Annals of Telecommunications (2022) 77:517–537
https://doi.org/10.1007/s12243-021-00896-2

A security and performance analysis of proof-based consensus protocols

Gabriel Antonio F. Rebello[1] · Gustavo F. Camilo[1] · Lucas C. B. Guimarães[1] · Lucas Airam C. de Souza[1] · Guilherme A. Thomaz[1] · Otto Carlos M. B. Duarte[1]

Received: 4 May 2021 / Accepted: 26 October 2021 / Published online: 15 November 2021

Abstract

Blockchain is a disruptive technology that will revolutionize the Internet and our way of living, working, and trading. However, the consensus protocols of most blockchain-based public systems show vulnerabilities and performance limitations that hinder the mass adoption of blockchain. This paper presents and compares the main proof-based consensus protocols, focusing on the security and performance of each consensus protocol. Proof-based protocols use the probabilistic consensus model and are more suitable for public environments with many participants, such as the Internet of Things (IoT). We highlight the centralization tendency and the main vulnerabilities of Proof of Work (PoW), Proof of Stake (PoS), and their countermeasures. We also analyze and compare alternative proof-based protocols, such as Proof of Elapsed Time (PoET), Proof of Burn (PoB), Proof of Authority (PoA), and Delegated Proof of Stake (DPoS). Finally, we analyze the security of the IOTA consensus protocol, a DAG-based platform suited for the IoT environment.

Keywords Blockchain · Consensus · Security

1 Introduction

Reaching consensus in distributed systems with asynchronous networks is a difficult problem that researchers have been studying for over 40 years. In 2008, however, Satoshi Nakamoto[1] revolutionized the field of distributed consensus by proposing the blockchain data structure and a new consensus model based on Proof of Work (PoW) [63]. Proof of Work does not require exchanging messages or knowing participants' identities to obtain consensus, which provides decentralization, pseudo-anonymity[2], and scalability at an unprecedented level in distributed systems. In Nakamoto's proposal, any person or organization can become a miner pseudo-anonymously, and thousands of nodes can participate in consensus rounds simultaneously using the Internet as a communication system. Due to the blockchain characteristics, researchers propose successful systems that use this innovative technology to provide security in several distributed applications such as network slices and multi-tenant domains [1, 66, 74, 76], access control [14, 15, 61], federated applications [65, 79], data sharing [40], and others.

Despite its innovation, Satoshi's Proof-of-Work protocol lacks the performance of centralized applications and incurs enormous energy expenditure. Several alternatives feature new proof-based protocols to replace the Bitcoin protocol in response to the performance limitations of proof of work. Nevertheless, the probabilistic nature of proof-based protocols, whether proof of work or alternative protocols, remains the primary source of protocol vulnerabilities. The non-determinism of consensus in proof-based algorithms allows a malicious agent to exploit the forks in the system and execute double-spending attacks against traders and brokers. An attacker can also exploit the fact that most proof-based systems use public peer-to-peer networks that operate over the Internet and, then, carry out attacks against the network or consensus participants.

[1] Satoshi Nakamoto is a pseudonym used by the creator or creators of the Bitcoin cryptocurrency. The real identity is unknown.

[2] The network nodes are identified by an asymmetric key pair, that provides a some level of anonymity. However, curious nodes can infer identity information based on blockchain history.

✉ Gabriel Antonio F. Rebello
gabriel@gta.ufrj.br

Otto Carlos M. B. Duarte
otto@gta.ufrj.br

[1] Grupo de Teleinformática e Automação, Univesidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

This paper presents and categorizes the main proof-based consensus protocols, addressing the performance, attacks, and security vulnerabilities of each protocol. Proof-based protocols are probabilistic consensus models that work on asynchronous communication systems such as the Internet. The probabilistic consensus is well suited to public applications, in which any user can participate in the consensus process. The paper focuses on the security of Proof of Work (PoW) and Proof of Stake (PoS), the most popular alternative proof-based protocol in cryptocurrencies and public blockchain platforms. The paper also compares the leading alternative cryptocurrencies and platforms that use probabilistic protocols, such as Hyperledger Sawtooth's Proof of Elapsed Time (PoET), Slimcoin's Proof of Burn (PoB), VeChain's Proof of Authority (PoA), and EOSIO's Delegated Proof of Stake (DPOS). Finally, we analyze the IOTA cryptocurrency security, which proposes an innovative data structure suited for micro-payments in an Internet of Things (IoT) environment.

This paper is an extended version of a previous conference publication [75]. In this article, we present detailed information about each consensus protocol operation. We describe new protocols, conceive comprehensive security analyses, and compare the advantages and disadvantages of each consensus protocol. We accomplish this with a complete discussion of the introduced protocols, presenting their most famous applications, scalability and throughput performance issues, and security vulnerabilities.

The remainder of the paper is organized as follows. Section 2 introduces the concept of consensus in distributed systems and the classification of deterministic and probabilistic consensus. Section 3 addresses the Proof of Work consensus protocol and analyzes possible attacks on the Bitcoin network. Section 4 describes Proof of Stake, the main alternative to Proof of Work, and outlines its security challenges. Section 5 presents and analyzes other known alternative proof-based consensus protocols. Section 6 presents and analyzes IOTA, a cryptocurrency that adopts a new concept of consensus based on directed acyclic graphs (DAG). Section 7 discusses and compares the security and performance of all the analyzed protocols. Section 8 presents works related to this paper. Section 9 concludes the paper by highlighting our main observations.

2 Classical consensus and Nakamoto's probabilistic consensus model

In general terms, consensus is the process by which a set of independent participants[3] reach a common decision that affects the entire distributed system. In this process, the consensus participants must communicate by exchanging messages either in a network or by using shared memory [4]. In blockchain-based distributed systems, the physical distance and lack of trust between participants obliges the use of network-based message exchanges. The messages of a consensus round use two generic primitives[4] [23, 38]:

- $propose(P, x)$: proposes a new input x to the set P of consensus participants. Only a special participant, the consensus leader, can issue this primitive;
- $decide(y)$: decides on a y output from the received input. Locally, each participant receives the input $\hat{x}$, processes it and decides on the output $\hat{y} = f(\hat{x})$ which, if there is consensus, will be equal to the final output y.

In an ideal scenario, consensus occurs whenever the leader proposes a new input x and every participant decides for the same output y. In practice, however, the participants may fail due to power outages or malicious behaviour, and messages can be lost in the network. In an unreliable environment, consensus through the two nentioned primitives occurs if and only if the following fundamental conditions are satisfied [35, 53, 54]:

- **Termination:** every correct consensus participant[5] decides on an output;
- **Agreement:** every correct consensus participant decides on the same output $\hat{y}$;
- **Validity:** if every consensus participant receives the value x as a proposal, then the final output $y = f(x)$;
- **Integrity:** every $\hat{y}$ local output by a correct consensus participant and the final consensus output y must have been proposed by a correct consensus participant.

In blockchain-based systems, the input x corresponds to a new block proposal that is yet to be approved through consensus. The $f(\hat{x})$ function corresponds to the validation method that the participants use to approve or reject the new block. The y output corresponds to the consensus-approved block that every participant must add to the blockchain to update the global state machine. Figure 1 depicts he adding of a block to the blockchain data structure. The protocol must handle failures of the network and of participants. A crash-faulty[6] participant stops responding to messages and fails to perform new operations during a consensus round [23, 38]. In the Byzantine failure model, the faulty participant can be a malicious agent and exhibit arbitrary

[3] This paper considers the terms nodes, pair, computer, component and process as synonyms for a consensus participant.

[4] The terms $\hat{x}$ and $\hat{y}$ refer to the local values of the consensus participant p and the terms x and y refer to the values as seen by an agent outside the system.

[5] A correct consensus participant is a participant that is not in a failed state.

[6] Some authors refer to crash faults as *fail-stop* failures. We consider both terms equivalent.

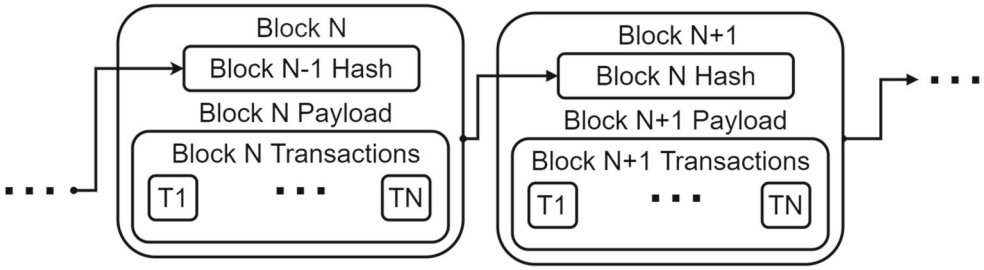

Fig. 1 A simplified blockchain data structure. Each block is securely linked to the previous block by the criptographic hash and every consensus participant stores a copy of the blockchain [75]

behavior that deviates from the protocol [17]. A malicious agent may issue correct, incorrect, or contradictory replies, in addition to not replying. The Byzantine failure model best captures participants' behavior in public blockchains because it is a hostile environment. Users can participate in the consensus rounds pseudo-anonymously and without the need for authorization.

The main objective of consensus protocols is to provide liveness and safety properties to the distributed system. The protocol guarantees liveness if it is certain that the consensus rounds always finish and, consequently, the system always adds new blocks to the blockchain. The safety property ensures that the added blocks are identical for all non-faulty participants and that a non-faulty participant proposed the block at the start of the consensus round. A fault-tolerant distributed system is always guaranteed to work correctly if and only if its consensus protocol provides both safety and liveness to the system. One of the main challenges in distributed systems, however, is the result of the impossibility of guaranteeing consensus, known as the FLP result[7] [35]. The FLP result proves that the consensus problem has no deterministic solution even in the presence of a single crash failure if the system operates over an asynchronous network like the Internet. For decades, consensus proposals circumvented the FLP result by assuming synchronous and partially synchronous communication systems, which provide different levels of guarantee of message delivery during a consensus round. Thus, classical consensus protocols focused on guaranteeing safety while trusting the communication system to deliver messages and provide liveness. Nevertheless, the protocols that depend on network synchronization do not meet the behavior of best-effort networks such as the Internet, in which there is no guarantee of message delivery and routing [21].

Since Nakamoto, there are two alternatives to circumvent the FLP result: ensuring safety, as the previous protocols did, or ensuring liveness by developing a proof-based algorithm that does not depend on synchrony to achieve a decision. Thus, two families of blockchain consensus protocols appear: deterministic and probabilistic consensus protocols. Protocols inspired by the classic deterministic consensus, such as Practical Byzantine Fault Tolerance (PBFT) [17], BFT-SMaRt [7], Tendermint [51], and Ripple [78], favor safety over liveness, creating consistent protocols that do not have forks. Unfortunately, deterministic protocols can halt if the communication system behaves asynchronously. Probabilistic consensus protocols, such as Proof of Work and Proof of Stake, favor liveness over safety by forcing a decision to occur even if it creates inconsistencies in the system. Any participant who provides correct irrefutable proof becomes the consensus leader in the probabilistic model and proposes the block. This approach dismisses the need for synchronous message exchanges but introduces a probability that two or more participants simultaneously provide proofs that propose different blocks, a fork. The system goal is to minimize such probability and develop a tie-breaking mechanism to eventually solve forks in the blockchain, e.g., the longest chain rule in Bitcoin. The probabilistic consensus is highly scalable since it is unnecessary to know all the participants or exchange messages in the network to reach consensus. Therefore, this type of consensus is better suited to public blockchains with many participants. The probabilistic approach led to the development of proof-based consensus protocols such as Proof of Work (PoW), Proof of Stake (PoS), Proof of Elapsed Time (PoET), Proof of Burn (PoB), Proof of Authority (PoA), Delegated Proof of Stake (DPoS), and others that power most cryptocurrencies today. We describe and address the main vulnerabilities of the main proof-based protocols in the next sections.

3 The Proof of Work (PoW) consensus protocol

Proof of Work (PoW) [63] is the first probabilistic consensus protocol, and it is used in the top cryptocurrencies in market value: Bitcoin and Ethereum. In PoW, a participant that proposes a block, henceforth called a miner[8], must provide proof that it can lead the consensus by spending resources to solve a computationally costly

[7] FLP is an acronym in honor of its authors: Michael J. Fischer, Nancy Lynch, and Mike Paterson.

[8] The name "miner" derives from the difficulty and enormous work required to overcome the mathematical challenge.

mathematical challenge. The cryptographic challenge of Proof of Work involves finding a nonce such that a hash function applied to the block and nonce results in a smaller number than a predetermined target. After solving the challenge, the participant broadcasts the block and the solution to the network. The other participants can easily verify the correct solution of the challenge by recalculating the block hash and checking the result. The minimum number of zeros in the starting bits defines the challenge's difficulty and is adjusted periodically to ensure a constant block creation rate. The winner of the challenge is well rewarded to encourage broad competition. Because participants mine independently, multiple miners may solve the challenge simultaneously, creating a fork in the blockchain and an inconsistent state in the system. Nakamoto's consensus introduces a tie-breaking mechanism that maintains the longest branch of the fork because it corresponds to the most significant number of solved challenges, which also corresponds to the most significant computing power and energy expenditure.

Table 1 summarizes the main advantages and disadvantages of Proof of Work. The main advantage of proof of work concerning the performance is high scalability since anyone can participate and mine blocks independently. Thus, public networks widely adopt PoW as a consensus protocol [6, 63, 81, 83]. The main disadvantages of proof of work are low transaction throughput, high confirmation delay, and high energy consumption. First, the addition of new blocks in Bitcoin shows an average throughput of one block per 10 min or seven transactions per second. This value is considerably less than the average of 2000 transactions per second recorded by credit card companies [8]. Second, legitimate blocks may be discarded after being confirmed to be in the blockchain due to the longest chain rule. Although the probability that the system discards a confirmed block decays over time, this means the user has to wait for several confirmations, each lasting around 10 min, to ensure his/her transaction is secured. The low throughput and high confirmation delay are the main performance characteristics that hinder the use of PoW-based cryptocurrencies for everyday purchases. However, the most critical drawback of PoW is the high computational cost involved in calculating Proof of Work in Bitcoin, which consumes an annual amount of energy that is comparable to the power consumption of Switzerland [29]. Most of the consumed energy is wasted because only the winner receives a reward, and even the winner can have its effort wasted if the system discards his/her block during a tie-break. The race for computing power in Bitcoin also leads to miner centralization because rich stakeholders build farms of hash power to obtain the rewards.

3.1 Proof of Work security analysis

High market-value cryptocurrencies use Proof of Work consensus, but the protocol presents many vulnerabilities. We classify the PoW vulnerabilities in categories: i) double-spending attacks, ii) attacks on consensus, and iii) attacks on the network.

Double-spending attacks aim to use the same currency in multiple transactions. Unlike physical currency, it is easy to replicate digital currency, and there is a risk of using the same currency more than once. Bitcoin proposes the blockchain structure that publicly stores all transaction history in a distributed and ordered manner to prevent double-spending [63]. Double-spending attacks, however, are still possible on the Bitcoin network [47]. An attacker A sends a transaction T_A^V to a seller V and a transaction T_A^A to an account controlled by the attacker. The time difference between the two transactions is $\Delta t \approx 0$. Then, a part of the network confirms the transaction T_A^V, and the seller V delivers the purchased product to the attacker. Meanwhile, the attacker publishes the transaction T_A^A with the help of multiple accounts to another part of the network, which confirms T_A^A. If a miner adds the transaction T_A^A to a block before adding the transaction T_A^V, the seller loses his/her product, and the attacker keeps his/her money.

Another way to double-spend is through the Finney attack, described by Hal Finney in a Bitcoin forum in 2011 [34]. In this attack, attacker A is a miner who issues a transaction T_A^A at a time $t_{T_A^A}$ to an account controlled by him/her, and mines a block B_A containing that transaction. The attacker then keeps the mined block for himself and sends a transaction T_A^V to a seller V at a time $t_{T_A^V}$. As the block B_A was not published and the transaction T_A^A was not validated, V accepts the transaction T_A^V and delivers the product to the attacker. After receiving the product, A publishes the block B_A containing the transaction T_A^A. Thus, as $t_{T_A^V} > t_{T_A^A}$, the network participants discard the transaction T_A^V, and V loses the product without remuneration.

The 51% attack on consensus consists of an attacker or group of attackers having more than 50% of the network's computational power since, in this case, the attackers can double spend. Although a 51% attack has never been successfully executed on Bitcoin, the four largest mining pools on the Bitcoin network already account for more than 50% of its computational power[9]. Collusion between only four independent entities would be able to subvert the system completely. Thus, contrary to the initial proposal of

[9] Available at https://btc.com/stats/pool. Accessed 15th March 2021.

Table 1 Main advantages and disadvantages of Bitcoin and Proof of Work consensus protocol

Advantages	Disadvantages
High scalability (thousands to millions of miners)	Extremely high energy expenditure
Works on asynchronous networks such as the Internet	Low throughput (∼7 transactions per second)
Provides pseudo-anonymity for users and mitigates Sybil attacks	High block confirmation time (∼10 min) and finality time (at least 1 hour)
Despite having many known vulnerabilities, the protocol never suffered a successful confirmed attack	Susceptible to centralization in mining pools and farms of ASIC

the decentralization of Bitcoin, four agents would centralize the power of the network. This type of attack occurred in alternative proof-based protocols[10,11].

Selfish mining [33] is an attack that exploits the consensus convergence algorithm and fork resolution. An attacker with a mining power of less than 51% of the network can adopt the selfish mining strategy to gain remuneration advantages or make double-spending attacks. For this, the malicious node mines and keeps new blocks confidential, creating a private blockchain. Eventually, the attacker shares his blocks to create forks, dividing the computational power of the miners. By creating a fork longer than that of honest miners, the malicious participant causes the network to converge on its state. In this way, the attacker can successfully execute double-spending attacks if he/she owns at least 25% of the total computational power of the network. Therefore, the miners who own blocks on old versions or abandoned forks in the blockchain waste computational resources attempting to find new blocks. The nodes forget all existing transactions in the abandoned fork if they do not exist in the attacker's blocks, allowing double-spending.

The block discarding attack [5] is an extension of the selfish mining attack that also targets the consensus. In this attack, the attacker controls a set of network nodes responsible for dropping newly discovered blocks as they are received. These nodes only publish the blocks obtained by the attacker, making selfish mining more effective by delaying the propagation of blocks proposed by other nodes in the network.

Finally, the bribery attack against consensus occurs when an attacker without sufficient computational power to attack the network bribes miners with higher processing capacity to form collusion during a given period [9]. Nevertheless, the network loses trust if the malicious node can use this strategy to carry out other attacks such as double-spending, thus devaluing the currency. Therefore, miners who are investors in the currency, since they own assets obtained by discovering new blocks, lose the money invested or have their profit reduced. Hence, the attacker must spend an amount that exceeds the losses to bribe miners, making the strategy expensive and impracticable in networks with high computational power.

Network attacks pose a significant threat to proof of work because of distributed blockchain environment, and the protocol allows for temporary inconsistencies. If the attacker is successful, network attack victims may remain in incorrect states for long periods due to a lack of information about the network global state.

Proof of Work mitigates the use of Sybil attacks, frequent in P2P networks such as those used in blockchains, to manipulate consensus. Since adding blocks to the blockchain depends on solving a computationally costly cryptographic challenge, creating new identities does not increase the likelihood that an attacker will solve the problem, as he/she will have to split the processing between his/her identities. Due to distributed communication, an attacker can create multiple identities to control the information delivered and sent by specific nodes. Thus, Sybil's attack can be applied to intermediate stages of more sophisticated attacks, such as selfish mining, double-spending, and eclipse attacks. We explain the latter below.

The eclipse attack [39] is another way of controlling information from part of the network. The malicious node creates several identities and forces its victim to add the accounts controlled by the attacker to the list of known nodes. Thus, if the victim only knows the attacker's nodes, the malicious participant starts to control the information and can create a local view different from the current state of the blockchain for the attacked node. Causing unavailability on the network requires enormous computational power and the knowledge of many participants due to the decentralization. Nevertheless, as some points in the network are more centralized, a Distributed Denial of Service (DDoS) attack can affect more important nodes, such as mining pool managers [45].

[10]The Bitcoin Gold cryptocurrency, at the time the 26^a largest currency, suffered a 51% attack in May 2018. The attackers double-spent for several days and stole more than US$18 million in Bitcoin Gold.

[11]The Krypton and Shift blockchains suffered 51% attacks in August 2016.

4 The Proof of Stake (PoS) consensus protocol

Proof of Stake (PoS) [13, 49, 56, 67] is the most widely-adopted alternative consensus category, as they provide similar characteristics to Proof of Work without requiring high energy expenditure. The main advantages of proof of stake over Proof of Work include increased energy efficiency and high performance, but PoS introduces new vulnerabilities and a tendency for centralization. Table 2 summarizes the main advantages and disadvantages of the most used public PoS implementations today.

Proof of Stake is a category of proof-based algorithms for public blockchains whose main characteristic is to achieve consensus based on each participant's amount of stake. Compared to Proof of Work, in which the probability of a participant proposing a block is proportional only to his/her computing power, in Proof of Stake, the probability of proposing a block is proportional to the number of coins that the participant stakes at the time of consensus. Due to the absence of "mining," i.e., spending computational power to obtain rewards, the PoS protocols introduce the concept of "virtual mining" and define its participants as validators or stakeholders instead of miners [80, 83]. In virtual mining, any participant who owns assets can become a validator by making their assets available as a deposit. Then, there is a round of consensus in which each participant's power is proportional to their respective deposits in relation to the total.

The implementation of a Proof-of-Stake consensus follows two main approaches: (i) a probabilistic approach, in which a participant with more stake is more likely to propose a block; or (ii) a deterministic approach based on a Byzantine agreement (BFT-based PoS), in which a set of validators confirms all the proposed blocks by voting with weights proportional to the stake of each validator [80, 83]. The bidder selection criterion is based on the stakes, as in the Ouroboros cryptocurrency [49], or on the election, as in the EOSIO cryptocurrency [56]. In addition to the two approaches, each consensus protocol presents specific details, such as how to incentivize validators and mechanisms to prevent attacks, which generates several practical ways to implement a Proof of Stake. Some PoS consensus protocols, called Bonded Proof of Stake (BPoS), require that participants deposit part of their stake to participate in the consensus protocol [51, 52]. In these type of PoS, participants lock their tokens for a period of time to obtain voting power in the consensus proportional to the amount of tokens locked. The bonded tokens can not be used during this period of time and may be destroyed in case of fault during a consensus round. As the amount of tokens at stake changes each round, BPoS participant set is dynamic and avoids centralizing power in few nodes in the network. Rather than looking at specific protocols, this paper focuses on the probabilistic approach to provide a general security analysis of Proof of Stake.

The probabilistic approach to Proof of Stake inherits characteristics similar to Nakamoto's Proof of Work [63], such as the pseudo-random selection of a participant to add a block, the longest chain rule, and the probabilistic finality. Bitcoin developers propose in 2011 the first family of probabilistic-based Proof of Stake consensus protocols, which today are known as Nakamoto-PoS or chain-based PoS. In this implementation, as in Nakamoto's proof of work, each participant must calculate a cryptographic hash. However, there is a limited time window, and the difficulty of the challenge decreases according to the participant's stake. Although the validation process is similar to the Proof of Work procedure, the average difficulty for solving the computational challenge is significantly lesser than that of Bitcoin. Therefore, PoS avoids the brute-force-based competition of Proof of Work, and, consequently, reduces energy costs.

More recent proposals such as Ouroboros randomly select validators that can propose blocks over some time. These protocols, known as committee-based PoS, use multi-party computation (MPC) to simulate a draw among the participants, giving more chances to participants with more stakes. The MPC receives the current blockchain state, which includes each participant's assets, and selects a pseudo-random sequence of upcoming bidders that any participant can verify. Participants can be chosen more than once and receive more time to propose blocks if they own more stake.

4.1 Proof of Stake security analysis

In the first Proof of Stake implementations, it is sufficient to own assets to participate and gain an advantage in the consensus process. However, the non-requirement of

Table 2 Main advantages and disadvantages of the Proof of Stake consensus protocol

Advantages	Disadvantages
Low energy expenditure	Introduces new vulnerabilities such as the nothing at stake problem and long range attacks
High scalability (thousands to millions of validators)	Susceptible to centralization in rich validators
Good throughput (hundreds to thousands of tx/sec)	Increases the probability of forks in the blockchain

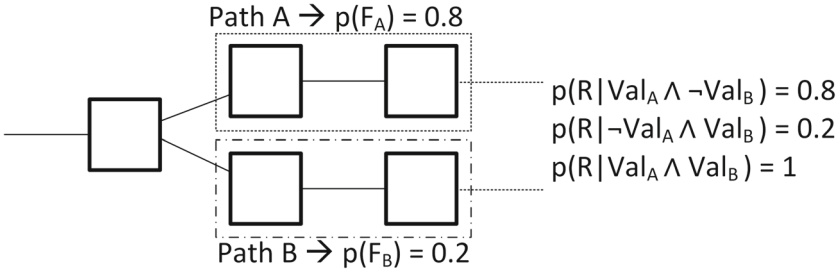

Fig. 2 A forked blockchain with two conflicting paths A and B with different probabilities of being finalized by the system. The best strategy for a participant to guarantee a R reward is to validate the two paths, contributing to the fork prolongation [75]

deposits allows the "nothing at stake" attack, in which participants can use assets to simultaneously participate in the validation of multiple conflicting blocks when a fork occurs. This behavior is the most advantageous and followed by any rational validator since there is no computational cost to validate transactions at multiple forks, in contrast with Proof of Work. The simultaneous validation of several forks becomes computationally efficient, which corresponds to a greater chance of winning without any risk of loss. Thus, the action that maximizes the probability of gains is to participate in all possible forks. Every rational participant who wants to maximize their profit follows this behavior.

We model the "nothing-at-stake" problem as a probability maximization problem to demonstrate this phenomenon. Let be a blockchain fork with two conflicting paths[12] A and B and a generic participant who owns a stake $s \in [0.1]$ of the total resources in the system. Figure 2 illustrates the problem scenario with conflicting paths.

The following possible events are defined:

- F_A: the system eventually finalizes[13] and abandons path A and path B.
- F_B: the system eventually finalizes path B and abandons path A.
- Val_X: the participant uses his/her resources to validate the path X.
- R: the participant wins the round and receives the agreed rewards.

In Proof of Stake, there is no expenditure of resources to validate one of the possible paths or mechanisms of punishment to avoid the validation of multiple paths. Thus, even though F_A and F_B are mutually exclusive events, the system allows the participant to use all their resources to validate both paths, i.e., $Val_A \wedge Val_B$, performing double stake without punishment. Considering each possible scenario, the rewarded odds of the participant are [75]:

$$p(R|(Val_A \wedge \neg Val_B)) = s.p(F_A), \quad (1)$$

when the participant validates only path A,

$$p(R|(\neg Val_A \wedge Val_B)) = s.p(F_B), \quad (2)$$

when the participant validates only path B, and

$$p(R|(Val_A \wedge Val_B)) = s[p(F_A) + p(F_B)], \quad (3)$$

when the participant validates both paths. Using the mutual exclusion property between F_A and F_B, the Equation 3 can be simplified, since $p(F_A) = 1 - p(F_B)$:

$$p(R|(Val_A \wedge Val_B)) = s[p(F_A) + 1 - p(F_A)] = s. \quad (4)$$

As $s > s.p(A)$ and $s > s.p(B)$, the expected value of validating both paths will always be greater than choosing only one of the paths. This behavior maximizes the likelihood of being rewarded in a round of consensus, which, consequently, maximizes the participant's long-term gains. This result shows that every rational participant in the system validates both paths. Consequently, the finality of one of the paths may not occur even without the presence of attackers. Besides, carrying out a double-spending attack becomes much easier since the attacker only needs to have more resources than altruistic participants[14]. In proof of work, this problem does not occur because the chance of mining a block does not increase when someone divides the computational power among the forks.

The primary countermeasure to the "nothing at stake" problem in the Proof of Stake protocols is the punishment of participants who validate two conflicting paths. Ethereum financially rewards users who discover conflicting votes from a misbehaving validator at any time. The system destroys all stake of a validator that confirms two conflicting paths and temporarily prevents it from participating in new block validation rounds.

Another vulnerability of Proof of Stake is the long-range attack, which aims to rewrite old blocks already accepted by the participants of the network [27]. To perform this attack on a blockchain $B = (b_0, b_1, b_2, ..., b_h)$, the attacker A must generate a fork at a height f prior to the current h length of chain. Thus, A generates a blockchain $B' = (b'_0, b'_1, b'_2, ..., b'_f, b'_{f+1}, ..., b'_{f_h})$ where $B = B'$ for blocks $b'_i, i < f$. In the generated fork, A copies several

[12] Conflicting paths are paths that start from the same source block and have the same height and, therefore, it is not enough to apply Nakamoto's rule of the largest chain [63].

[13] Finalizing a path means considering it as the correct path between conflicting paths.

[14] Altruistic participants are participants who preserve the proper functioning of the system, validating only one of the possible paths

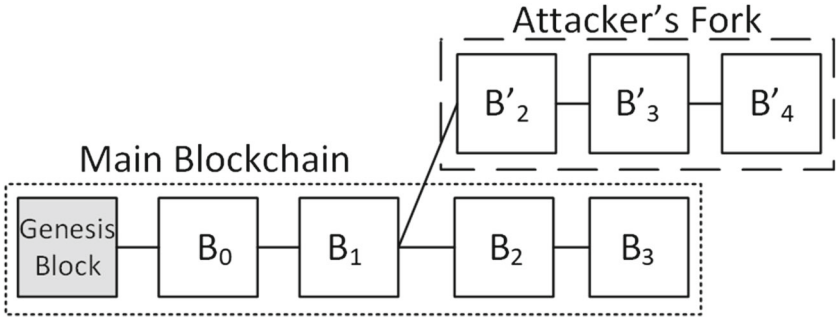

Fig. 3 Execution of a long-range attack [75]. The attacker creates a fork in a block accepted by the network and tries to rewrite the main chain

transactions from the main chain to maximize the reward for generating blocks. The attacker's goal is to mine blocks without revealing them to other participants, aiming to replace the main blockchain. The attacker A needs to control a significant portion of the network's assets at the time of the fork f. Long-range attacks take advantage of the low cost of building blocks to recreate block sequences longer than the main blockchain, easily subverting the longest chain rule. This attack is not effective on blockchains that use Proof of Work since the computational cost of rewriting the blockchain from the beginning is very high. Figure 3 illustrates the long-range attack. Checkpoints that restrict the blockchain at height before the checkpoint mitigates long-range attacks. This countermeasure limits the range of the attack by preventing attackers from generating forks at points very far from the main blockchain.

5 Proof-based alternatives: Proof-of-X (PoX)

The proof-based algorithms are alternatives to proof of work that seek to mitigate the performance limitations and excess energy expenditure of the Proof of Work. Besides, the proposals try to avoid the "nothing at stake" and the long-range attack problems of the Proof of Stake [49]. Follows explanations of the most well-known protocols.

5.1 Proof of Elapsed Time (PoET)

In the Proof of Elapsed Time (PoET) consensus protocol[15], participants need to wait a random time to propose a block [70]. Each round consists of a distributed lottery system in which every participant draws a random timer, and the first participant that can prove his/her timer has expired becomes leader. When a participant's timer expires, and he/she knows of no other expired timers, he/she propagates a signed certificate to the network indicating that he/she is the randomized block leader for that round. Table 3 exhibits the main strong and weak points of the PoET consensus protocol.

The system uses a Trusted Execution Environment (TEE) provided by Intel's Software Guard Extensions (SGX) technology [22] to ensure that adversaries cannot control the random-time generation algorithm. Therefore, the SGX-based environment must guarantee that (i) each participant honestly executes the random timer selection algorithm in a tamper-proof manner, and (ii) the system can correctly verify the proof provided by a winner participant that waited for the specified time. The chance of proposing a block in PoET is proportional to the number of trusted CPU provided by a participant. The random wait time provides a fair lottery system where two participants with the same number of trusted CPU achieve the same chance of being elected the leader [18]. Nevertheless, Stephan et al. demonstrate critical SGX vulnerabilities, which allow attackers to accomplish side-channel attacks and dump protected data [77]. This vulnerability leverages the need for validation mechanisms such as statistical tests to mitigate arbitrarily short wait times generated by a compromised CPU. Z-score metrics[16] allow every node to verify if the participants follow the expected probability distribution of being a leader throughout the rounds. However, Chen et al. demonstrated that an attacker, which controls a fraction, ϕ, of nodes, could follow the honest wait time distribution while replicating the fastest honest participants' behavior to control consensus [18]. The fraction ϕ is given by

$$\phi = \Theta(log(log(n))/log(n)), \tag{5}$$

where n is the number of nodes. Assuming a blockchain with 1000 participants, ϕ corresponds to 30% of network nodes. Hence, PoET becomes much more vulnerable to collusion than the PoW, which requires 50% of nodes, and ϕ decreases even further as the number of nodes increases.

PoET can reach more than 1000 transactions per second in small permissioned blockchains up to hundreds of nodes, which is a much higher throughput when compared

[15] PoET is the main consensus protocol used in the Hyperledger Sawtooth platform, which is maintained by the Linux Foundation.

[16] Z-score measures how much the winning rate deviates from the expected mean.

Table 3 Main advantages and disadvantages of the Proof of Elapsed Time consensus protocol

Advantages	Disadvantages
Energy-efficient consensus protocol	All participants must support Intel SGX or other TEE technologies
Good performance in permissioned blockchains without the need for message exchanges	There are known vulnerabilities of Intel SGX that can compromise consensus
Fair vote system: "One CPU, one vote"	Low performance in comparison with quorum-based protocols
More people can participate due to low cost	Limited scalability (at most hundreds of nodes)

to Bitcoin PoW throughput [25]. However, the protocol presents important scalability limitations. As highlighted by Dang et al., the probability, C, of two or more blocks being proposed at the same time is given by

$$C \approx \frac{n\delta}{T}, \tag{6}$$

where n is the number of consensus participants, δ is the network propagation delay and T is the average block time [25]. The authors use this model to prove that two or more participants can simultaneously generate certificates and propose conflicting blocks without being aware of the others due to network delays. Similar to Bitcoin, the participants must spend more time deciding between conflicting blocks as the stale block rate increases, thus leading to lower throughput. The authors demonstrated that the throughput decreases consistently as the number of nodes increases, which limits the scalability to hundreds of nodes.

5.2 Proof of Burn (PoB)

The Proof of Burn (PoB) consensus protocol is a proof-based alternative to Proof of Work and Proof of Stake in which a participant burns coins to win the right to propose a block. Iain Stewart proposes the PoB consensus in 2012 in the Bitcoin forum [42]. The probability that a participant wins the right to propose a block is proportional to the number of coins the participant burns. To burn a coin, the participant makes a burn transaction to the burn address, which is a predetermined verifiably unspendable address because it owns no associated private key. Once a participant transfers digital money to this address, the money is burnt and becomes impossible to recover.

In the PoB consensus, miners invest their money in burning coins instead of mining hardware. The main idea is that burning coins provides virtual resources that are more sustainable than physical resources that waste a lot of energy in PoW. PoB incentivizes the miners by rewarding them with transaction fees when they win the consensus round as compensation for the investment, like in Bitcoin [41]. The hash of a burn transaction is a burn hash that the consensus algorithm uses to decide the consensus leader [48]. All nodes calculate the burn hashes through Equation 7, and the participant with the lower value of burn hash becomes the consensus leader who will propose the next block [44]:

$$Burn\ hash = (Internal\ hash) \times Multiplier. \tag{7}$$

The $Internal\ hash$ and the $Multiplier$ are given by Equations 8 and 9, respectively:

$$Internal\ hash = HASH(T_h \mid t \mid B_n), \tag{8}$$

and

$$Multiplier = \frac{e^{\frac{t}{T_d}}}{Burned\ coins}, \tag{9}$$

where T_h is the hash of the transaction containing burned coins, t is the elapsed time since the transaction, B_n is the current block number, and T_d is the time after which the coin value decays. The burn transactions have a time to maturity to prevent participants from gaining instantaneous mining power. This condition also increases the consensus security by preventing that a participant creates a fork on the blockchain to invalidate the burn transaction and recover the burned coins [41]. The consensus security also relies on the initial burned coin security. If the old coin is vulnerable, then the PoB will have security issues. The burn transactions can be deleted whenever the ledger of the old cryptocurrency, used to burn coins, is vulnerable to modifications. Hence, a malicious consensus participant can recover his/her investment or even prevent the existence of other participants' burning transactions from centralizing the power on the PoB consensus protocol.

It is easier to mine and reinvest the mining reward in consensus rounds to increase the probability of proposing new blocks in the early days of the system when there is a small number of burned coins in the network. Hence, it may be difficult for new consensus participants to compete against old participants that already own many coins and can invest more in the consensus round. The network needs a mechanism to prevent the "rich get richer" situation faced by other cryptocurrencies. As a countermeasure, the value of burned coins in PoB decays exponentially as time passes to avoid centralization in the oldest consensus participants. Besides, the decay simulates the aging of the

Table 4 Main advantages and disadvantages of the Proof of Burn consensus protocol

Advantages	Disadvantages
Energy-efficient consensus protocol	Low fault-tolerance and a high probability of forks
Economically stable and value increases over time	Difficulty to scale in the number of consensus nodes
Independent of specific mining hardware	Lack of analysis on network security
Miners have high commitment to the network	Implementations are initially based on the burning of PoW coins

mining hardware in Bitcoin. The more time passes, the more outdated the hardware becomes and the same occurs with the "virtual mining" in PoB.

Slimcoin [71] is a cryptocurrency that uses PoB combined with the PoS from PPCoin [50] and the PoW from Bitcoin [63]. Also, the Counterparty cryptocurrency burn Bitcoin coins, BTC, to generate the Counterparty currency, XCP[17] [28].

Besides the use on consensus protocols for cryptocurrencies, Proof of Burn is used to convert money from one cryptocurrency to another [28, 64] and bootstrap new cryptocurrencies, providing a fair initial currency distribution between participants [41]. Also, the participants that burn money have a high commitment to the network because the burnt money is irrecoverable. Hence, due to the engagement provided, PoB can offer notarization [19] and establish identity [2].

The coins that are based on PoB are economically stable and increase their value over time since the amount of available currency decreases when a coin is burned [41]. Another advantage of PoB is the low energy consumption and the independence of specific mining hardware. Finally, the consensus participants have a high commitment to the network since the burned money is irrecoverable and the only way to recover the investment is by proposing new blocks and maintaining the network secure.

The Proof of Burn consensus protocol, however, provides low fault-tolerance [73] and is highly susceptible to forks because the participants need to verify in every received block if it contains the lowest burn hash. Thus, PoB presents difficulty in scaling the number of consensus participants and incurs high transaction latency on public networks. Current information about the Slimcoin blockchain shows that there are only 18 consensus participants and less than one transaction per minute [24]. Nevertheless, theoretically, the PoB consensus can scale to approximately 4000 transactions per second [26].

Also, the energetic efficiency of PoB is criticized because the implementations are based on burning PoW coins that waste a lot of energy. Table 4 exhibits the main strong and weak points of the PoB consensus protocol.

5.3 Proof of learning

Proof of Learning is a hybrid consensus algorithm that combines Algorand Byzantine Agreement[18] and Proof of Storage to create a distributed machine learning repository [10]. Algorand Byzantine Agreement* ($BA^\star$) [37] is a hybrid consensus protocol for asynchronous networks that combines vote-based consensus with Proof of Stake. Quorum consensus increases the throughput, while it uses PoS to prevent Sybil attacks in the voting system. The protocol randomly selects a small set of nodes to participate in the consensus steps, modifying the traditional Byzantine Agreement (BA). Proof of Learning substitutes the Proof of Stake in the $BA^\star$ protocol is for a Proof of Storage, where the storage capacity is related to machine learning models and datasets. The proposed blockchain has its coin, WekaCoin, and utilizes financial mechanisms like Bitcoin to incentivizes nodes to process transactions and maintain network health.

There are three node types in the proposed blockchain network: suppliers, trainers, and validators. The suppliers provide a machine learning problem to other nodes in the network. Also, they share a dataset related to the task, split into a training set and test set. The trainers use the training set to create machine learning models. The trainer responsible for publishing the best model selected for a task receives a reward from the supplier with a transaction fee. Finally, after the machine learning model submission, validators nodes can verify the model metrics, which depends on the task of interest, to reach a consensus on the best model submitted. Validators also are responsible for publishing new blocks and validate transactions. Each consensus round has three tasks to establish agreement: the block transactions, the task of evaluation, and the best model selected. These decisions use Algorand Byzantine Agreement*.

Since the blockchain does not support Big Data, the authors proposed a hybrid storage structure. IPFS, an off-chain distributed file system, maintains large files, like datasets and machine learning models. In the main chain remains small information that needs immutability

[17] Over 2,100 bitcoins were burned, which exceeds 109 million dollars today's price, to create XCP in January 2014.

[18] Some authors refer to Algorand's consensus protocol as Pure Proof of Stake (PPoS).

Table 5 Main advantages and disadvantages of WekaCoin and Proof of Learning consensus protocol

Advantages	Disadvantages
The data created forms a distributed machine learning repository	High energy expenditure in the training process
Works on asynchronous networks such as the Internet	There are no practical evaluations of the proposal
More eco-friendly than Proof of Work and have a high scalability in the number of consensus nodes	The latency of the network can be very high due to the multiple decisions using Byzantine Agreement* in one round and the models' verification process

guarantee like hashes, pointers to the files, and signed transactions.

To avoid trainers cheating on the machine learning contest, the authors apply the hold-out approach, removing the labels of the test set and revealing it only when the competition finishes. Nevertheless, the proposal is vulnerable to the misbehavior of nodes since they can forge their identity to execute multiple roles on the network or send multiples solutions to the same problem. The authors mitigate the problem by imposing a transaction fee and expecting that this behavior is not profitable. The other Table 5 summarizes the main advantages and disadvantages of Proof of Learning adoption.

5.4 Proof of Authority (PoA)

The Proof of Authority (PoA) consensus protocol presents a faster and energy-efficient alternative to the PoW protocol. In PoA, a set of N known and trusted nodes, called authorities, exchange messages to determine the next block of the blockchain [3]. The protocol requires a predetermined known and certified set of validators to participate in the consensus protocol, which, usually, restricts the number of authorities in the network. This characteristic makes PoA suitable for permissioned blockchains, in which every participant in the network knows each other. Well-known Ethereum client platforms implement PoA in private networks, such as Clique in Geth[19] and Aura in Parity[20]. Although PoA is mostly used in private networks, the VeChain Thor and POA cryptocurrencies adopt PoA as their main consensus protocol.

The PoA protocol is similar to PoS but instead of using money, a validator stakes his/her authority to propose a block in a consensus round. Thus, in PoA, every validator holds the same decision power regardless of his/her resources. As the validator stakes his/her authority, he/she can be voted out of the consensus by other validators/authorities if the majority of validators detects malicious intent in a failed consensus round. To achieve that, PoA assumes that $\frac{N}{2} + 1$ of the N validators are honest, composing an honest majority of validators to vote malicious participants out correctly.

The Authority Round (Aura) is a PoA protocol implementation available in the Parity Ethereum client software. Aura splits time into multiple steps in which a validator proposes the next block of the blockchain. The protocol defines each step, s, by

$$s = t_{UNIX}/\Delta t_s, \tag{10}$$

where t_{UNIX} is the UNIX time and Δt_s is the duration of a time step[21]. Therefore, Aura assumes a synchronous network with every validator synchronized within the same UNIX time t_{UNIX} [3]. A unique identification i identifies each of the N authorities in the Aura consensus protocol. In each step s, Aura calculates $l = s \pmod N$ and assigns the role of the leader to the validator N_i with identification $i = l$. The leader then proposes a block b and broadcasts b to every authority. Each authority broadcasts the received block b to the other authorities to verify if they received the same block. If a majority of the network accepts the block b, b is committed to the blockchain. Suppose the majority of authorities refuses b. In that case, a smart contract starts a voting process in which a majority decides if the leader l should be voted out of the network based on if he acted maliciously or not.

A predetermined agreement sets the authorities, and their identities are public and verifiable by any member of the network [3]. The main advantage is the authorities' easy inspection, and the main disadvantage is the centralization of authorities with no possibility of an election. Concerning performance issues, the requirement for predetermined known nodes restricts the use of the protocol to permissioned blockchains. As the protocol relies on message exchanges rather than cryptographic puzzles, the PoA throughput outperforms the throughput of the PoW consensus protocol for a low number of participants.

Table 6 shows the main advantages and disadvantages of the PoA consensus protocol. The main advantage of PoA consensus concerning performance is the low

[19] Available at https://geth.ethereum.org/. Accessed 15th March 2021.

[20] Available at https://www.parity.io/ethereum/. Accessed 15th March 2021.

[21] Available at https://openethereum.github.io/Aura. Accessed 15th March 2021.

Table 6 Main advantages and disadvantages of the Proof of Authority consensus protocol

Advantages	Disadvantages
Energy-efficient consensus protocol	Consensus is centralized in few validators (low scalability)
Fast transaction processing	Protocol relies on time synchronization to work
Power divided equally among the validators	Disclosure of the identity of validators makes authorities susceptible to attacks

transaction processing time compared to other consensus protocols [30]. That happens because Parity processes transactions at a constant rate, enforcing maximum client requests. On the other hand, the enforcement of a constant transaction rate implies lower throughput when compared to other consensus protocols for permissioned blockchains. Dinh et al. analyze the performance of consensus protocols used in permissioned blockchains [30] and verify that the transaction throughput in PoA reaches 46 transactions per second in the Parity implementation, lower than other permissioned consensus protocols, such as PBFT.

PoA consensus protocol presents two main security vulnerabilities: network synchronization and Authorities centralization. As the protocol relies on UNIX time synchronization, De Angelis et al. analyze the security and consistency of the Aura consensus protocol implementation [3], and they state there may be periods of inconsistencies caused by out-of-sync clocks of the network validators. During this period, disjoint authorities group $\mathcal{A}_1$ and $\mathcal{A}_2$ diverge regarding the current time step and, consequently, the current round leader. Therefore, if $\mathcal{A}_1$ contains $\frac{N}{2} + 1$ of the participants of the network, $\mathcal{A}_1$ owns the majority of the vote and recognizes leaders in $\mathcal{A}_2$ as malicious. That leads to every authority in $\mathcal{A}_2$ being voted out. Ekparinya et al. developed the cloning security attack, in which a malicious authority clones his/her private key and starts to act in two instances of the blockchain [32]. In a network with n odd authorities, it issues a transaction to only $(n-1)/2$ authorities so that both groups, aware of the transaction or not, believe it to be the $((n-1)/2) + 1$ majority. To perform a double-spend, the attacker explores the network topology by connecting authorities to delay the branch with the transaction. If the branch is delayed long enough, the other branch becomes the longest.

5.5 Delegated Proof of Stake (DPoS)

The Delegated Proof of Stake (DPoS) consensus protocol, proposed by Dan Larimer and used in the EOSIO platform, is an energy-efficient, scalable, and low-latency alternative to the previous proof-based consensus protocols [56]. The protocol concentrates the decisions on elected delegates to improve throughput and latency. However, the protocol preserves decentralization by ensuring the system selects delegates through a stake-based election. During the election phase, all participants choose 21 representatives by publishing votes with weights proportional to their stakes. Once the election finishes, each of the 21 elected delegates receives 6s of block producing time that is split in 12 time windows of 0.5s[22]. The system orders delegates alphabetically and every delegate receives the same amount of time regardless of how many votes it received. The delegates produce 12 blocks each per election, which totals 126 s per epoch. The consensus participants discard invalid transactions to create a valid block. Finally, the delegates check the proposed blocks and verify its validity by performing a byzantine agreement. When the block receives more than $\lceil \frac{2}{3} + 1 \rceil$ of the total consensus participants votes, which represents 15 delegates, the block is approved and inserted on the blockchain. Malicious or unresponsive delegates can be voted out of the elected quorum to guarantee network liveness and high transaction throughput [84].

The EOSIO presents a reward scheme to incentivize the elected delegates to produce blocks in every epoch. The delegates divide 0.25% of the initial amount of 0.75% of the total money proportionately to the number of votes each delegate received [43]. However, the delegates can fail or misbehave, causing a minority fork or many forks. The general rule to resolve forks is that the longest chain wins as Proof of Work consensus protocol. Honest delegates that see a valid longer chain switch from its current fork to the longer one [55]. The minority fork occurs when $\lfloor \frac{1}{3} \rfloor$ or less of the delegate try to create two different global states. Nevertheless, the minority fork will produce fewer blocks per second than the majority. The honest majority will always achieve consensus finality because they follow the longest chain rule. The same situation occurs when the minority attempts to produce an unlimited number of forks since the principal fork grows faster than the minority forks. Besides, the network can fragment, in which case no fork has a majority of the block producers.

Daniel Larimer also proposes the concept of the last irreversible block on the DPoS consensus protocol. When there is $\lceil \frac{2}{3} \rceil + 1$ of different delegate blocks after a chain

[22]The number of delegates, size of time windows, and total received time are optimized by Dan Larimer for the EOSIO implementation. The optimal values may change in different environments.

position, the block is irreversible because of the hypothesis that $\lceil \frac{2}{3} \rceil + 1$ of delegates are honest, and the longest chain rule. Thus, the only way to create a valid fork before that block is by corrupting $\lceil \frac{2}{3} \rceil$ or more of the delegates [55]. The EOSIO protocol uses incremental Merkle as a data structure to boost performance. Thus, transactions are associated with previous blocks in the network because the incremental Merkle is implemented in parallel to the multi-index table [57, 84]. The Incremental Merkle, however, is susceptible to timing attacks as transactions are not necessarily processed sequentially but rather subjectively, based on ease of processing [84].

On the one hand, the centralization in delegates presents the advantage of increasing efficiency. On the other hand, the centralization of the DPoS model presents clear security vulnerabilities, such as (i) a collusion among a few users with large stakes is enough to elect malicious delegates. (ii) The election of only a few malicious delegates allows double-spending attacks. (iii) After the election, delegates have the same power regardless of the number of votes received. It is easy for an attacker to create a denial of service in the network since the network knows the elected delegates in every epoch and the number of delegates is small. This issue could be covered by the use of cryptography sortition and increasing the number of delegates like Algorand's proposal [37]. Also, the fact that delegates do not need the same amount of votes received facilitates collusion, as attackers need to bet only on the least voted delegates, which corresponds to a small set of stakes. The EOSIO protocol authors state that there are more consensus delegates than other vote-based consensus protocols. Moreover, to avoid collusion in the election processes, the delegate quorum changes in every epoch. Table 7 summarizes the main advantages and disadvantages of the DPoS implementation.

5.6 Proof of Quality of service (PoQ)

Proof of Quality of Service (PoQ) [85] is a hybrid consensus protocol that aims to provide a scalable solution to consensus and presents many similarities with Delegated Proof of Stake. Like in DPoS, nodes in the network select delegates that adopt a simple BFT-based algorithm such as PBFT [17] to propose new blocks. However, the delegate election phase also considers quality of service criteria along with the amount of deposited stake and incentives delegate rotation. Hence, the protocol supposedly conserves the scalability and efficiency characteristics of DPoS but mitigates the tendency for centralization in a few nodes that have high resource capacity.

Nodes in PoQ are divided in groups or regions. The system assumes each region is highly synchronized and that nodes can join and leave regions at will. To perform a block proposal, the nodes in each region will select a number of possible candidates by evaluating four quality of service metrics: (i) the deposit ratio $\eta_i = \frac{m_i}{M} \in [0, 1]$, which represents the amount m_i that a candidate deposited in relation to the total of deposits M; (ii) the error rate $\beta_e = \frac{s_e}{S} \in [0, 1]$, which represents the number s_e of times the candidate failed to proposed a block over the total S rounds it was elected as a delegate before; (iii) the activity rate $\gamma_i = \frac{b_i}{B} \in [0, 1]$, which represents the number b_i of times the node was elected over the total number B of rounds since the node joined the network; and (iv) a reference factor ϕ that represents the reputation of the node in the network. Briefly, each of the four parameters account for a QoS metric: the deposit ratio indicates how much the node invested, the error rate indicates how many times the node has already failed to propose blocks when elected, the activity rate indicates how often the node is elected and the reference factor indicates how much the region trusts it in general. The parameters form a vector $\vec{v} = [\eta_i, \beta_e, \gamma_i, \phi]$ which is multiplied by a vector of weights $\vec{w} = [\alpha_1, -\alpha_2, -\alpha_3, \alpha_4]$ to obtain the final QoS value $\xi = \vec{v}.\vec{w}$. Note that high error and activity rates incur lower overall QoS values, hence incentivizing rotation and good node behavior. The nodes in the region put all candidates with enough QoS in a list and select one of them with a common random seed. Hence, all honest nodes select the same delegate. In the next phases, the delegates of each region perform PBFT consensus and broadcast the blocks at the end.

The main security issues of PoQ lie on the nomination process. Because the default values for the error rate and the activity rate are the best possible, the protocol is prone to attacks of malicious participants that constantly change their public key to appear as new candidates. This advantageous behavior may compromise the system by electing malicious delegates that can disrupt the BFT consensus process. Although the authors do not discuss this vulnerability in detail, a straightforward countermeasure would be to give

Table 7 Main advantages and disadvantages of the Delegated Proof of Stake consensus protocol

Advantages	Disadvantages
Energy-efficient consensus protocol	Low fault-tolerance
High transaction throughput	Vulnerable to denial of service attacks
Diversity of consensus participants when there is no collusion	Vulnerable to collusion among a few users with large stakes

a heavy weight the other two QoS parameters, i.e., deposit ratio and reference factor. The byzantine agreement phase of PoQ protocol also presents the same vulnerabilities as the Delegated Proof of Stake protocol, such as the possibility of denial of service attacks and collusion among delegates. Likewise, the performance of PoQ is similar to DPoS, reaching a throughput of at most a few thousand transactions per second with a few seconds of delay [85]. Table 8 presents the overall advantages and disadvantages of the protocol.

5.7 Proof of Vote (PoV)

Proof of Vote [58] is a consensus protocol based on voting mechanism proposed in 2017 that presents low-latency transaction confirmation and is suited for consortium blockchains. In PoV, a special set of nodes controls the core of the network, detains the voting rights, and delegates the task of creating a block to other nodes.

Proof of Vote presents four types of nodes: (i) commissioners, (ii) butlers, (iii) butler candidates, and (iv) ordinary users. Commissioners are institutions and enterprises that compose a committee, maintain a consortium blockchain, vote for blocks, and delegate the task of block producing to butlers. Users can only join the network as commissioners if accepted by the rest of the committee and are properly identified. Butlers are elected nodes that gather transactions from the transaction pool and pack them into a block. Commissioners vote in butler candidates and the most voted candidate nodes become butlers, as the number of butlers is limited. A network participant becomes a butler candidate by submitting an application, being assigned by one of the commissioners, or submitting a deposit. Ordinary users forward blocks and transactions but do not participate in the consensus protocols.

The Proof of Vote consensus protocol is divided in tenure cycles, composed by N_r rounds and one butler elected as block proponent per round. In each tenure cycle, butlers are assigned a number from 0 to $N_b - 1$, where N_b is the number of butlers. An elected butler b_i for a consensus round j groups transactions in a block B_j and send B_j to all commissioner nodes. The block B_j is valid if at least $\frac{N_c}{2} + 1$ signs the block header, where N_c is the number of commissioners. After receiving $\frac{N_c}{2} + 1$ signatures, the butler b_i sends the block B_j to a NTP server that provides the timestamp, signs the block header, and returns B_j to the butler. The butler, then, generates a random number R between 0 and $N_b - 1$. The butler B_R that received the number R at the beginning of the tenure cycle is elected to propose the next block B_{j+1}. The last block of a tenure cycle contains only election information about the next tenure, including the elected butlers for the next tenure and a random number to select the first block proponent in the next tenure.

As in DPoS, the centralization of delegating block proponents promotes fairness among voters and increases transaction throughput, achieving low-latency transaction processing. However, this centralization of decisions in the network in few nodes makes the protocol vulnerable to denial of service attacks, as the comissioner nodes do not change and are well-known. The protocol also relies on a trusted NTP server to provide a timestamp on each transaction, which creates a single point of failure and makes the protocol vulnerable to Byzantine behavior of the NTP server. Thus, the NTP centralized NTP server may (i) reject transactions from honest participants, (ii) halt the consensus by not signing blocks, or (iii) make it easier to perform a double-spend attack by signing out-of-order transactions. Table 9 presents the main advantages and disadvantages of the proof of vote consensus protocol.

6 DAG-based consensus: IOTA Tangle

In the Internet of Things, security and privacy can be easily compromised by attackers due to the hardware limitations of devices [62]. IOTA is a cryptocurrency that aims to provide trustful decentralized machine-to-machine (M2M) micro-payments while maintaining the security and privacy of users in resource-restricted environments. IOTA takes inspiration from peer-to-peer applications to eliminate the separation between clients and miners. In IOTA, a user that wishes to issue a new transaction must contribute to the system by validating previous transactions. Hence, users are simultaneously clients and miners. Several researchers [72, 80, 83] regard IOTA as the next generation of distributed ledger technologies because IOTA claims to provide (i) high throughput and scalability because the more users join the

Table 8 Main advantages and disadvantages of the Proof of Quality of Service consensus protocol

Advantages	Disadvantages
Low latency transaction confirmation	Vulnerable to key-changing attacks
Allows participants to select delegates in a fine-grained manner	Low fault tolerance
Mitigates centralization in rich nodes	Vulnerable to denial of service attacks

Table 9 Main advantages and disadvantages of the Proof of Vote consensus protocol

Advantages	Disadvantages
Low latency transaction confirmation	Use restricted to consortium blockchains
Energy-efficient consensus protocol	Decision power is highly centralized in few nodes
High transaction throughput	Single point of failure of NTP server

network, the more mining power the network achieves; (ii) tax-free transactions, because the transaction issuer works for its transaction instead of sending it to a miner; and (iii) efficient micro-payment channels, which IoT devices can use to trade data automatically and with low latency. Table 10 highlights the main advantages and disadvantages of the IOTA implementation.

The IOTA consensus protocol, formalized by Popov in 2017 [72], uses an innovative data structure called the Tangle. The Tangle is a distributed ledger structure that organizes transactions in a directed acyclic graph (DAG) rather than a blockchain. The DAG structure allows participants to publish transactions concurrently and asynchronously because it allows two transactions to point to the same previous transaction, which would be equivalent to a fork in the blockchain. As a consequence, a notable feature of the IOTA consensus compared to the blockchain consensus is that the system considers that different participants in the network may have different views on transactions. This characteristic contrasts sharply with a global view of the blockchain, in which all transactions are identical in any participant. The main disadvantage of the DAG structure is that the tie-breaking mechanism in IOTA must consider all the possible different views and find one which it considers to be correct.

Figure 4 illustrates an example of a Tangle data structure. Each vertex of the graph represents a transaction, and each edge represents the result of validating a transaction. The user must confirm at least two unconfirmed transactions to add his/her transaction to the Tangle[23]. Unconfirmed transactions are called "tips" of the Tangle. To add a transaction to the ledger, the user must include the IDs of two tips and add the source and destination addresses to the new transaction. Then, he/she solves a challenge based on Proof of Work and disseminates the result on the network. The Proof-of-Work challenge in IOTA is way easier than in Bitcoin as it serves only as a mechanism to control transaction spamming. Adding a transaction creates two new directed edges in the graph that confirm the previous transactions, and thus the structure functions as a generalized version of the hash sequence of the blockchain. IOTA does not reward transaction validators because the incentive is to add the transaction itself. All currency in the system derives from the first transaction.

[23] In the current implementation of IOTA, the number of confirmations required to add a transaction to the network is exactly two.

If there are conflicting tips with the same source address, each user needs to decide which one to approve with their new transaction. The main mechanism for choosing a tip is to perform multiple rounds of the default tip selection algorithm and verify which of the two conflicting tips is most likely to be chosen. For example, if the algorithm selects one of the tips 95 times in 100 executions, we would say the system has 95% confidence that the tip is correct. IOTA currently uses a tip selection algorithm based on random walks and Markov Chain Monte Carlo (MCMC) methods that prioritize transactions with greater cumulative weight. Briefly, the algorithm introduces a particle at some past transaction and randomly walks through the graph with transition probabilities proportional to the cumulative weight of each transaction. The algorithm stops when it reaches a tip. Because the transition probability is proportional to the cumulative weight, the particle is likely to reach the tip that points to the heaviest path, and thus, the system converges to select it as the correct tip. Selecting the heaviest path in IOTA is similar to selecting the longest chain in Bitcoin, as it privileges the path with more transactions and associated energy expenditure.

Despite innovating with the Tangle structure, the security of the IOTA protocol remains an open challenge. Popov, a co-founder of IOTA, already predicts the Tangle could be explored to create multiple attacks [72]. For instance, an attacker can create an offline parasite chain that overtakes the main chain and point it to a past transaction, creating a fork [12]. The main problem, however, is that IOTA depends on user hash power to validate previous transactions and to improve the security of the system. This problem causes the need for the Coordinator, a centralized validator controlled by the IOTA Foundation that issues null transactions only to validate previous transactions. Because the hashing power on the network is highly dynamic, the hashing power of an attacker can be higher than the honest users. The lack of a financial reward also contributes to the insecurity in the system because users are only incentivized to validate older transactions if they intend to issue new ones.

7 Comparing consensus protocols

Proof-based protocols present possibilities for forks because any participant can propose a block and there is a probability of simultaneously proposing blocks. Regardless of

Table 10 Main advantages and disadvantages of the IOTA consensus protocol

Advantages	Disadvantages
Theoretically unlimited scalability	Complex and costly tiebreaking mechanism
Allows offline and concurrent transaction processing	Highly vulnerable to double spending in offline payments
Energy-efficient consensus protocol	Depends on user engagement to be secure

the specific protocol implementation, malicious participants can exploit the temporary inconsistencies in probabilistic consensus to launch attacks that are not possible in deterministic protocols. However, each protocol presents specific security and performance issues that stem from their implementation and assumptions. Table 11 presents a comparison between all the analyzed protocols concerning throughput, latency, scalability, and main vulnerabilities.

Proof of Work is the first probabilistic consensus protocol to be successfully applied to a public network and Bitcoin's PoW never suffered a confirmed attack in over a decade of operation. Its energy cost, however, is prohibitive. Rewarded mining based on costly mathematical challenges leads to the centralization of powerful miners who can afford high-performance hardware. Moreover, PoW presents a low throughput, a high latency, and a centralization tendency.

Proof of Stake is an energy-efficiency, high throughput, and low latency alternative to Proof of Work, but it introduces new vulnerabilities such as the "nothing at stake" problem and "long range" attacks. Proof of Stake also requires rewards to incentivize the "bets," and the centralization tendency should be a problem. The block and transaction creation rates are high because there is no time spent to solve a challenge. Therefore, Proof of Stake presents a high number of forks, which increases the risk of attacks.

The other proof-based alternatives present protocol-specific security issues and performance. The main vulnerability of Proof of Elapsed Time lies in the security of the Trusted Execution Environment (TEE) that draws a random timer for each participant. Several authors show Intel SGX is prone to attacks and that an adversary can compromise the PoET consensus with less than 50% of the CPU in the network [18, 77]. The transaction throughput of PoET is high in comparison with PoW and PoS, but the high probability of forks limits its scalability to at most a few hundred nodes [25].

Proof of Burn (PoB) is a consensus protocol in which a participant burns coins to win the right to propose a block. The money is burnt and becomes impossible to recover. The pros are energy efficiency and economic stability. On the other hand, it presents a high probability of forks, low fault tolerance, and does not scale well.

Proof of Learning is more eco-friendly than Proof of Work and creates a distributed machine learning repository. The consensus protocol also works on asynchronous networks such as the Internet. However, the latency of the network can be very high due to the multiple decisions using Byzantine Agreement* in one round and the time-consuming models' verification process. Besides, the proposal can have problems with high energy expenditure in the model training process, and there are no practical evaluations.

The Proof of Authority (PoA) consensus protocol requires a predetermined known and certified set of validators, called authorities, to participate in the consensus. This protocol is well suited to permissioned blockchain because the authorities exchange messages to determine the next

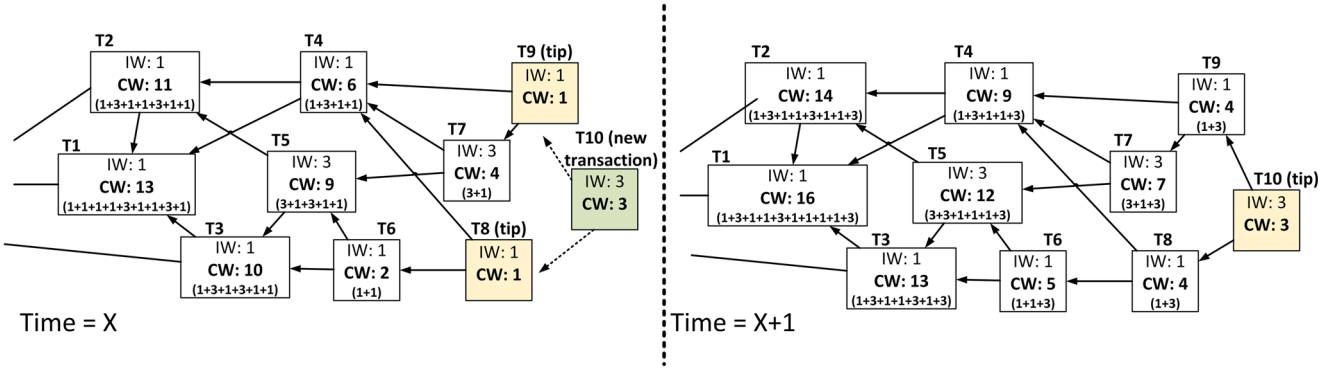

Fig. 4 The addition of a new transaction, T10, into the Tangle data structure. Each transaction has an individual weight (IW) and a cumulative weight (CW), which corresponds to the sum of the individual weights of all transactions that have approved it directly or indirectly. After selecting and validating two tips, the new transaction becomes a tip and its individual weight propagates to the cumulative weights of previous transactions

Table 11 Comparison of Proof-based consensus protocols

Consensus protocol	Platform	Maximum throughput	Latency	Scalability (#validators)	Known vulnerabilities
Proof of Work (PoW)	Bitcoin	≈ 7 tx/s	10 min.	Thousands	Double-spending and Finney attacks [34, 47]; 51%, selfish mining, block discarding, and bribery attacks [5, 9, 33]; eclipse and network attacks [39, 45].
	Ethereum	≈ 15 tx/s	15 s	Thousands	
Proof of Stake (PoS)	Cardano	≈ 250 tx/s	20 s	Hundreds	The "nothing at stake" problem [11] and long range attacks [27].
Proof of Elapsed Time (PoET)	Hyperledger Sawtooth	≈ 1000 tx/s	Variable (default: 20 s)	Hundreds	Vulnerabilities on Intel SGX [77] and consensus subversion [18].
Proof of Burn (PoB)	Slimcoin	≈ 4000 tx/s	A few seconds (theoretical) A few minutes (mainnet)	Dozens	The oldcoin vulnerabilities [41].
Proof of Learning (PoL)	WekaCoin	≈ 1000 tx/s‡	≈ 5 s‡	Thousands‡	Sybil attacks [10].
Proof of Authority (PoA)	Aura, VeChain Thor, POA	≈ 45 tx/s (Aura)	Variable (default: 5 s)	Dozens	Cloning attacks [32] and consistency issues [3].
Delegated Proof of Stake (DPoS)	EOSIO	≈ 4000 tx/s	0.5 s	Dozens	Collusion among users with large stakes to elect malicious delegates, denial-of service attacks onto delegates, and timing attacks [84].

Table 11 (continued)

Consensus protocol	Platform	Maximum throughput	Latency	Scalability (#validators)	Known vulnerabilities
Proof of Quality of Service (PoQ)	-	≈ 1000 tx/s	A few seconds (theoretical)	Dozens	Collusion among delegates, denial-of service attacks onto delegates, and key-changing vulnerabilities during delegate nomination [84, 85].
Proof of Vote (PoV)	-	-	15 s (theoretical)	Hundreds	Denial of service attacks and centralized trust in NTP server.
IOTA Tangle	IOTA	≈ 80000 tx/s (theoretical) ≈ 30 tx/s (mainnet)	A few seconds (theoretical) A few minutes (mainnet)	Unlimited (theoretical) Thousands (mainnet)	Parasite chains [72], double-spending on offline payments [12], and user-dependent transaction validation.

‡We estimate these values based on the Algorand consensus protocol [37] since Proof of Learning utilizes the $BA^{\star}$ to reach consensus among the participants.

block of the blockchain, which restricts the number of authorities in the network. The protocol relies on network synchronization, which is a great security drawback because the participants may suffer denial of service attacks.

Delegated Proof of Stake combines the scalability of proof-based consensus with the determinism of vote-based protocols. The delegated model, however, is more centralized than the Proof of Work and Proof of Stake, which improves its throughput performance to thousands of transactions per second. On the other hand, it is more sensitive to collusion between malicious participants. Proof of Quality of service (PoQ) adopts a similar concept but defines a more complex manner of selecting delegates to mitigate centralization on rich nodes. Hence, it achieves similar performance values and suffers from similar vulnerabilities as DPoS.

Proof of Vote presents a low-latency transaction confirmation in a more energy-efficient protocol than proof of work. However, PoV is restricted to consortium blockchains and is highly centralized in few nodes, which creates single point of failure and becomes vulnerable to malicious behavior.

The IOTA protocol presents an innovative data structure that aims to replace the blockchain as a distributed ledger technology. Nevertheless, IOTA currently depends on a centralized authority to validate transactions and it introduces several vulnerabilities that remain unexplored.

8 Related works

Blockchain plays a paradigm shift in today's society, with Bitcoin and Ethereum cryptocurrencies leading the market and being the precursors to several other cryptocurrencies. For this reason, the consensus protocols for the blockchains attract the attention of several research groups [16, 31, 68, 69]. The consensus vulnerabilities associated with each consensus protocol and their respective countermeasures are not widely explored.

Gervais et al. propose a framework for security analysis in on Proof-of-Work-based blockchains [36]. Xiao et al. model the security of Proof of Work according to the participants' connectivity concerning selfish mining attacks and the collusion between participants [82]. Conti et al. analyze various components and their respective vulnerabilities in the Bitcoin blockchain [20]. Li et al. analyze the security of consensus based on proof of stake [59]. Li et al. Summarize the main security vulnerabilities in blockchain systems [60]. Besides, the authors present real cases of attacks on the two largest market capital cryptocurrencies: Bitcoin and Ethereum. The works, however, do not extend the analysis and proposals across different probabilistic protocols.

Xiao et al. [83] and Joshi et al. [46] bring together different deterministic and probabilistic consensus protocols for blockchain. The papers analyze the security of different probabilistic and deterministic blockchains. Zhang et al. divide the blockchain architecture into six layers and analyze the security of each one [86]. However, the consensus layer is not widely analyzed.

This paper, different from previous works, summarizes the leading aspects of the most widely used proof-based consensus protocols, focusing on the main performance characteristics and the crucial vulnerabilities and attacks of each protocol, with their respective countermeasures. Furthermore, we describe the IOTA protocol that proposes Tangle, a distributed ledge structure organized as a directed acyclic graph, to serve in IoT environments.

9 Conclusion

This paper analyzed the security and performance of several proof-based consensus protocols that aim to substitute Proof of Work as the main probabilistic consensus protocol. We conclude, however, that despite being the protocol with the largest number of known vulnerabilities, it is a fact that Bitcoin's security is exceptional in practice, as there has been no successful attack on the protocol in more than 11 years of existence. Any other consensus that will replace it must prove that it presents this robustness to attacks. We also observe the protocols exhibit a trade-off between performance, i.e., throughput and latency, and scalability. The two protocols that seem to achieve the best trade-off potential are (i) Delegated Proof of Stake (DPoS), which centralizes consensus in delegates to improve performance, but still allows thousands of users to choose the delegates in a decentralized manner; and (ii) IOTA, which presents a new consensus concept in which the more users participate, the more throughput and scalability the system provides. Both DPoS and IOTA, however, introduce new vulnerabilities that remain unexplored.

In future works, we intend to study hybrid protocols. We expect that the best consensus proposal combines deterministic consensus with probabilistic consensus to achieve the best result in the observed performance-scalability trade-off.

Funding This work was financed by CNPq, CAPES, FAPERJ, and FAPESP (2018/23292-0, 15/24485-9, 14/50937-1).

References

1. Alvarenga ID, Rebello GAF, Duarte OCMB (2018) Securing configuration management and migration of virtual network functions using blockchain. In: IEEE/IFIP NOMS 2018, Pp. 1–9
2. Taaki A et al (2014) OpenBazaar. https://openbazaar.org/
3. Angelis SD, Aniello L, Baldoni R, Lombardi F, Margheri A, Sassone V (2018) PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain. In: Italian Conference on Cyber Security (06/02/18). https://eprints.soton.ac.uk/415083/
4. Attiya H, Bar-Noy A, Dolev D (1995) Sharing memory robustly in message-passing systems. Journal of the ACM (JACM) 42(1):124–142
5. Bahack L (2013) Theoretical Bitcoin attacks with less than half of the computational power (draft). arXiv preprint arXiv:1312.7013
6. Bano S et al (2017) Consensus in the age of blockchains. CoRR abs/1711.03936. 1711.03936
7. Bessani A, Sousa J, Alchieri EEP (2014) State machine replication for the masses with BFT-SMART. In: 2014 44Th annual IEEE/IFIP international conference on dependable systems and networks, pp 355–362. https://doi.org/10.1109/DSN.2014.43
8. BitcoinWiki (2019) Bitcoin scalability. https://en.bitcoin.it/wiki/Scalability
9. Bonneau J, Felten EW, Goldfeder S, Kroll JA, Narayanan A (2016) Why buy when you can rent? In: ICFCDS, pp 19–26. Springer
10. Bravo-Marquez F, Reeves S, Ugarte M (2019) Proof-of-learning: a blockchain consensus mechanism based on machine learning competitions. In: International conference on decentralized applications and infrastructures (DAPPCON), pp 119–124. IEEE
11. Brown-Cohen J, Narayanan A, Psomas A, Weinberg SM (2019) Formal barriers to longest-chain proof-of-stake protocols. In: Proceedings of the 2019 ACM Conference on Economics and Computation, pp 459–473
12. Bu G, Gürcan Ö, Potop-Butucaru M (2019) G-IOTA: Fair and confidence aware tangle. In: IEEE INFOCOM WKSHPS, pp 644–649
13. Buterin V (2019) Proof-of-Stake FAQ. https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ
14. Camilo GF, Rebello GAF, de Souza LAC, Duarte OCMB (2020) A secure personal-data trading system based on blockchain, trust, and reputation. In: 2020 IEEE International conference on blockchain (blockchain), pp 379–384. https://doi.org/10.1109/Blockchain50366.2020.00055
15. Camilo GF, Rebello GAF, de Souza LAC, Duarte OCMB (2020) Autavailchain: Automatic and secure data availability through blockchain. In: IEEE GLOBECOM, pp 1–6
16. Carrara GR, Burle LM, Medeiros DS, de Albuquerque CVN, Mattos DM (2020) Consistency, availability, and partition tolerance in blockchain: a survey on the consensus mechanism over peer-to-peer networking. Ann Telecommun, pp 1–12
17. Castro M, Liskov B (1999) Practical byzantine fault tolerance. In: Proceedings of the Third Symposium on Operating Systems Design and Implementation, OSDI '99. USENIX Association, USA, pp 173–186
18. Chen L, Xu L, Shah N, Gao Z, Lu Y, Shi W (2017) On security analysis of proof-of-elapsed-time (poET). In: International symposium on stabilization, safety, and security, pp 282–297. Springer
19. Clark J, Essex A (2012) Commitcoin: Carbon dating commitments with bitcoin. In: International conference on financial cryptography and data security, pp 390–398. Springer
20. Conti M, Kumar ES, Lal C, Ruj S (2018) A survey on security and privacy issues of Bitcoin. IEEE Communications Surveys & Tutorials 20(4):3416–3452
21. Costa LHMK, Fdida S, Duarte OCMB (2006) Incremental service deployment using the hop-by-hop multicast routing protocol. IEEE/ACM Trans Networking 14(3):543–556
22. Costan V, Devadas S (2016) Intel SGX explained. IACR Cryptol. ePrint Arch. 2016(86):1–118
23. Coulouris G, Dollimore J, Kindberg T, Blair G (2011) Distributed systems: Concepts and design. 5th. USA: Addison-Wesley Publishing Company 662:665–668

24. cryptoID (2021) Slimcoin Blockchain Explorer. https://chainz.cryptoid.info/slm/
25. Dang H, Dinh A, Chang EC, Ooi BC (2018) Chain of trust:, Can trusted hardware help scaling blockchains? arXiv preprint arXiv:1804.00399
26. Decentralized Web (2017) Slimcoin: First Proof of Burn currency. https://bitcointalk.org/index.php?topic=1141676.1915;wap2
27. Deirmentzoglou E, Papakyriakopoulos G, Patsakis C (2019) A survey on long-range attacks for proof of stake protocols. IEEE Access 7:28712–28725
28. Dermody R., Krellenstein A., Slama O. (2014) Counterparty. https://counterparty.io/
29. Digiconomist (2020) Bitcoin Energy Consumption Index. https://digiconomist.net/bitcoin-energy-consumption/
30. Dinh TTA, Wang J, Chen G, Liu R, Ooi BC, Tan KL (2017) BLOCKBENCH: A Framework for Analyzing Private Blockchains. In: Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD '17, pp 1085–1100. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3035918.3064033
31. Dong Y, Boutaba R (2019) Elasticoin: Low-volatility cryptocurrency with proofs of sequential work. In: 2019 IEEE International conference on blockchain and cryptocurrency (ICBC), pp 205–209. IEEE
32. Ekparinya P, Gramoli V, Jourjon G (2019) The attack of the clones against proof-of-authority. arXiv preprint arXiv:1902.10244
33. Eyal I, Sirer EG (2018) Majority is Not Enough: Bitcoin Mining is Vulnerable. Commun. ACM 61(7):95–102. https://doi.org/10.1145/3212998
34. Finney H (2011) Best practice for fast transaction acceptance-how high is the risk?. https://bitcointalk.org/index.php?topic=3441.msg48384#msg48384
35. Fischer MJ, Lynch NA, Paterson MS (1985) Impossibility of distributed consensus with one faulty process. JACM 32(2):374–382
36. Gervais A, Karame GO, Wüst K, Glykantzis V, Ritzdorf H, Capkun S (2016) On the security and performance of proof of work blockchains. In: ACM SIGSAC, pp 3–16
37. Gilad Y, Hemo R, Micali S, Vlachos G, Zeldovich N (2017) Algorand: Scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th Symposium on Operating Systems Principles, pp 51–68
38. Hadzilacos V, Toueg S (1994) A Modular Approach to the Specification and Implementation of Fault-Tolerant Broadcasts. Tech. rep., Department of Computer Science, Cornell University, New York - USA
39. Heilman E, Kendler A, Zohar A, Goldberg S (2015) Eclipse attacks on bitcoin's peer-to-peer network. In: USENIX Security'15, pp 129–144
40. Hoang VH, Lehtihet E, Ghamri-Doudane Y (2020) Privacy-preserving blockchain-based data sharing platform for decentralized storage systems. In: 2020 IFIP Networking conference (networking), pp 280–288. IEEE
41. Stewart I (2012) Proof of Burn. https://en.bitcoin.it/wiki/Proof_of_burn
42. Stewart I (2012) Proof of burn - a potential alternative to proof of work and proof of stake. https://bitcointalk.org/index.php?topic=131139.msg1404195
43. InfStones (2018) The Economics of EOS Blockchain. https://medium.com/infstones/the-economics-of-eos-blockchain-621d5d1e45b8
44. Ismail L, Materwala H (2019) A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions. Symmetry 11(10):1198
45. Johnson B, Laszka A, Grossklags J, Vasek M, Moore T (2014) Game-theoretic analysis of DDos attacks against Bitcoin mining pools. In: ICFCDS, pp 72–86
46. Joshi AP, Han M, Wang Y (2018) A survey on security and privacy issues of blockchain technology. MFC 1(2):121
47. Karame GO, Androulaki E, Capkun S (2012) Double-spending fast payments in bitcoin. In: ACM CCS 2012, Pp. 906–917
48. Karantias K, Kiayias A, Zindros D (2020) Proof-of-burn. In: International conference on financial cryptography and data security, pp 523–540. Springer
49. Kiayias A, Russell A, David B, Oliynykov R (2017) Ouroboros: a provably secure proof-of-stake blockchain protocol. In: CRYPTO, pp 357–388
50. King S, Nadal S (2012) PPCOin: Peer-to-peer crypto-currency with proof-of-stake. self-published paper, August 19
51. Kwon J (2014) Tendermint: Consensus without mining. Draft v. 0.6 fall 1(11)
52. Kwon J, Buchman E (2019) Cosmos whitepaper
53. Lamport L (1998) The part-time parliament. ACM Transactions Computer Systems 16(2):133–169
54. Lamport L, Shostak R, Pease M (1982) The Byzantine Generals Problem. ACM TOPLAS 4(3):382–401. https://doi.org/10.1145/357172.357176
55. Larimer D (2017) DPoS Consensus Algorithm - The Missing White Paper. https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper
56. Larimer D (2017) EOS.IO White Paper. https://developers.eos.io/-welcome/latest/protocol/consensus_protocol
57. Larimer D et al (2018) EOS.IO Technical White Paper v2. https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md
58. Li K, Li H, Hou H, Li K, Chen Y (2017) Proof of vote: a high-performance consensus protocol based on vote mechanism amp; consortium blockchain. In: 2017 IEEE 19Th international conference on high performance computing and communications; IEEE 15th international conference on smart city; IEEE 3rd international conference on data science and systems (HPCC/smartcity/DSS), pp 466–473. https://doi.org/10.1109/HPCC-SmartCity-DSS.2017.61
59. Li W, Andreina S, Bohli JM, Karame G (2017) Securing proof-of-stake blockchain protocols. In: DPM/CBT, pp 297–315. Springer
60. Li X, Jiang P, Chen T, Luo X, Wen Q (2020) A survey on the security of blockchain systems. FGCS 107:841–853
61. Lunardi RC, Michelin RA, Neu CV, Zorzo AF (2018) Distributed access control on IoT ledger-based architecture. In: NOMS 2018-2018 IEEE/IFIP Network operations and management symposium, pp 1–7. IEEE
62. Mossé D, Pötter H, Lee S (2020) Maintaining privacy and utility in IoT system analytics. In: 2020 Second IEEE international conference on trust, privacy and security in intelligent systems and applications (TPS-ISA), pp 157–164. IEEE
63. Nakamoto S (2008) Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf
64. Natoli C, Yu J, Gramoli V, Esteves-Verissimo P (2019) Deconstructing blockchains: A comprehensive survey on consensus, membership and structure. arXiv preprint arXiv:1908.08316
65. Nguyen DC, Pathirana PN, Ding M, Seneviratne A (2020) Blockchain and edge computing for decentralized emrs sharing in federated healthcare. In: GLOBECOM 2020-2020 IEEE Global communications conference, pp 1–6. IEEE
66. Nguyen DC, Pathirana PN, Ding M, Seneviratne A (2020) Blockchain for 5g and beyond networks: a state of the art survey. Journal of Network and Computer Applications p 102693
67. NXT community (2014) Nxt whitepaper. https://nxtwiki.org/wiki/Whitepaper:Nxt

68. Oliveira MT et al (2019) Towards a performance evaluation of private blockchain frameworks using a realistic workload. In: ICIN, pp 180–187. IEEE
69. de Oliveira MT et al (2020) Blockchain reputation-based consensus: a scalable and resilient mechanism for distributed mistrusting applications. Computer Networks p 107367
70. Olson K, Bowman M, Mitchell J, Amundson S, Middleton D, Montgomery C (2018) Sawtooth: an introduction linux foundation
71. P4Titan (2014) Slimcoin a peer-to-peer crypto-currency with proof-of-burn "Mining without Powerful Hardware". https://github.com/slimcoin-project/slimcoin-project.github.io/raw/master/whitepaperSLM.pdf
72. Popov S (2017) The Tangle. cit. on p. 131. http://www.descryptions.com/Iota.pdf
73. Praveen G, Anand M, Singh PK, Ranjan P (2020) An overview of blockchain consensus and vulnerability. In: International conference on information and communication technology for intelligent systems, pp 459–468. Springer
74. Rebello GAF, Alvarenga ID, Sanz IJ, Duarte OCM (2019) BSEc-NFVO: A blockchain-based security for network function virtualization orchestration. In: IEEE ICC, pp 1–6
75. Rebello GAF, Camilo GF, Guimarães LCB, de Souza LAC, Duarte OCMB (2020) On the security and performance of proof-based consensus protocols. In: 2020 4Th conference on cloud and internet of things (CIot), pp 67–74. https://doi.org/10.1109/CIoT50422.2020.9244295
76. Rebello GAF et al (2019) Providing a sliced, secure, and isolated software infrastructure of virtual functions through blockchain technology. In: IEEE HPSR, pp 1–6
77. van Schaik S, Kwong A, Genkin D, Yarom Y (2020) SGAxe: How SGX fails in practice
78. Schwartz D, Youngs N, Britto A (2014) The ripple protocol consensus algorithm. Ripple Labs Inc White Paper. https://ripple.com/files/ripple_consensus_whitepaper.pdf
79. de Souza LAC, Rebello GAF, Camilo GF, Guimarães LC, Duarte OCM (2020) DFEdforest: decentralized federated forest. In: 2020 IEEE International conference on blockchain (blockchain), pp 90–97. IEEE
80. Wang W et al (2018) A survey on consensus mechanisms and mining management in blockchain networks. CoRR abs/1805.02707. 1805.02707
81. Wood G (2014) Ethereum: A secure decentralised generalised transaction ledger. http://bitcoinaffiliatelist.com/wp-content/uploads/ethereum.pdf
82. Xiao Y, Zhang N, Lou W, Hou YT (2020) Modeling the impact of network connectivity on consensus security of proof-of-work blockchain. arXiv preprint arXiv:2002.08912
83. Xiao Y, Zhang N, Lou W, Hou YT (2020) A survey of distributed consensus protocols for blockchain networks. IEEE Communications Surveys & Tutorials 22(2):1432–1465
84. Xu B, Luthra D, Cole Z, Blakely N (2018) EOS: An architectural, performance, and economic analysis. Retrieved June 11, 2019
85. Yu B, Liu J, Nepal S, Yu J, Rimba P (2019) Proof-of-qos: Qos based blockchain consensus protocol. Computers & Security 87:101580
86. Zhang P, Zhou M (2020) Security and trust in blockchains: Architecture, key technologies, and open issues. IEEE TCSS 7(3):790–801