ethermine	27%
F2pool	18%
SparkPool	16%
Nanopool	12%
MiningPoolHub	10%
Others	17%

For simplicity's sake, in Table 6, we grouped "others" together in a single mining pool. In this instance, the Fairness coefficient $Ether\_FI$ for Ethereum would be:

$$Ether\_FI = 0.90 \tag{6}$$

This indicates that Ethereum unfair to 10% of its miners. In the same vein, the mining coefficient can be computed for all other blockchain platforms.

5. Drawbacks of the Current Trend in Consensus Algorithm Development

In reality, blockchain technology is considered to be a fast-paced topic, and numerous consensus protocols have been proposed in recent times. New consensus protocols regularly emerge, as do the improvements, thus ensuring that the challenges and applicability of blockchain consensus protocols are continual. With regard to realistic consensuses, these are definitely not ideal (flawless). Indeed, they are always subjected to trade-offs concerning the efficiency of the performance, security, and scalability of the blockchain. Prioritizing a specific feature over another is made with regard to the specific purpose that it is serving. More precisely, when a new consensus algorithm is proposed, it favors miners or validators with specific attributes; either it gravitates toward the activity, the amount of staked money, or the hardware capacity in the network. Whenever a new consensus occasionally emerges, it only changes its favorite mining attributes. When considering such a situation, it is notable that those attributes do not differ from the circumstances which enable miners with the same attributes to always reap the most rewards. As a trend, this would leave the blockchain open to risks such as the 51% attack, where limited parties can monopolize the system. This has become a possible scenario with the existence of mining pools. Figure 11 shows that this has occurred; some mining pools have reached an alarming hashing rate, where, for a short, limited period of time, some have been able to monopolize nearly 51% of the network's hashing power. Moreover, the nodes which have the most favored attributes, and are thus eligible to add a new block, can abuse this power; this is because it is continually possible for them to add new blocks to the ledger, and therefore, they are also able to include their desired transactions. They can boycott less powerful users′ transactions and never commit them to the ledger. In such a scenario, the least favored nodes would see themselves powerless, and they would be obliged to leave the network; however, a new approach, if attempted, may lead to better performance. The main changes would require varying the attributes of the favored miners. This would be ascertained in accordance with the circumstances of the less powerful nodes, or such nodes would be favored, but only if they can prove that they have been a victim of abuse (e.g., falling victim to boycotts). In actuality, negligence when committing valid transactions is a concrete concern in most blockchain platforms. In the case of Bitcoin, where the maximum block size is 1 megabyte, given the daily average number of transactions per block and the average transaction size per day, one can therefore deduce the average number of transactions in a 1-megabyte block.

From Table 7 [93,94], it is evident that even though there are still pending transactions left over, in the Mempool, blocks are not filled to their full authorized potential. This is because if a transaction is left over for too long, without being confirmed, it is pushed away from the Mempool by the new transactions coming in. In forum discussions for Bitcoin cryptocurrency users [95,96], some users vent their frustration regarding some of their transactions as they have been pending for days, awaiting confirmation. In terms of confronting this problem, however, the only solution that is usually proposed is to raise the transaction fees when attracting miners.

Table 7. In the Mempool, blocks are not filled to their full authorized potential despite still having pending transactions left over.

Date	24 April 2022	5 May 2022	30 May 2022	6 June 2022
Average transaction size (Bytes)	308	479	713	396
Average number of transactions per block	1709	1973	1248	1235
Number of transactions if the block size was 1MB	3246	2087	1402	2525
Size of Pending transactions in the Mempool (Kilobytes)	80	30	20	0.5

To sum up, the more often that random miners become eligible to mine a block, the less likely it is for a given party to monopolize the mining process. Similarly, if less favored nodes are subject to abuse or bullying by powerful nodes, the consensus protocols should enable them to confront it, thus giving them proof of abuse.

Figure 13 illustrates the market share of each mining pool. It uses various circle sizes to represent periods of time, and a color scale is used to show the dominance of mining power in the Bitcoin network. No dominance (0% mining power) is depicted in light yellow, whereas complete dominance (>50%) is illustrated in dark red [97].

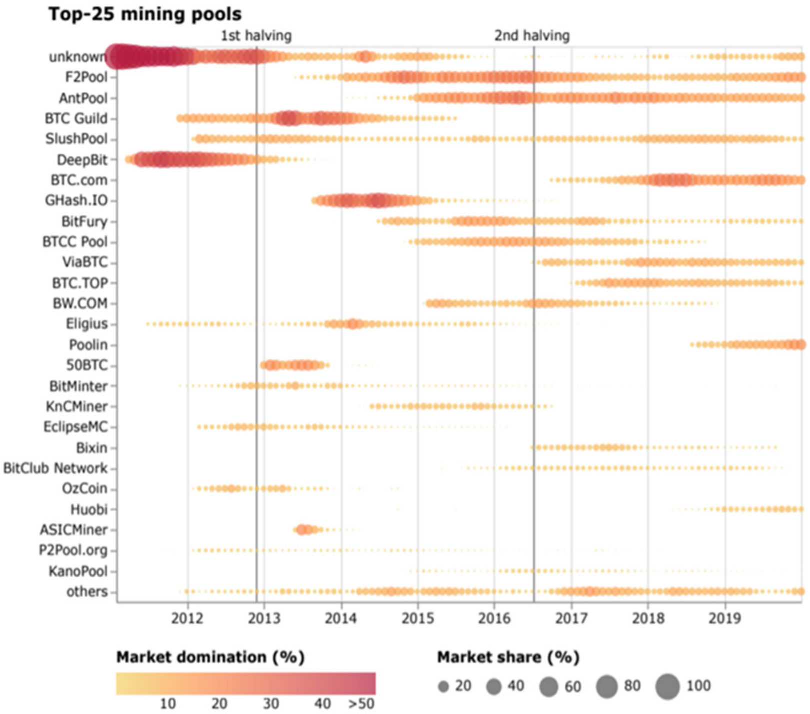


Figure 13. Evolution of market shares and domination for the top 25 mining pools.

6. Proposed Solutions

6.1. Random Selection of Miners, Our Proposed Solution, Preventing 51% Attacks

When blockchain technology first emerged, the 51% attack was only a potential threat that was technically possible but not realistic [98]; however, 51% attacks are real and many blockchain cryptocurrencies that mainly use PoW consensus algorithms are affected. In January 2019, Ethereum Classic fell victim to a 51% attack that resulted in double spending [99]. During July to August 2020, Ethereum Classic again fell victim to multiple 51% attacks. Bitcoin SV was also a victim of a 51% attack. The 51% attack was detected by an independent cryptocurrency data aggregator who denounced it in a tweet [100].

All the above mentioned 51% attacks were overcome after forking was noticed in the tampered blocks; however, the fact that such attacks keep occurring shows how vulnerable such platforms are, and that a more permanent preventive solution is needed.

A fully randomized selection of miners would completely eradicate the 51% threat, making the blockchains immune to such attacks; however, this would require an unpredictable random number generated by an impartial entity such as a smart contract. In such a circumstance, miners could apply to mine the next block by invoking a smart contract. Candidates can be stored in an array structure in the smart contract, then, the candidate with the index that corresponds to the randomly generated number would be selected as the next block miner; however, such an approach poses several challenges:

- First, interested miner candidates must call for a special smart contract function in order to apply; however, calls for smart contract functions are considered to be transactions, and in accordance with the specific function call, the state of the smart contract cannot be updated in the public ledger until this last call is also confirmed in the ledger. Hence, miner applications must first be mined to be considered.
- Second, it is not especially easy to generate a completely random number with a smart contract. In fact, there are only three ways to generate pseudo-random numbers with a smart contract, and none of them are completely random. The first way is to use the

current block hash, block hash size, or recent mining difficulty, all of which are affected by the throughput of the block, over which the miners have some control. In fact, it is the miners who calculate the block hash, which can lead to possible collusions among miners who wish to regulate the throughput of the block; therefore, this method is risky, as it means that it is possible to predict the generated random number. The second option is to assign an off-chain oracle to generate the random number that would be used in the smart contract; however, this approach is fully centralized, and it contradicts the spirit of decentralization inherent in the blockchain. It also opens loopholes for corrupt centralized entities, which becomes even more important if financial stakes are tied to the generation of the random number. The last approach, and the most popular one, is to use an oracle-based VRF. The argument in favor of such a method, stipulates that the oracle cannot know when the random number is requested; therefore, it uses the timestamp of the request as the seed for the VRF, which allows it to generate the random number that is sent to the smart contract, along with verifiable proof that it was randomly generated using the VRF. The VRF consist of three different functions. They are as presented in Equation (7) [101]:

$$\begin{gathered} Generate(x) = \left(P^{Key}, S^{Key}\right) \\ F\left(x, S^{Key}\right) = y \\ PROOF\ \left(y,\ P^{Key}\right) \end{gathered} \tag{7}$$

The VRF must satisfy the three (3) conditions mentioned in our previous work, namely [101]:

- Uniqueness: $\nexists\ (x_1,\ y_1),\ (x_2,\ y_2)\ :\ PROOF\ \left(y_1, P^{Key1}\right) = PROOF\ \left(y_2, P^{Key2}\right)$
- Pseudo-randomness: Assuming a set of n inputs $S_{input} = \{x_1,\ \ldots,\ x_n\}$, and the corresponding set of outputs $S_{output} = \{y_1,\ \ldots,\ y_n\}$, it is not possible to extrapolate a mathematical relation that allows one to infer the outputs from the respective inputs.
- Provability: the first function takes a seed input x, to generate a public/secret key pair $\left(P^{Key}, S^{Key}\right)$. The second function generates a random output from the seed input and S^{Key}. The third function serves as a proof of randomness for $F\left(x, S^{Key}\right)$; it allows the verification of the randomness of the output y based on P^{Key}. Although VRF is an established method for random number generation in smart contract applications [102,103], it still uses a seed generated by the smart contract, which is not fully random and can be subject to manipulation.

Moreover, our proposal is that a smart contract should be used to generate random numbers. In the proposed system:

- The smart contract must have a feature that allows miners to apply for mining.
- The allowed number of admitted candidates must be defined in the smart contract.
- An array must be available to store mining candidates.
- A function that allows mining candidates to send an encrypted seed number must be available.
- A function that allows mining candidates to reveal their seed number must be available.
- There must be an internal function to verify that the mining candidates′ encrypted seed numbers match the disclosed seed number.
- There must be a function to calculate a random number based on the candidates′ committed seeds.

The Genesis block, together with the smart contract, represents the first two blocks of the chain. The next block is mined using a PoW algorithm or another conventional consensus algorithm; however, the mined block must contain transactions that are part of the miner application, in accordance with the permitted applications, as defined above. In other words: if the defined number of applications is 50, there must be 50 transactions that correspond with 50 different mining applications. After the mining applications are committed to the ledger and stored in the dedicated array data structure, the next block to be mined must contain the corresponding transactions of all the encrypted seed numbers of the mining applicants. Then, the next block must contain the request for all mining candidates to disclose their encrypted seed number. The Disclose Seed Number function is locked (cannot be invoked) until all the candidates' encrypted messages have been committed to the general ledger. When the smart contract function requesting the disclosure of the candidates′ encrypted seed numbers is called by all miner candidates, the smart contract first checks whether each candidate has disclosed a seed number that matches the previously sent encrypted message. Those who have sent non-matching numbers are disqualified. This process is shown in the flowchart in Figure 14.

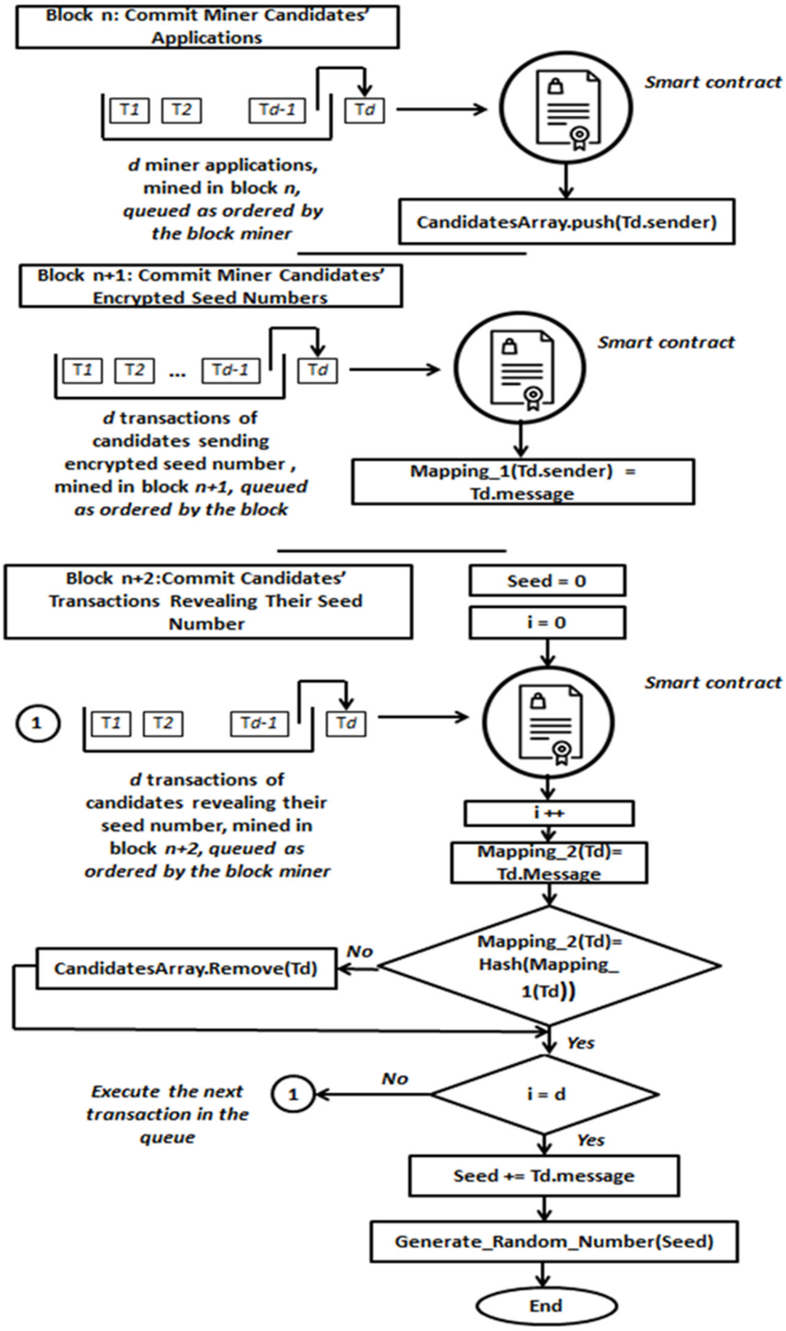

Block n: Commit Miner Candidates' Applications
T1
T2
Td-1
Td
Smart contract
d miner applications, mined in block n, queued as ordered by the block miner
CandidatesArray.push(Td.sender)
Block n+1: Commit Miner Candidates' Encrypted Seed Numbers
T1
T2
...
Td-1
Td
Smart contract
d transactions of candidates sending encrypted seed number , mined in block n+1, queued as ordered by the block
Mapping_1(Td.sender) = Td.message
Block n+2:Commit Candidates' Transactions Revealing Their Seed Number
Seed = 0
i = 0
1
T1
T2
Td-1
Td
Smart contract
d transactions of candidates revealing their seed number, mined in block n+2, queued as ordered by the block miner
i ++
Mapping_2(Td)= Td.Message
Mapping_2(Td)= Hash(Mapping_ 1(Td))
No
CandidatesArray.Remove(Td)
Yes
Execute the next transaction in the queue
1
No
i = d
Yes
Seed += Td.message
Generate_Random_Number(Seed)
End

Figure 14. Process of selecting a random miner, wherein one new block is added to a sequence of four blocks.

Moreover, in theory, if all the candidate miners have different permissions, they are counterparties who would never collude. Each miner encrypts a random number of their choosing, and they transmit it to the smart contract. Ultimately, candidates′ numbers are committed to the blockchain′s public ledger; only they can reveal their number. If this is the case, it generates a random number by adding the seed of the candidates. This scheme ensures completely decentralized and unpredictable random number generation, thus allowing miners to select one of four blocks completely at random. The random number can be generated by hashing the seed, as shown in expression (8)

$$Random\_Number = Hash(Seed) \quad \%CandidatesArray.length() \tag{8}$$

The selected miner is the candidate with the index "$Random_{Number}$" in the candidates' array. As is evident, a miner can be randomly selected to mine one of four blocks. The process of randomly selecting a miner is composed of a sequence where three blocks must be mined using conventional consensus protocols with additional conditions (as explained earlier), then, only the fourth block can be mined by the randomly selected candidate.

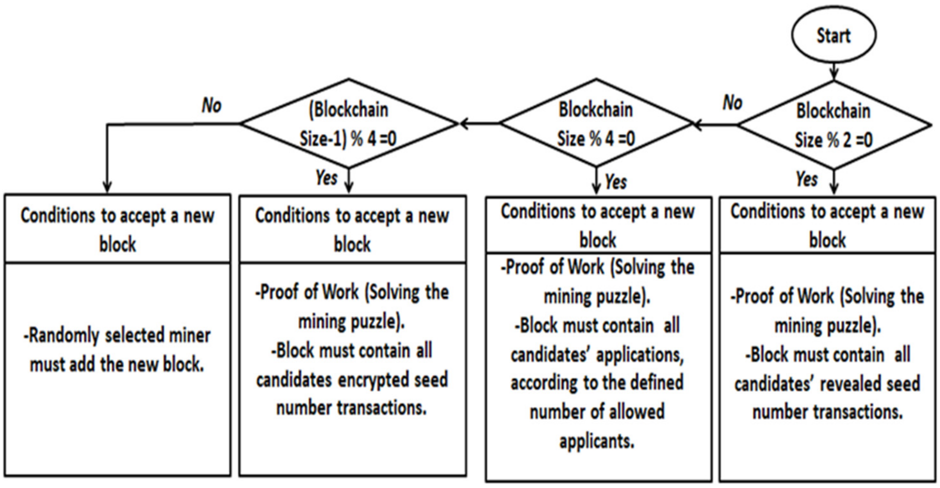
In a random selection sequence, three types of blocks can be distinguished:

- The block containing the applications of the mining candidates BA.
- The block containing the encrypted seed number transactions of the mining candidates BE.
- The block containing the disclosed seed number transactions of the miner candidates BR.
- The block generated by a randomly selected miner BW.

Aside from the first two blocks (i.e., the Genesis block and the Smart Contract Deployment block), for a random selection sequence i to account for each of the previously defined blocks, the block number in the blockchain would be N_i for the given sequence, as presented in Equation (9):

$$\begin{cases} N_i\ (BA) = ((i-1)*4)+1; \\ N_i\ (BE) = ((i-1)*4)+2; \\ N_i\ (BR) = ((i-1)*4)+3; \\ N_i\ (BW) = ((i-1)*4)+4 \end{cases} \quad i \geq 1 \tag{9}$$

As is evident, $\forall\ i > 1\ (N_i\ (BA) - 1)$ is always a multiple of four and two, whereas $(N_i\ (BR) - 1)$ is always a multiple of two and never of four. $(N_i\ (BE) - 1)$ and $(N_i\ (BW) - 1)$ are odd numbers; thus, the conditions for a block can be defined in accordance with the newly proposed block number in a verifiable manner, as shown in the flowchart in Figure 15.



Start
Blockchain Size % 2 =0
No
Yes
Conditions to accept a new block
-Proof of Work (Solving the mining puzzle).
-Block must contain all candidates' revealed seed number transactions.
Blockchain Size % 4 =0
Yes
Conditions to accept a new block
-Proof of Work (Solving the mining puzzle).
-Block must contain all candidates' applications, according to the defined number of allowed applicants.
(Blockchain Size-1) % 4 =0
No
Yes
Conditions to accept a new block
-Proof of Work (Solving the mining puzzle).
-Block must contain all candidates encrypted seed number transactions.
Conditions to accept a new block
-Randomly selected miner must add the new block.

Figure 15. How peers can check if a newly added block is valid based on the block numbers' properties.

6.2. Proof of Abuse, Our Proposed Solution against Transaction Boycotting

The blockchain is purposely designed so that not everyone is eligible to add a new block. Indeed, miners must prove that they have invested a valuable asset in order to be able to add a new block, which may be futile if the block is not validated; however, it must provide options to casual users who join the network only to receive and send transactions, and if they are proven to be victims of abuse, as explained earlier. A practical aspect of the abuse that such users may suffer, is that their transactions are neglected and not transferred to new blocks. This can be frustrating, especially if it is an urgent transaction that needs to be issued or received. It has been suggested that miners should be entitled to a reduced difficulty level if they see their transactions being neglected and if the block they are proposing contains older transactions; however, transaction timestamps are not reliable until they are in the public ledger, and therefore, it is difficult to know how old transactions in a newly proposed block are. Nevertheless, a UTXO-based blockchain provides reliable knowledge of newly issued, uncommitted transactions; this is because their input must be a committed transaction in the public ledger. Indeed, UTXO blockchains have a coin age [104], which is only relevant for UTXO-based blockchains. It refers to the age of transaction inputs, wherein age is measured in blocks. Nevertheless, it is important that such a system is not exploited by powerful nodes that intentionally age their transactions to enjoy a lower difficulty. One solution to this issue might involve setting a minimum time threshold for new blocks to be accepted by miners who bargain the difficulty, thus ensuring that powerful nodes are not tempted.

6.3. Proposed Solution against Selfish Mining

Despite the problem of high-power consumption in the PoW consensus, PoW algorithms would be infallible in terms of security if the problem of selfish mining was solved. Honest users are most affected by fraudulent behavior, and therefore, we believe that they are the ones who need to take proactive and responsible action against it. One way to do this, is for every honest user, for whom the longevity of their chosen blockchain platform matters, to keep an off-chain record of their last block hash and the size of their last blockchain transaction. As a result, they will be able to keep track of blocks with mismatched hashes. When an honest node receives an updated blockchain ledger, they can check if the last recorded hash matches the hash of the block number that corresponds with the last recorded blockchain size. If the hashes match, the node can update their records with the most recent block hash and blockchain size. On the other hand, if the hashes do not match, it means that selfish mining has taken place. To detect selfish mining, the honest node must find the last valid block before mining. To do this, it loops through the blockchain blocks until it finds the block that was used to calculate the last recorded hash. How an off-chain record can help detect selfish mining and recover the last block before it happens is illustrated in Algorithm 1 and Figure 16, respectively.

Algorithm 1. Detect_Selfish_Mining

```
Variables
    Uint $S$ /*The Honest User Last Recorded Blockchain Size*/
    Bytes32 $H$ /*The Honest User Last Recorded Valid Block Hash*/
    Bytes32[] $Blockchain$ /*An array storing the block hashes of
    the new blockchain, indexed by the respective block number */
End Variables
Procedure: $DetectSelfishMining(S, H)$ Returns Bool: $SM$
    If ($Blockchain[S] != H$) Then
        Returns $SM$ = True /*Selfish mining occurred*/
    Else
        Returns $SM$ = False /*Selfish mining occurred*/
    End If
End Procedure
```

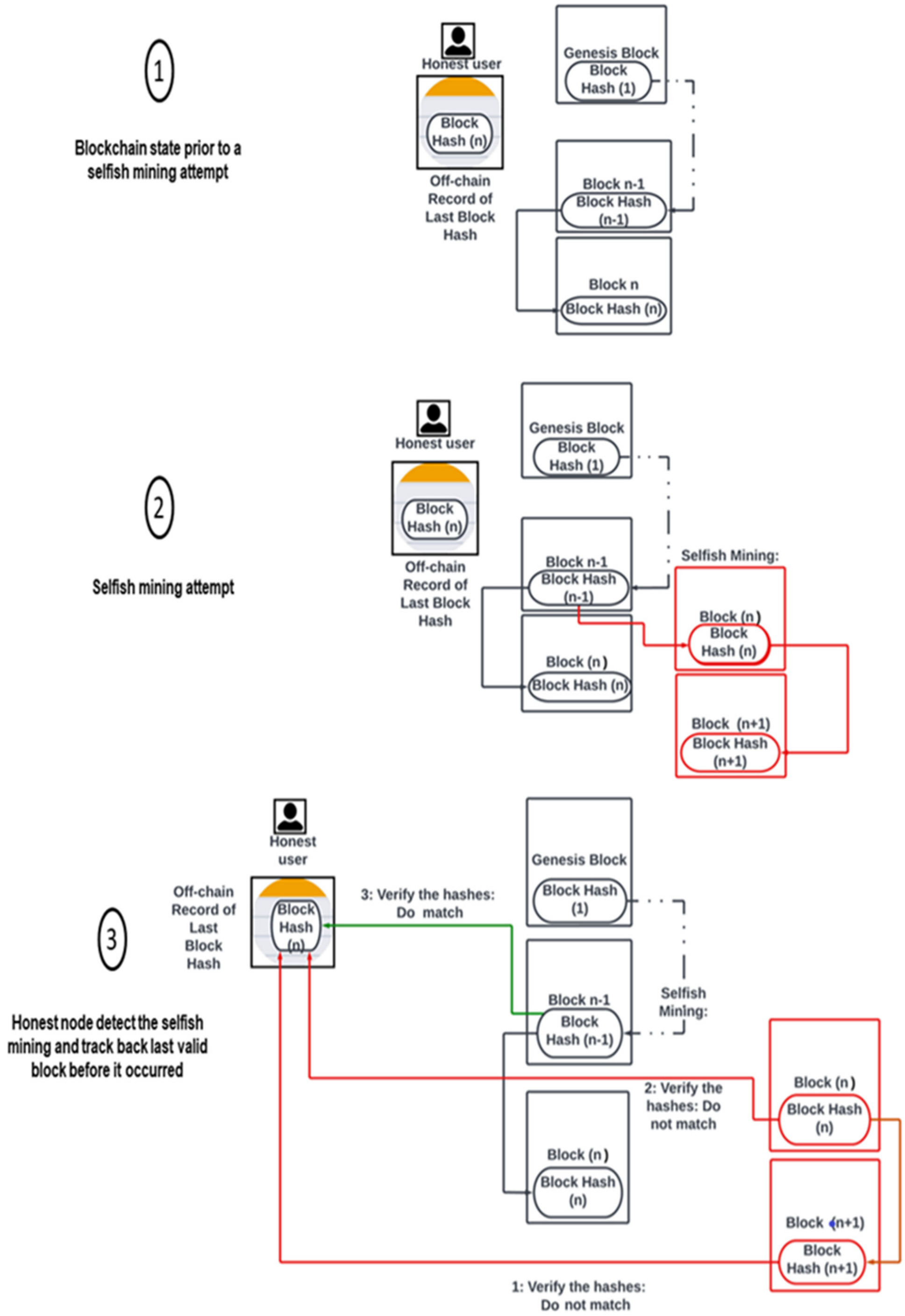


Figure 16. Last block hash and off-chain record scheme against selfish mining.

If honest nodes can detect the last valid block before selfish mining takes place, they can perform a hard fork on that block and migrate off the main chain. Indeed, this is possible, and mainstream blockchain platforms have seen several hard forks where users have migrated from the main blockchain to a forked branch. Table 8 shows some of Bitcoin's hard fork history. Bitcoin hard forks occur primarily when a group of users agree on protocol upgrades that remove or relax rules on the main chain.

Table 8. Bitcoin's hard fork history [105].

Bitcoin Fork	Date of the Fork	Block on the Main Chain wherein the Fork Occurred
Bitcoin Cash	1 August 2017	Forked at block 478,558
Bitcoin SV	15 November 2018	Forked at block 556,766
eCash	15 November 2020	Forked at block 661,648
Bitcoin Gold	24 October 2017	Forked at block 491,407

6.4. Proposed Solution against SYBIL Attacks

A Sybil attack refers to a single user operating with multiple active identities (or Sybil identities) that are considered to be distinct entities in a peer-to-peer network. This type of attack allows a single user (or group of users), who has become the most influential in the network, to undermine the reliability of the system without being detected. In a blockchain network, if each user is identified only by a single and unique address, users who monopolize the network could be easily detected, as the number of blocks mined by each user could be accurately verified by all peer nodes. In this regard, several solutions have already been proposed [106,107]; however, we believe that Proof of Burn is the most secure and efficient solution [108].

In essence, PoB requires sending a certain amount of cryptocurrency to an address on the blockchain, which cannot spend the currency, in order to mine a block. Unlike PoS, the PoB deposit is non-refundable. We propose that burning is not only required to mine a block, but rather, it should be considered as a registration fee for all nodes. Transactions would therefore have to include a PoB in order to be validated. As a result, transactions would need to be authenticated by the sender′s digital signature, along with the PoB. If priced appropriately, token burning may ensure that it is too expensive for a Sybil attacker to duplicate identities; this would continue for as long as the tokens are burned with each new identity. Such a system is also relevant to the proposed system of random mining, as it reduces the likelihood that mining applicants are different peer users.

6.5. Proof of HashRate PoH, Our Proposed Solution to PoW′s Excessive Energy Consumption

We believe that PoW is the consensus algorithm used by the highest market capital blockchain platforms for a reason, namely, that it is the most robust consensus algorithm to date. We believe that PoW′s drawbacks, such as the high energy consumption, can be addressed without changing its core concept, which requires miners to be the first to solve a mathematical puzzle in order to mine a new block. An example of work along these lines can be found in [109], wherein the authors propose energy-efficient hashing hardware. Although much work has focused on developing energy-efficient hardware, we believe that this is not sufficient to curb miner energy consumption, as there is no mechanism to control the hardware used by miners in a blockchain network. If the hashing performance of miners can be verified on a consortium basis, the network could comprise a defined, fixed mining difficulty that regulates the time taken for mining blocks at a fixed and known hashing rate. Even if the miners had the same hashing performance, the mining would still involve an element of randomness, since the transactions selected by the miners and their order would affect the nonce that produces a valid digest that satisfies the difficulty constraints. Two blocks mined by different miners cannot be identical because each block contains a reward transaction sent to that particular miner. A fixed hashing rate imposed on the network would result in fixed energy consumption during mining, and thus, such a solution could solve the hashing races that are prevalent with PoW algorithm use, in addition to solving the resultant increase in energy consumption.

As previously explained, miners try different nonce values to find a valid hash. The nonce is a 32-bit integer value. How a valid hash is found by miners is shown in Figure 17.

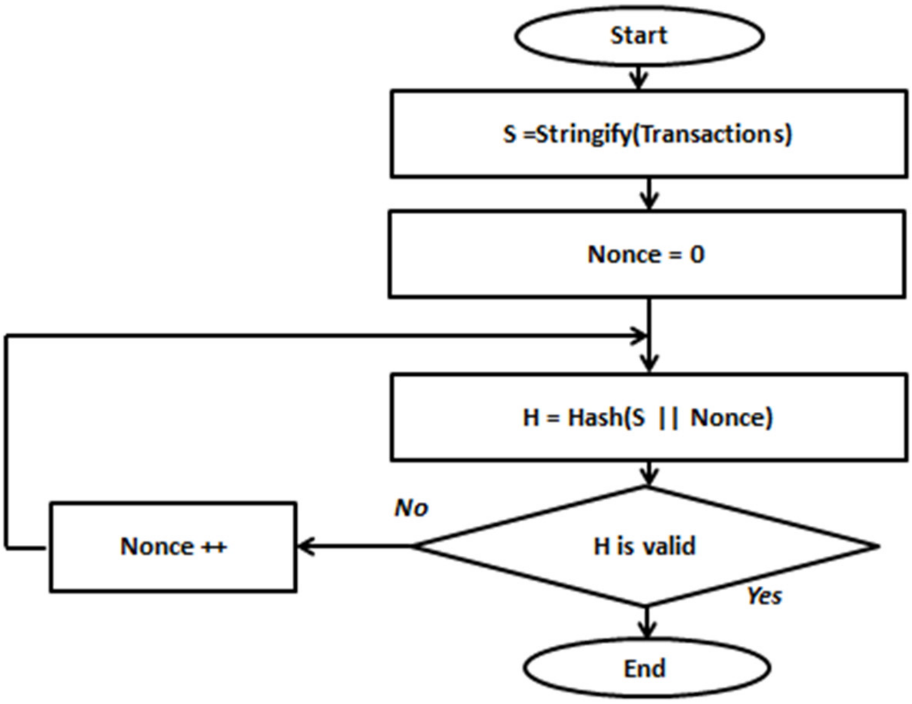


Figure 17. Mining Process: trying different nonce values until a valid hash is found.

As shown in Figure 17, the nonce N that produces a valid hash represents the exact number of hashes that were performed in order to find the valid hash; thus, the time T required to mine a block with a defined hash power H can be accurately derived by the simple relationship shown in (10).

$$T = H\ /N. \tag{10}$$

When a new block is mined, the nonce is used to generate the lock hash, and the block mining time is known to all peers in the network; therefore, the hash rate used to mine the new block can be easily verified. The newly proposed block should only be accepted if it was hashed in accordance with the allowed hash rate.

7. Future Eminent Threat to Current Blockchain Technology

In order to outline future trends, we believe that it would be undoubtedly useful to also shed light upon another challenge threatening blockchain technology and its applications. In fact, this challenge is relevant to all technologies using current cryptographic algorithms. Indeed, it is true that quantum computing is promising in terms of future technological developments. In fact, the anticipated advent of quantum computing would make current blockchain platforms vulnerable, meaning that they would require the implementation of new quantum-resistant cryptography [110,111]. Current blockchain technology uses cryptographic algorithms for key pair generation, such as RSA and elliptic curve cryptography. Key pair generation is essential for blockchain security, since key pairs are used for transaction signatures and their verification. This has worked, and continues to work, due to the fact that cryptographic functions are one-way; in other words, computers are currently incapable of isolating a private key from its respective public key. However, quantum computers are able to crack current cryptographic functions. In fact, Shor′s algorithm for quantum computing can be used to crack RSA cryptographic key pairs [112]. Quantum computers remain at the embryonic stage of research; however, quantum computing technology is expected to be operational for the next decade [113]. Moreover, researchers are working on developing new quantum-resistant cryptographic algorithms, as current algorithms would lose relevance in the era of quantum computer dominance. This new cryptography, which is invulnerable to quantum computing, is called post-quantum cryptography (PQC). In the context of blockchains, PQC requires the development of a new cryptographic signature, which needs to be immune to quantum computers. PQC is intended to be secure against quantum computer threats, in addition to being implementable on a conventional computer. Many proposals have been submitted to the National Institute of Standards and Technology NIST; however, only three post-quantum cryptographic digital signature algorithms have passed the three-phase selection process to become finalists. One algorithm will be selected as the new standard [114]. The remaining finalists are CRYSTALS -Dilithium [115], Falcon [116], and Rainbow [117].

8. Summary

In this article, we provided a critical overview of blockchain-based consensus algorithms. First, we presented the background of blockchain technology and its related applications. We provided an overview of blockchain technology, and we explained how it enables decentralized architecture to function in a peer-to-peer network, wherein nodes can interact in a reliable manner without having to trust each other or a central authority. Moreover, it is the architecture of the platform that ensures the integrity of the network and the data exchanged between nodes. A literature review was conducted on how decentralized cryptocurrencies are enabled or achieved, and applications and areas where this technology is most relevant and appropriate were also assessed. The focus of this work was consensus protocols which have significant market capital in blockchain technologies; thus, we presented and compared a variety of consensus algorithms, their advantages, and disadvantages.

As is evident from the findings of this paper, consensus, as an aspect of blockchain technology, is ripe for exploration, as no consensus protocol is perfect. We found that most consensus protocols in permissionless blockchain networks are proof-based, with the 51% attack being the biggest threat. After discussing and comparing many blockchain consensus algorithms, we discussed the performance metrics of the protocols, focusing mainly on the fairness of the protocols in relation to smaller clients (i.e., clients with fewer resources that want to participate in the network). Here, we presented a new strategy for measuring the fairness of these protocols using the Jains Fairness Index.

We also discussed the development trend of the consensus protocols and their drawbacks, in order to provide an outlook on existing works. We also proposed solutions to some of the identified problems.

Finally, we provided insight into how quantum computing might threaten current blockchain technology, which we may consider as more of an anticipated threat.

Author Contributions: Conceptualization, methodology, and writing, Y.M. and M.H.H.; software, Y.M.; validation, supervision, and revision: M.H.H., M.R.I., T.S.G., E.A.A.E., F.E.M.S. and M.M. Funding: M.H.H., M.R.I., E.A.A.E. and F.E.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through the Research Group Program under Grant Number (R.G.P.1/147/43).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was conducted at the IoT and Wireless Communication Protocols Laboratory at the ECE department, International Islamic University Malaysia (IIUM). Yaçine MERRAD is grateful to the IIUM Tuition Fee Waiver Program.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Salimitari, M.; Chatterjee, M.; Fallah, Y.P. A Survey on Consensus on Methods in Blockchain for Resource-constrained IoT Networks. *Internet Things* **2020**, *11*, 100212. [CrossRef]
2. Bouraga, S. A taxonomy of blockchain consensus protocols: A survey and classification framework. *Expert Syst. Appl.* **2020**, *168*, 114384. [CrossRef]
3. Zhang, J.; Zhong, S.; Wang, T.; Chao, H.-C.; Wang, J. Blockchain-based Systems and Applications: A Survey. *J. Internet Technol.* **2020**, *21*, 1–14.
4. Nofer, M.; Gomber, P.; Hinz, O.; Schiereck, D. Blockchain. *Bus. Inf. Syst. Eng.* **2017**, *59*, 183–187. [CrossRef]
5. Lamport, L.; Shostak, R.; Pease, M. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.* **1982**, *4*, 382–401. [CrossRef]
6. Monrat, A.; Schelén, O.; Andersson, K. A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access* **2019**, *7*, 117134–117151. [CrossRef]
7. Zhang, S.; Lee, J.-H. Analysis of the main consensus protocols of blockchain. *ICT Express* **2020**, *6*, 93–97. [CrossRef]
8. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System, Decentralized Business Review, 21260. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 28 June 2022).
9. Raval, S. *Decentralized Applications: Harnessing Bitcoin's Blockchain Technology*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2016; p. 118.
10. Aste, T.; Tasca, P.; Matteo, T.D. Blockchain technologies: The foreseeable impact on society and industry. *Computer* **2017**, *50*, 18–28. [CrossRef]
11. Biryukov, A.; Khovratovich, D.; Pustogarov, I. Deanonymisation of clients in Bitcoin P2P network. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 15–29.
12. Haber, S.; Scott Stornetta, W. How to Time-Stamp a Digital Document. In *Conference on the Theory and Application of Cryptography*; Springer: Berlin/Heidelberg, Germany, 1990; pp. 437–455.
13. Brown, D.R.L. Recommended Elliptic Curve Domain Parameters. In *Standards for Efficient Cryptography*, 3rd ed.; Certicom Research: Mississauga, ON, Canada, 2010; p. 33.
14. Blundo, C.; Lovino, V.; Persiano, G. Private-key hidden vector encryption with key confidentiality. In Proceedings of the International Conference on Cryptology and Network Security, Kanazawa, Japan, 12–14 December 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 259–277.
15. Hoy, M.B. An introduction to the blockchain and its implications for libraries and medicine. *Med. Ref. Serv. Q.* **2017**, *36*, 273–279. [CrossRef]
16. Padmavathi, M.; Suresh, R. Secure P2P intelligent network transaction using litecoin. *Mob. Netw. Appl.* **2019**, *24*, 318–326. [CrossRef]
17. Mukhopadhyay, U.; Skjellum, A.; Hambolu, O.; Oakley, J.; Yu, L.; Brooks, R. A brief survey of cryptocurrency systems. In Proceedings of the 2016 14th annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 12–14 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 745–752.
18. Coinmap. Available online: https://coinmap.org/ (accessed on 4 June 2022).
19. Xu, X.; Pautasso, C.; Zhu, L.; Gramoli, V.; Ponomarev, A.; Tran, A.B.; Chen, S. The blockchain as a software connector. In Proceedings of the 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), Venice, Italy, 5–8 April 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 182–191.
20. Swan, M. *Blockchain: Blueprint for a New Economy*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2015.
21. Verma, A.K.; Garg, A. Blockchain: An analysis on next-generation internet. *Int. J. Adv. Res. Comput. Sci.* **2017**, *8*, 429–432. [CrossRef]
22. Miller, D. Blockchain and the internet of things in the industrial sector. *IT Prof.* **2018**, *20*, 15–18. [CrossRef]
23. Kouhizadeh, M.; Sarkis, J. Blockchain Practices, Potentials, and Perspectives in Greening Supply Chains. *Sustainability* **2018**, *10*, 3652. [CrossRef]
24. Polge, J.; Robert, J.; Le Traon, Y. Permissioned blockchain frameworks in the industry: A comparison. *ICT Express* **2021**, *7*, 229–233. [CrossRef]
25. Boireau, O. Securing the blockchain against hackers. *Netw. Secur.* **2018**, *2018*, 8–11. [CrossRef]
26. Heineman, B.W. Who's Responsible for the Walmart Mexico Scandal. *Harv. Bus. Rev.* **2014**, *15*.
27. Mohamed, N.; Al-Jaroodi, J. Applying blockchain in industry 4.0 applications. In Proceedings of the IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; pp. 852–858.
28. Batubara, F.R.; Ubacht, J.; Janssen, M. Unraveling Transparency and Accountability in Blockchain. In Proceedings of the 20th Annual International Conference on Digital Government Research, Dubai, United Arab Emirates, 18–20 June 2019; pp. 204–213.
29. Luu, L.; Chu, D.-H.; Olickel, H.; Saxena, P.; Hobor, A. Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 254–269.
30. Christidis, K.; Devetsikiotis, M. Blockchains and smart contracts for the internet of things. *IEEE Access* **2016**, *4*, 2292–2303. [CrossRef]
31. Klomp, R.; Bracciali, A. On symbolic verification of Bitcoin's script language. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*; Springer: Cham, Switzerland, 2018; pp. 38–56.

32. Wang, S.; Ouyang, L.; Yuan, Y.; Ni, X.; Han, X.; Wang, F.Y. Blockchain-enabled smart contracts: Architecture, applications, and future trends. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2266–2277. [CrossRef]
33. Motepalli, S.; Jacobsen, H. Decentralizing permissioned blockchain with delay towers. *arXiv* **2022**, arXiv:2203.09714.
34. Elsden, C.; Nissen, B.; Jabbar, K.; Talhouk, R.; Lustig, C.; Dunphy, P.; Vines, J. HCI for blockchain: Studying, designing, critiquing and envisioning distributed ledger technologies. In Proceedings of the Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, CHI EA '18 2018, Montreal, QC, Canada, 21–26 April 2018; pp. 1–8.
35. Wood, D.D. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
36. Labazova, O.; Dehling, T.; Sunyaev, A. From Hype to Reality: A Taxonomy of Blockchain Applications. In Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS 2019), Maui, HI, USA, 8–11 January 2019; pp. 4555–4564.
37. Rouhani, S.; Deters, R. Performance analysis of ethereum transactions in private blockchain. In Proceedings of the 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 24–26 November 2017; pp. 70–74.
38. Neudecker, T.; Hartenstein, H. Network layer aspects of permissionless blockchains. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 838–857. [CrossRef]
39. Pahlajani, S.; Kshirsagar, A.; Pachghare, V. Survey on private blockchain consensus algorithms. In Proceedings of the 1st International Conference on Innovations in Information and Communication Technology (ICIICT), Chennai, India, 25–26 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
40. Suresh, A.; Nair, A.R.; Lal, A.; Kumaran, M.; Sarath, G. A Hybrid Proof based Consensus Algorithm for Permission less Blockchain. In Proceedings of the Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 15–17 July 2020; pp. 707–713.
41. Young, J. Why the Actual Cost of Mining Bitcoin Can Leave It Vulnerable to a Deep Correction. 7 June 2020. Available online: https://www.forbes.com/sites/youngjoseph/2020/06/07/why-the-actual-cost-of-mining-bitcoin-can-leave-it-vulnerable-to-a-deep-correction/?sh=28f377826067 (accessed on 28 June 2022).
42. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 3–16.
43. Kiayias, A.; Zindros, D. Proof-of-work sidechains. In Proceedings of the International Conference on Financial Cryptography and Data Security, St. Kitts, Saint Kitts and Nevis, 18–22 February 2019; pp. 21–34.
44. Cap, Today's Cryptocurrency Prices by Market. 3 July 2021. Available online: https://crypto.com/price (accessed on 28 June 2022).
45. Georgiadis, E. How many transactions per second can bitcoin really handle? Theoretically. *Cryptol. Eprint Arch.* **2019**.
46. O'Dwyer, K.J.; Malone, D. Bitcoin mining and its energy footprint. In Proceedings of the 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), Limerick, Ireland, 26–27 June 2014; pp. 280–285.
47. Criddle, C. Bitcoin Consumes 'More Electricity than Argentina. Cambridge University: 10 February 2021. Available online: https://www.bbc.com/news/technology-56012952 (accessed on 28 June 2022).
48. Ye, C.; Li, G.; Cai, H.; Gu, Y.; Fukuda, A. Analysis of security in blockchain: Case study in 51%-attack detecting. In Proceedings of the 5th International Conference on Dependable Systems and Their Applications (DSA), Dalian, China, 22–23 September 2018; pp. 15–24.
49. Zhang, S.; Lee, J.H. Double-spending with a Sybil attack in the Bitcoin decentralized network. *IEEE Trans. Ind. Inform.* **2019**, *15*, 5715–5722. [CrossRef]
50. Vasin, P. Blackcoin's Proof-of-Stake Protocol v2. 2014. Available online: https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf (accessed on 28 June 2022).
51. Nguyen, G.-T.; Kim, K. A Survey about Consensus Algorithms Used in Blockchain. *J. Inf. Processing Syst.* **2018**, *14*, 101–128.
52. King, S.; Nadal, S. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. *Self-Publ. Pap.* **2012**, *19*, 6.
53. Shibata, N. Proof-of-search: Combining blockchain consensus formation with solving optimization problems. *IEEE Access* **2019**, *7*, 172994–173006. [CrossRef]
54. Sameeh, T. *Two-New-Models-Double-Spending Attacks-Bitcoins Blockchain*; CoinDesk Inc.: New York, NY, USA, 2016.
55. Hanke, T.; Movahedi, M.; Williams, D. Introducing Dfinity Crypto Techniques. 19 May 2018. Available online: https://arxiv.org/abs/1805.04548 (accessed on 28 June 2022).
56. Rosic, A. What Is Ethereum Casper Protocol? 4 May 2020. Available online: https://blockgeeks.com/guides/ethereum-casper/ (accessed on 4 June 2022).
57. Jain, A.; Arora, S.; Shukla, Y.; Patil, T.; Sawant-Patil, S.T. Proof of Stake with Casper the Friendly Finality Gadget Protocol for Fair Validation Consensus in Ethereum. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. IJSRCSEIT* **2018**, *3*, 291–298.
58. Schwarz-Schilling, C.; Neu, J.; Monnot, B.; Asgaonkar, A.; Tas, E.; Tse, D. Three Attacks on Proof-of-Stake Ethereum. *arXiv* **2021**, arXiv:2110.10086.
59. Buterin, V.; Griffith, V. Casper the Friendly Finality Gadget. 2017. Available online: http://arxiv.org/abs/1710.09437 (accessed on 4 June 2022).

60. Li, W.; Andreina, S.; Bohli, J.M.; Karame, G. Securing proofof-stake blockchain protocols. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology, Proceedings of the ESORICS 2017 International Workshops, DPM 2017 and CBT 2017, Oslo, Norway, 14–15 September 2017*; LNCS; Springer: Cham, Switzerland, 2017; Volume 10436, pp. 297–315.
61. Larimer, D. Delegated Proof-of-Stake (DPOS), Bitshare Whitepaper. 3 April 2014. Available online: https://cryptorating.eu/whitepapers/BitShares/bitshares-general.pdf (accessed on 1 January 2021).
62. Beikverdi, A. NEM Launches, Targets Old Economy with Proof-of-Importance. 1 April 2015. Available online: https://cointelegraph.com/news/nem-launches-targets-old-economy-with-proof-of-importance (accessed on 28 June 2022).
63. Košt'ál, K.; Krupa, T.; Gembec, M.; Vereš, I.; Ries, M.; Kotuliak, I. On Transition between PoW and PoS. In Proceedings of the 2018 International Symposium ELMAR, Zadar, Croatia, 16–19 September 2018; pp. 207–210.
64. ANTMINER S9i. 2021. Available online: https://shop.bitmain.com/promote/antminer_s9i_asic_bitcoin_miner/specification (accessed on 4 June 2022).
65. Stratis Blockchain Explorer. 2021. Available online: https://chainz.cryptoid.info/strat/ (accessed on 4 June 2022).
66. NavCoin Blockchain Explorer. 2021. Available online: https://chainz.cryptoid.info/nav/ (accessed on 4 June 2022).
67. SpectreCoin. 8 October 2020. Available online: https://www.coinlore.com/coin/spectrecoin (accessed on 4 June 2022).
68. Zhang, X.; Qin, R.; Yuan, Y.; Wang, F. An analysis of blockchain-based bitcoin mining difficulty: Techniques and principles. In Proceedings of the 2018 Chinese Automation Congress, CAC, Xi'an, China, 30 November–2 December 2018; pp. 1184–1189.
69. Kwaasteniet, A.D. Miners, Block Time and Orphans, a Trinity. Available online: https://medium.com/coinmonks/miners-block-time-and-orphans-a-trinity-680f45f8dd42 (accessed on 12 June 2022).
70. Ethereum Hashrate. Available online: https://2miners.com/eth-network-hashrate (accessed on 12 June 2022).
71. Total Hash Rate (TH/s). Available online: https://www.blockchain.com/fr/charts/hash-rate (accessed on 12 July 2021).
72. DigiByte Hashrate Chart. Available online: https://www.coinwarz.com/mining/digibyte/hashrate-chart#:~{}:text=DigiByte%20hashrate%20is%20a%20calculated,per%20Second%20or%20H%2Fs (accessed on 12 June 2022).
73. Litecoin/Hashrate Chart. Available online: https://bitinfocharts.com/comparison/litecoin-hashrate.html (accessed on 12 June 2022).
74. Dash/Hashrate Chart. Available online: https://bitinfocharts.com/comparison/dash-hashrate.html (accessed on 12 June 2022).
75. Syscoin/Hashrate-Chart. Available online: https://www.coinwarz.com/mining/syscoin/hashrate-chart (accessed on 12 June 2022).
76. Firo (Zcoin) Hashrate. Available online: https://2miners.com/firo-network-hashrate (accessed on 12 June 2022).
77. Bitinfocharts/Blackcoin. Available online: https://bitinfocharts.com/blackcoin%20/ (accessed on 12 June 2022).
78. Karantias, K.; Kiayias, A.; Zindros, D. Proof-of-Burn. In Proceedings of the International Conference on Financial Cryptography and Data Security, Kinabalu, Malaysia, 10–14 February 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 523–540.
79. Architecture Overview. What Is the Sawtooth Lake Distributed Ledger? 2015. Available online: https://sawtooth.hyperledger.org/docs/core/releases/0.7/contents.html (accessed on 12 July 2021).
80. Salimitari, M.; Chatterjee, M. A survey on consensus protocols in blockchain for iot networks. *arXiv* **2018**, arXiv:1809.05613. Available online: https://ui.adsabs.harvard.edu/abs/2018arXiv180905613S (accessed on 28 June 2022).
81. Cai, X.; Ren, Y.; Zhang, X. Privacy-Protected Deletable Blockchain. *IEEE Access* **2019**, *8*, 6060–6070. [CrossRef]
82. Top PoS Tokens by Market Capitalization. Available online: https://coinmarketcap.com/view/pos/ (accessed on 15 July 2022).
83. Mazieres, D. The Stellar Consensus Protocol: A Federated Model for Internet-Level Consensus. *Stellar Dev. Found.* **2015**, *32*, 1–97.
84. Lamport, L. The Part-Time Parliament. In *Concurrency: The Works of Leslie Lamport*; ACM Digital Library: New York, NY, USA, 2019; Volume 16, pp. 277–317.
85. Omar, D.; Brousmiche, K.-L.; Durand, A.; Thea, E.; Ben Hamida, E. Consortium blockchains: Overview, applications and challenges. *Int. J. Adv. Telecommun.* **2018**, *11*, 51–64.
86. Sousa, J.; Bessani, A.; Vukolić, M. A byzantine faulttolerant ordering service for the hyperledger fabric blockchain platform. In Proceedings of the 48th annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Luxembourg, 25–28 June 2018; pp. 51–58.
87. Castro, M.; Liskov, B. Practical Byzantine fault tolerance. In Proceedings of the Third Symposium on Operating Systems Design and Implementation, New Orleans, LA, USA, 22–25 February 1999; pp. 173–186.
88. Sukhwani, H.; Martínez, J.; Chang, X.; Trivedi, K.S.; Rindos, A. Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric). In Proceedings of the IEEE 36th Symposium on Reliable Distributed Systems (SRDS), Hong Kong, China, 26–29 September 2017; pp. 253–255.
89. Sediq, A.B.; Gohary, R.H.; Schoenen, R.; Yanikomeroglu, H. Optimal Tradeoff Between Sum-Rate Efficiency and Jain's Fairness Index in Resource Allocation. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 3496–3509. [CrossRef]
90. Jain, R.; Chiu, D.M.; Hawe, W.R. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. *ACM Trans. Comput. Syst.* 1998; *Submitted*.
91. Hashrate Distribution. Available online: https://www.blockchain.com/charts/pools (accessed on 4 February 2022).
92. Ethereum Mining Pools Reviews. Available online: https://miningpools.com/ethereum/ (accessed on 4 February 2022).
93. Bitcoin Average Transactions per Block. 3 August 2021. Available online: https://ycharts.com/indicators/bitcoin_average_transactions_per_block (accessed on 4 February 2022).

94. Bitcoin Block Time Chart. 3 August 2021. Available online: https://markets.bitcoin.com/crypto/BCH/chart/transaction-size (accessed on 4 February 2022).
95. Bitcointalk.org. 3 August 2021. Available online: https://bitcointalk.org/index.php?topic=2947405.0 (accessed on 4 February 2022).
96. The Bitcoin Forum Index. 3 August 2021. Available online: https://forum.bitcoin.com/technical-support/unconfirmed-bitcoin-transaction-for-more-than-5-days-t57444.html (accessed on 4 February 2022).
97. Natkamon, T.; Soulié, N.; Isenberg, P. Visual analytics of bitcoin mining pool evolution: On the road toward stability? In Proceedings of the 11th IFIP International Conference on New Technologies, Mobility & Security, Paris, France, 19–21 April 2021; pp. 1–5.
98. Christian, B.; Lu, Y.; Zikas, V. A rational protocol treatment of 51% attacks. In *Annual International Cryptology Conference*; Springer: Cham, Switzerland, 16 August 2021; pp. 3–32.
99. Grimmelmann, J. All smart contracts are ambiguous. *JL Innov.* **2019**, *2*, 11.
100. Twitter Web App. CoinGecko, Twitter Web App. 4 August 2021. Available online: https://twitter.com/coingecko/status/1422905165531779073 (accessed on 15 July 2022).
101. Merrad, Y.; Habaebi, M.; Islam, M.; Gunawan, T.; Mesri, M. Robust Decentralized Proof of Location for Blockchain Energy Applications Using Game Theory and Random Selection. *Sustainability* **2022**, *14*, 6123. [CrossRef]
102. Shu, F.; Lei, K. Vger: A VRF based cross-chain mechanism for blockchains. *J. Phys.* **2021**, *1780*, 12038. [CrossRef]
103. Mathews, E.; Chacko, A.; Anagha, T. *BSCDL: A Blockchain Based Smart Contract Digitized Lottery Scheme*; EasyChair Preprint: Manchester, UK, 2020; Volume 19.
104. Quintyne-Collins, M. Short Paper: Towards Characterizing Sybil Attacks in Cryptocurrency Mixers. Available online: https://eprint.iacr.org/2019/1111 (accessed on 28 June 2022).
105. Yi, E.; Cho, Y.; Sohn, S.; Ahn, K. After the splits: Information flow between Bitcoin and Bitcoin family. *Chaos Solitons Fractals* **2021**, *142*, 110464. [CrossRef]
106. Khalil, M.; Azer, M.A. Crypto-SAP protocol for sybil attack prevention in VANET. In *Advances in Computer, Communication and Computational Sciences*; Springer: Singapore, 2021; pp. 143–152.
107. Swathi, S.; Modi, C.; Patel, D. Preventing sybil attack in blockchain using distributed behavior monitoring of miners. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6 July 2019; pp. 1–6.
108. Bochem, A.; Leiding, B. Rechained: Sybil-resistant distributed identities for the Internet of Things and mobile ad hoc networks. *Sensors* **2021**, *21*, 3257. [CrossRef]
109. Suresh, V.B.; Kattam, C.S.; Rajagopalan, S.; Zhou, T.Z.; Patel, A.K.; Rakha, R.; Gopalakrishna, N.K.; Mathew, S.; Hukkoo, A. Bonanza Mine: An Ultra-Low-Voltage Energy-Efficient Bitcoin Mining ASIC. In Proceedings of the 2022 IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, CA, USA, 20–26 February 2022; pp. 354–356.
110. Ikeda, K. Chapter Seven—Security and Privacy of Blockchain and Quantum Computation. *Adv. Comput.* **2018**, *111*, 199–228.
111. Kiktenko, E.O.; Pozhar, N.O.; Anufriev, M.N.; Trushechkin, A.S.; Fedorov, A.K. Quantum-secured blockchain. *Quantum Sci. Technol. IOP Sci.* **2018**, *3*, 35004. [CrossRef]
112. Cai, Z.; Qu, J.; Liu, P.; Yu, J. A Blockchain Smart Contract Based on LightWeighted Quantum Blind Signature. *IEEE Access* **2019**, *7*, 138657–138668. [CrossRef]
113. Mavroeidis, V.; Vishi, K.; Zych, M.D.; Jøsang, A. The Impact of Quantum Computing on Present Cryptography. *IJACSA Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 1–10. [CrossRef]
114. An Official Website of the United States Government. PQC Standardization Process: Third Round Candidate Announcement. 22 July 2020. Available online: https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement (accessed on 28 June 2022).
115. Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Dilithium: A Lattice-Based DigitalSignature Scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2018**, *2018*, 238–268. [CrossRef]
116. Kiningham, K.; Levis, P.; Anderson, M.; Boneh, D.; Horowitz, M.; Shih, M. Falcon—A flexible architecture for accelerating cryptography. In Proceedings of the 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Monterey, CA, USA, 4–7 November 2019; pp. 136–144.
117. Yasuda, T.; Sakurai, K. A multivariate encryption scheme with rainbow. In Proceedings of the International Conference on Information and Communications Security, Singapore, 29 November–2 December 2016; pp. 236–251.