On the Security and Performance of Proof-based Consensus Protocols

Gabriel Antonio F. Rebello, Gustavo F. Camilo, Lucas C. B. Guimarães,
Lucas Airam C. de Souza, Otto Carlos M. B. Duarte

Universidade Federal do Rio de Janeiro - GTA/COPPE/UFRJ

***Abstract*—Blockchain is a disruptive technology that will revolutionize the Internet and our way of living, working, and trading. Despite the innovation, current blockchain-based public systems such as Bitcoin and Ethereum present significant security and performance limitations and consume excessive energy due to their proof-of-work consensus protocols. This paper presents and compares the main proof-based alternatives to proof-of-work, 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 and proof of stake, as well as their countermeasures. We also analyze the IOTA consensus protocol, a DAG-based platform suited for the IoT environment.**[1]

I. Introduction

The consensus problem in distributed systems with asynchronous networks is a known problem that researchers have been studying for over 40 years. In 2008, however, Satoshi Nakamoto[2] revolutionized the distributed consensus area by proposing a new consensus model based on proof of work (PoW) [1]. In this new consensus model, a participant that proposes a block, called a miner[3], must provide proof that it can lead the consensus by spending resources to solve a computationally costly mathematical challenge independently. The winner of the challenge is well rewarded to encourage broad competition. There is the possibility that multiple miners solve the challenge simultaneously, creating a fork in the blockchain and, consequently, an inconsistent state in the system. Unlike the consensus protocols studied so far, proof of work uses the concept of probabilistic consensus. Nakamoto's consensus introduces the probabilistic solution of maintaining the longest branch of the fork, which corresponds to the greatest number of solved challenges and thus the greatest energy expense. Proof of work does not require exchanging messages or knowing the identities of participants to obtain consensus, which provides decentralization, anonymity, and scalability at an unprecedented level in distributed systems. In Nakamoto's proposal, any person or organization can become a miner anonymously, and thousands of participants can participate in the consensus simultaneously using the Internet as a communication system. Due to its characteristics, researchers create systems that use the blockchain to provide security in several distributed applications [2], [3], [4], [5], [6], [7]. The proof-of-work protocol presents a low performance in comparison with the performance of centralized applications and incurs enormous energy expenditure. In response to the performance limitations of proof of work, several alternatives feature new proof-based protocols to replace the Bitcoin protocol. Nevertheless, the probabilistic nature of proof-based protocols, whether proof of work or alternative protocols, remains the primary source of vulnerabilities in the protocol. Non-determinism of the consensus allows a malicious agent to exploit the forks in the system to 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 carry out attacks against the network or consensus participants.

This paper presents and categorizes the main proof-based consensus protocols, exposing the attacks and security vulnerabilities of each protocol. Proof-based protocols use, like Bitcoin, the probabilistic consensus model that works on asynchronous communication systems such as the Internet, and serve public applications in which any user can participate in the consensus process. The paper focuses on proof of work (PoW) and proof of stake (PoS), the most popular alternative proof-based protocol in cryptocurrencies and public blockchain platforms. The paper compares the leading cryptocurrencies and platforms that use probabilistic protocols, such as Bitcoin, Ethereum, Cardano, EOSIO, and Hyperledger Sawtooth. We also analyze the security of the IOTA cryptocurrency, which proposes an innovative data structure suited for micro-payments in an Internet of Things environment.

The remainder of the paper is organized as follows. Section II introduces concepts of distributed systems and the classification of deterministic and probabilistic consensus. Section III details the proof-of-work consensus and analyzes possible attacks on this mechanism. Section IV describes proof of stake, the main alternative to proof of work, outlining its main security challenges. Section V presents and analyzes other alternative proof-based consensus protocols. Section VI presents and analyzes IOTA, a cryptocurrency that uses DAG-based consensus. Section VII presents works related to this

[1]This paper was funded by CNPq, CAPES, FAPERJ and FAPESP (18/23292-0, 2015/24514-9, 2015/24485-9 2014/50937-1).

[2]Satoshi Nakamoto is a pseudonym used by the creator or creators of the Bitcoin virtual currency. Its real identity is unknown.

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

paper. Section VIII concludes the paper by comparing the security vulnerabilities each consensus protocol presents.

II. The Probabilistic Consensus

The consensus protocol[4] is the distributed algorithm that ensures that the system evolves, adding new blocks correctly. Figure 1 shows the structure of the blockchain. To achieve consensus, however, the protocol must deal with a possible network or participant failures. A consensus protocol can tolerate crash faults or Byzantine faults. A crash-fault participant does not respond and does not perform new operations during a consensus round. In the Byzantine failure model, the failing participant can be a malicious agent that exhibits arbitrary behavior that deviates from the protocol and takes any action. The malicious agent might issue correct, incorrect, or contradictory replies, in addition to not replying. The Byzantine failure model best captures the behavior of participants in public blockchains, such as Bitcoin and Ethereum, in which system users can participate in the consensus anonymously, without permission, and act in a manner malicious.

The main objective of consensus protocols is to provide the safety and liveness properties to the system. The protocol guarantees liveness when there is a certainty that the consensus rounds always finish and, consequently, the system always adds new blocks. The safety property ensures that the added blocks are identical for all honest participants and that an honest participant proposed the block at the start of the consensus round. To ensure that the system works correctly and with fault tolerance, a system must build a consensus protocol that provides both safety and liveness properties.

One of the main consensus challenges in distributed systems is the result of the impossibility of guaranteeing consensus, known as the FLP impossibility in honor of the authors Fischer, Lynch, and Patterson that formulated the theorem. The result proves that the consensus has no deterministic solution even in the presence of a single crash failure in a system that operates over an asynchronous network like the Internet [8]. For decades, several consensus proposals have circumvented the FLP result by assuming synchronous, partially synchronous, or possibly synchronous communication systems, which provide different levels of guarantee of message delivery during a consensus round. Thus, consensus protocols focused on the safety property and trusted that the communication system would provide the liveness property by ensuring the delivery of messages. Nevertheless, these 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 [9]. Since Nakamoto, there are two alternatives to circumvent the FLP result: ensuring safety, as the previous protocols did, or ensuring liveness. Thus, two families of blockchain protocols appear. Protocols inspired by the classic deterministic consensus, such as Practical Byzantine Fault Tolerance (PBFT), BFT-SMaRt, Tendermint, and Ripple, favor safety over liveness, creating protocols that do not have forks. On the other hand, they can block if the communication system behaves asynchronously. Probabilistic consensus protocols, such as proof of work and proof of stake, favor liveness over safety, guaranteeing its functioning on the Internet but being susceptible to blockchain forks. As a consequence, we classify the consensus protocols as deterministic or probabilistic, respectively. The most common consensus protocols in public systems are those based on proof algorithms, in which a block proponent must provide proof that he/she can lead the consensus [1], [10]. Proof-based algorithms provide probabilistic consensus and follow similar models to Nakamoto's proof of work. Probabilistic consensus has the main advantage of scalability since it is not necessary to know all the participants in the network to reach consensus. Therefore, this type of consensus is better suited to public blockchains with many participants. In probabilistic consensus, two or more participants can propose correct blocks simultaneously, causing a fork in the blockchain. Selecting the branch of the fork that has the longest chain is the tiebreaker rule in Bitcoin and the best known in blockchains.

[4]Consensus is the process by which a group of independent participants reaches the same collective decision to accept or refuse the addition of a new block to the blockchain.

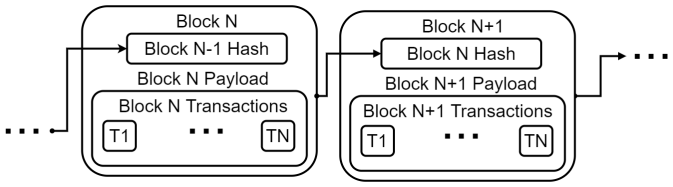

Fig. 1: Blockchain simplified structure.

III. The Proof of Work - PoW

Proof of Work (PoW) is a consensus mechanism proposed by Satoshi Nakamoto, used by the two top cryptocurrencies in market value: Bitcoin [1] and Ethereum. The proof of work gives the right to a participant to propose a new block in the blockchain. Proof of work is a computationally costly cryptographic challenge that a miner must solve to have the right to append a new block in the blockchain. If a participant solves the challenge, he/she receives a remuneration. The main advantage of the proof of work is its high scalability since anyone can participate and mine blocks, being an adequate consensus for public networks.

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 number lesser than a predetermined number. After solving the challenge, the participant publishes the block and the solution. Other members of the network can verify that the challenge was solved correctly 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 establish a constant block creation rate. PoW rewards the block miner to encourage participants to spend computing power and solve the challenge.

The main disadvantages of proof of work are high energy consumption and low transaction throughput. The high

computational cost involved in calculating proof of work in Bitcoin consumes annually an amount of energy comparable to the power consumption of Switzerland [11]. Furthermore, the addition of new blocks to the blockchain in Bitcoin is slow, with the cryptocurrency showing an average throughput of 7 transactions per second. This value is considerably lesser than the average of 2000 transactions per second recorded by credit card companies [12], preventing the use of PoW-based cryptocurrencies for everyday purchases.

A. 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 as double-spending attacks, attacks on consensus, or attacks on the network.

Double-spending attacks aim to use the same currency in multiple transactions. Unlike physical currency, it is easy to replicate the 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 [1]. Double-spending attacks, however, are still possible on the Bitcoin network [13]. 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 the addition of 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 [14]. 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 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[5]. Collusion between only four independent entities would be able to subvert the system completely. Thus, contrary to the initial proposal of the decentralization of Bitcoin, four agents would centralize the power of the network. This type of attack occurred in alternative proof-based protocols[6,7].

Selfish mining [15] is an attack that exploits the convergence algorithm or 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 to 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. Besides, the nodes forget all existing transactions in the abandoned fork if they do not exist in the attacker's blocks, thus allowing double-spending.

The block discarding attack [16] is an extension of the selfish mining attack. 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.

The bribery attack occurs when an attacker without sufficient computational power to attack the network bribes miners with higher processing capacity to form a collusion during a given period [17]. The network, however, loses trust if the malicious node can use this strategy to carry out other attacks such as double-spending, thus devaluing the currency. In this way, miners who are investors in the currency, since they have assets thanks to the incentive obtained by the discovery of new blocks, lose the money invested, or have their profit reduced. Therefore 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, as communication in blockchain environments is distributed, and the protocol allows for temporary inconsistencies. If the attacker is successful, victims of network attacks can remain in incorrect states for long periods due to a lack of information about the global state of the network.

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

[5]Available at https://btc.com/stats/pool. Accessed August 6, 2020.

[6]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.

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

processing between his/her identities. Due to distributed communication, an attacker, however, 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. The latter cited below.

Another way of controlling information from part of the network is to perform the eclipse attack [18]. For this, 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 knows only the nodes controlled by the attacker, 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 a large number of 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 [19].

IV. The Proof of Stake Consensus (PoS)

Proof of Stake (PoS) is the most known alternative consensus to Proof of Work, as they provide similar characteristics without requiring high energy expenditure. The main advantages of proof of stake over proof of work include high energy efficiency, high performance, and greater security.

Proof of stake is a category of proof-based algorithms for public blockchains whose main characteristic is to achieve consensus based on the amount of stake held by each participant.Compared to proof of work, in which the probability of a participant proposing a block is proportional only to its hashpower, in proof of stake, the probability of proposing a block is proportional to the amount of stake 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 [20], [21]. 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 the power of each participant is proportional to their respective deposits in relation to the total.

The implementation of a proof-of-stake consensus can follow two main approaches: i) a probabilistic proof of stake in which a participant with more stake is more likely to propose a block; or ii) a deterministic proof of stake 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 [21], [20]. The criterion for selecting the bidder can be a drawing based on the stakes, as in the Ouroboros cryptocurrency [22], or an election, as in the EOSIO cryptocurrency [23]. In addition to the two approaches, each consensus protocol presents specific details, such as how to incentive validators and mechanisms to prevent attacks, which generates several practical ways to implement a proof of stake. Rather than looking at specific protocols, this paper focuses on a probabilistic approach to provide a general security analysis of proof of stake.

The probabilistic proof of stake inherits characteristics similar to Nakamoto's proof of work [1], such as the pseudo-random selection of a participant to add a block, the longest chain rule, and the probabilistic finality. Bitcoin developers proposed in 2011 the first family of probabilistic proof of stake consensuses, 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 decreases according to the participant's stake. Although the validation process is similar to the proof of work procedure, the average difficulty of solving the computational challenge is significantly lesser than that of Bitcoin. Therefore, PoS avoids brute force based competition, characteristic of the 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 chance to participants with more stake. The MPC receives the current state of the blockchain, which includes the assets of each participant, and selects a pseudo-random sequence of upcoming bidders that can be verified by any participant. Participants can be chosen more than once and receive more time to propose blocks if they have more stake.

A. Probabilistic 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. The non-requirement of deposits, however, 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, which will be followed by any rational validator since there is no computational cost to validate transactions at multiple forks, as opposed to proof of work. Therefore, the behavior of simultaneously validating several forks becomes computationally efficient, which corresponds to several chances of winning without any risk of loss. Thus, the action that maximizes the probability of gains is to participate in all possible forks. Consequently, every rational participant who wants to maximize their profit follows this behavior.

The "nothing-at-stake" problem can be modeled mathematically as a probability maximization problem. Let be a blockchain fork with two conflicting paths[8] 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:

[8]Conflicting paths are paths that start from the same source block and have the same height and, therefore, it is not enough to simply apply Nakamoto's rule of the largest chain [1].

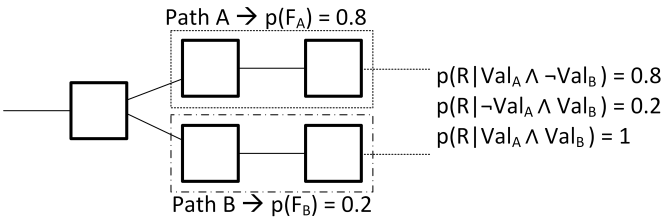

Fig. 2: A forked blockchain with two conflicting paths A and B with different probabilities of being finalized by the system. The best strategy for a participant to guarantee a R reward is to validate the two paths, contributing to the fork prolongation.

- F_A: the system eventually finalizes[9] 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. The odds that the participant will be rewarded considering each possible scenario are:

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

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, and that, 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[10]. In proof of work, this problem does not occur, since dividing the computational power between the forks does not increase the chance of mining a block.

The main 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 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 rounds of block validation.

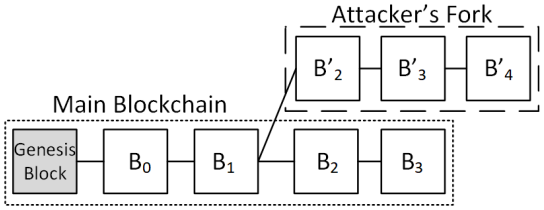

Fig. 3: Execution of a long-range attack. The attacker creates a fork in a block accepted by the network and tries to rewrite the main chain.

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 [24]. 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 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.

One of the ways to mitigate long-range attacks involves the implementation of checkpoints that restrict forking the blockchain at height before the checkpoint. This countermeasure limits the range of the attack by preventing attackers from generating forks at points very far from the main blockchain.

V. Proof-Based Alternatives: Proof-of-X (PoX)

The proof-based algorithms alternative to proof of work seek to mitigate the performance limitations and excess energy expenditure of the proof of work, in addition to the "nothing at stake" and the long-range attack problems of the proof

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

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

of stake [22]. Table I presents a performance and scalability comparison of the main alternative protocols and the proof of work. Follows explanations of the most well-known protocols.

Delegated Proof of Stake (DPoS)[11]**.** Participants use their assets to elect delegates in a quorum that defines the next block. The number of votes of a miner is proportional to its stake [23]. Centralizing delegates has the advantage of increasing efficiency. Nevertheless, the centralization of the DPoS model presents clear 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. 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 stake.

Proof of Authority (PoA)[12]**.** Similar to DPoS, however, the set of delegates (authorities) is predetermined by agreement, and their identities are public and verifiable by any member of the network [25]. The main advantage is the easy inspection of the authorities, and the main disadvantage is the centralization in authorities with no possibility of an election. Ekparinya *et al.* developed the cloning attack, in which a malicious delegate clones his/her private key and starts to act in two instances of the blockchain [26]. In a network with n odd delegates, it issues a transaction to only $(n-1)/2$ delegates, 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 delegates to delay the branch with the transaction long enough. Then the other branch becomes the longest.

Proof of Elapsed Time (PoET)[13]**.** Each participant sets a random decreasing timer, and the node whose timer ends first proposes the next block [10]. The consensus protocol works exclusively on hardware that supports Intel Software Guard eXtensions (SGX) technology. Intel SGX guarantees, through private memory regions, the random distribution of timers and that no entity has access to more than one consensus participant. The main advantage is to provide safe and efficient consensus without high processing costs, and the main disadvantage is the dependence on specific hardware. The PoET protocol security depends on the SGX security and Intel hardware enclaves, which have been exploited by attackers in the past. Chen *et al.* demonstrate that if technology can be compromised, the protocol security is inversely proportional to the number of participants, which undermines its scalability [27]. The authors prove that it is necessary to commit only $\Theta(log(log(n))/log(n))$ of the participants to subvert the consensus that corresponds to 30% of 1000 participants.

[11]The EOSIO platform uses DPoS as its consensus protocol.

[12]VeChain Thor and POA cryptocurrencies use proof of authority as consensus protocol.

[13]The PoET is used in the Hyperledger Sawtooth platform.

VI. DAG-BASED CONSENSUS: IOTA TANGLE

IOTA is a cryptocurrency built to serve machine-to-machine (M2M) micro-payments in the Internet of Things. Its consensus protocol, formalized by Popov in 2017 [28], 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. A notable feature of IOTA consensus compared to the blockchain consensus is 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. 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 [20], [28], [21] regard IOTA as the next generation of distributed ledger technologies as IOTA claims to provide: i) unlimited throughput and scalability because the more users join the network, the more hashing 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.

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[14]. 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.

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

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

TABLE I: Comparison of blockchain consensus protocols.

Platform/Protocol	Consensus Mechanism	Maximum Throughput	# validators
Bitcoin	Proof of Work (PoW)	$\approx$ 7 tx/s	Thousands
Ethereum/Ethash	Proof of Work (PoW)	$\approx$ 15 tx/s	Thousands
Cardano/Ouroboros	Proof of Stake (PoS)	$\approx$ 250 tx/s	Hundreds
EOSIO	Delegated Proof of Stake (DPoS)	$\approx$ 4000 tx/s	Dozens
VeChain Thor, POA	Proof of Authority (PoA)	$\approx$ 165 tx/s	Thousands
Hyperledger Sawtooth	Proof of Elapsed Time (PoET)	$\approx$ 1150 tx/s	Hundreds

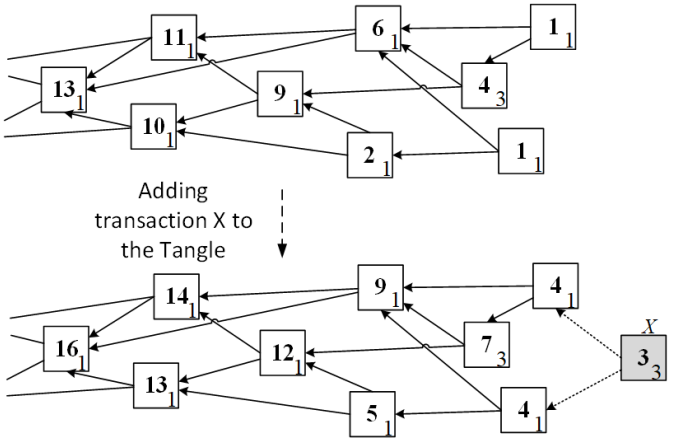

Fig. 4: The addition of a tip, X, into the Tangle data structure. The lower right corner of each box represents the individual weight of each transaction and the number in the center of the box represents the cumulative weight.

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 do 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 Tangled could be explored to create multiple attacks [28]. 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 [29]. The main problem, however, is that IOTA depends on user hashpower 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.

VII. Related Works

Cryptocurrencies play a paradigm shift in today's society, with Bitcoin and Ethereum 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 [30], [31], [32]. 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 blockchains based on proof of work [33]. Xiao *et al*. model the security of proof of work according to the participants' connectivity concerning selfish mining attacks and the collusion between participants [34]. Conti *et al.* analyze various components and their respective vulnerabilities in the Bitcoin blockchain [35]. Li *et al.* analyze the security of consensus based on proof of stake [36]. Li *et al.* Summarize the main security vulnerabilities in blockchain systems [37]. 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.* [20] and Joshi *et al.* [38] 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 [39]. 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 crucial vulnerabilities and attacks of each protocol, with their respective countermeasures.

VIII. Conclusion

Proof-based protocols, unlike deterministic protocols, present possibilities for forks, since any participant can propose a block and there is a probability of simultaneously proposing blocks. Malicious participants can exploit these temporary inconsistencies to launch various attacks, which are not possible in deterministic protocols. Proof of work is the first probabilistic consensus protocol successfully applied to a public network. Its energy cost, however, is prohibitive. Rewarded mining leads to the centralization of powerful miners who can afford high-performance hardware.

Proof of stake is an energy-efficiency alternative to proof of work, but it presents new vulnerabilities such as "nothing at stake" and "long range". Proof of stake also requires reward to incentive the "bets", and 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 that increases the risk of attacks. The 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 and, therefore, more sensitive to collusion between malicious participants. 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.

Despite the number of vulnerabilities of the proof of work consensus, it is a fact that, in practice, Bitcoin's security is exceptional, 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.

In future works, we intend to study hybrid protocols. We expected that the best consensus proposal combines the deterministic consensus with the probabilistic consensus.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, Last access: 20 July 2020. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] M. T. de Oliveira *et al.*, "Blockchain reputation-based consensus: A scalable and resilient mechanism for distributed mistrusting applications," *Computer Networks*, p. 107367, 2020.

[3] G. A. F. Rebello *et al.*, "Providing a sliced, secure, and isolated software infrastructure of virtual functions through blockchain technology," in *IEEE HPSR*, 2019, pp. 1–6.

[4] G. R. Carrara, L. H. Reis, C. V. Albuquerque, and D. M. Mattos, "A lightweight strategy for reliability of consensus mechanisms based on software defined networks," in *GIIS*. IEEE, 2019, pp. 1–6.

[5] G. A. F. Rebello, I. D. Alvarenga, I. J. Sanz, and O. C. M. Duarte, "Bsec-nfvo: A blockchain-based security for network function virtualization orchestration," in *IEEE ICC*, 2019, pp. 1–6.

[6] I. D. Alvarenga, G. A. F. Rebello, and O. C. M. B. Duarte, "Securing configuration management and migration of virtual network functions using blockchain," in *IEEE/IFIP NOMS 2018*, Apr. 2018, pp. 1–9.

[7] G. F. Camilo, G. A. F. Rebello, L. A. C. de Souza, and O. C. M. B. Duarte, "AutAvailChain: Automatic and secure data availability through blockchain," in *IEEE GLOBECOM*, 2020, pp. 1–6, to be published.

[8] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *JACM*, vol. 32, no. 2, pp. 374–382, 1985.

[9] L. H. M. K. Costa, S. Fdida, and O. C. M. B. Duarte, "Incremental service deployment using the hop-by-hop multicast routing protocol," *IEEE/ACM Transactions on Networking*, vol. 14, no. 3, pp. 543–556,, 2006.

[10] K. Olson, M. Bowman, J. Mitchell, S. Amundson, D. Middleton, and C. Montgomery, "Sawtooth: An Introduction," *Linux Foundation*, 2018.

[11] Digiconomist, "Bitcoin Energy Consumption Index," 2020, Last access: 20 July 2020. [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption/

[12] BitcoinWiki, "Bitcoin Scalability," 2019, Last access: 20 July 2020. [Online]. Available: https://en.bitcoin.it/wiki/Scalability

[13] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *ACM CCS 2012*, 2012, pp. 906–917.

[14] H. Finney, "Best practice for fast transaction acceptance-how high is the risk?" 2011, Available at: https://bitcointalk.org/index.php?topic=3441.msg48384#msg48384. Last access: 20 July 2020.

[15] I. Eyal and E. G. Sirer, "Majority is Not Enough: Bitcoin Mining is Vulnerable," *Commun. ACM*, vol. 61, no. 7, pp. 95–102, Jul. 2018. [Online]. Available: http://doi.acm.org/10.1145/3212998

[16] L. Bahack, "Theoretical Bitcoin attacks with less than half of the computational power (draft)," *arXiv preprint arXiv:1312.7013*, 2013.

[17] J. Bonneau, E. W. Felten, S. Goldfeder, J. A. Kroll, and A. Narayanan, "Why buy when you can rent?" in *ICFCDS*. Springer, 2016, pp. 19–26.

[18] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *USENIX Security'15*, Aug. 2015, pp. 129–144.

[19] B. Johnson, A. Laszka, J. Grossklags, M. Vasek, and T. Moore, "Game-theoretic analysis of DDoS attacks against Bitcoin mining pools," in *ICFCDS*, 2014, pp. 72–86.

[20] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[21] W. Wang *et al.*, "A survey on consensus mechanisms and mining management in blockchain networks," *CoRR*, vol. abs/1805.02707, 2018. [Online]. Available: http://arxiv.org/abs/1805.02707

[22] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *CRYPTO*, 2017, pp. 357–388.

[23] D. Larimer, "EOS.IO White Paper," 2017, Available at: https://developers.eos.io/-welcome/latest/protocol/consensus_protocol. Last access: 20 July 2020.

[24] E. Deirmentzoglou, G. Papakyriakopoulos, and C. Patsakis, "A survey on long-range attacks for proof of stake protocols," *IEEE Access*, vol. 7, pp. 28 712–28 725, 2019.

[25] S. D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain," in *Italian Conference on Cyber Security (06/02/18)*, January 2018. [Online]. Available: https://eprints.soton.ac.uk/415083/

[26] P. Ekparinya, V. Gramoli, and G. Jourjon, "The attack of the clones against proof-of-authority," *arXiv preprint arXiv:1902.10244*, 2019.

[27] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, "On security analysis of proof-of-elapsed-time (POET)," in *International Symposium on Stabilization, Safety, and Security*. Springer, 2017, pp. 282–297.

[28] S. Popov, "The Tangle," *cit. on*, p. 131, 2017, Last access: 20 July 2020. [Online]. Available: http://www.descryptions.com/Iota.pdf

[29] G. Bu, Ö. Gürcan, and M. Potop-Butucaru, "G-IOTA: Fair and confidence aware tangle," in *IEEE INFOCOM WKSHPS*, 2019, pp. 644–649.

[30] D. M. F. Mattos, F. Krief, and S. J. Rueda, "Blockchain and artificial intelligence for network security," 2020.

[31] M. T. Oliveira *et al.*, "Towards a performance evaluation of private blockchain frameworks using a realistic workload," in *ICIN*. IEEE, 2019, pp. 180–187.

[32] G. R. Carrara, L. M. Burle, D. S. Medeiros, C. V. N. de Albuquerque, and D. M. Mattos, "Consistency, availability, and partition tolerance in blockchain: a survey on the consensus mechanism over peer-to-peer networking," *Annals of Telecommunications*, pp. 1–12, 2020.

[33] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *ACM SIGSAC*, 2016, pp. 3–16.

[34] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "Modeling the impact of network connectivity on consensus security of proof-of-work blockchain," *arXiv preprint arXiv:2002.08912*, 2020.

[35] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of Bitcoin," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[36] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *DPM/CBT*. Springer, 2017, pp. 297–315.

[37] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *FGCS*, vol. 107, pp. 841–853, 2020.

[38] A. P. Joshi, M. Han, and Y. Wang, "A survey on security and privacy issues of blockchain technology," *MFC*, vol. 1, no. 2, p. 121, 2018.

[39] P. Zhang and M. Zhou, "Security and trust in blockchains: Architecture, key technologies, and open issues," *IEEE TCSS*, vol. 7, no. 3, pp. 790–801, 2020.